INTENT-IN-THE-LOOP: EXPLICIT CINEMATOGRAPHIC INTENT FOR 3D DANCE CAMERA SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthesizing 3D camera trajectories from music and dance requires more than generating plausible camera motion. It requires translating high-level cinematographic intent into precise and dynamically expressive camera movements. Existing methods have made substantial progress by directly mapping music and dance to camera trajectories. However, they leave cinematographic intent implicit and provide no explicit mechanism for verifying whether the generated trajectory reflects that intent. We introduce **Intent-in-the-Loop**, a closed-loop framework that makes cinematographic intent explicit and verifiable through a **plan-generate-align** process. First, in **plan**, we plan shot-level cinematographic intent from music and 3D dance motion. Second, in **generate**, we generate a camera trajectory in a coarse-to-fine manner conditioned on the planned intent. Third, in **align**, we recover intent from the output trajectory and align it with its planned intent during training. This loop improves consistency between planned intent and generated trajectories. Extensive experiments on the DCM dataset demonstrate state-of-the-art camera trajectory quality, substantially smoother motion dynamics, and improved ground truth intent fidelity, while providing explicit shot-level interpretability. Our source code will be publicly available upon publication.

1 INTRODUCTION

In dance performances, camera work enhances the audience's storytelling experience by better capturing and emphasizing the dancer's key movements. Therefore, generating effective camera work requires careful synthesis guided by highly planned cinematographic intent, including the interpretation of 3D body motion, effective framing, performer visibility, smooth transitions within each shot, and abrupt transitions between shots. However, due to these complex factors, camera trajectory synthesis still remains challenging.

Existing dance-camera methods have approached this problem primarily by learning direct mappings from music and dance to camera motion. DanceCamera3D (Wang et al., 2024a) employs a transformer-based diffusion model to synthesize 3D camera trajectory from paired music and dance inputs. DanceCamAnimator (Wang et al., 2024b) divides the full sequence into several shots using keyframes and then employs tween functions to generate smooth camera trajectories within each shot. TemMEGA (Xu et al., 2025) represents camera motions as discrete tokens and predicts them using a conditional masked transformer with temporal context. These methods directly synthesize camera trajectories from music and dance without explicitly modeling the cinematographic intent underlying the resulting camera motion. Due to the lack of explicitly disentangled cinematographic intent, these methods may produce smooth and physically plausible trajectories while still failing to capture the high-level cinematographic decisions that distinguish purposeful camera work from generic motion synthesis.

To address this challenge, we propose *Intent-in-the-Loop (IITL)*, which explicitly incorporates cinematographic intent into the intermediate stages of camera trajectory synthesis. Fig. 1 demonstrates the comparison between previous direct camera synthesis approach and our Intent-in-the-Loop. Unlike direct synthesis, which leaves cinematographic intent implicit, IITL explicitly incorporates it through a plan–generate–align process, enabling camera trajectories to better reflect high-level cinematographic decisions. In **plan**, we predict cinematographic intent from music and 3D dance motion, including shot boundaries, framing scale, visual focus, camera motion primitives, camera

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.

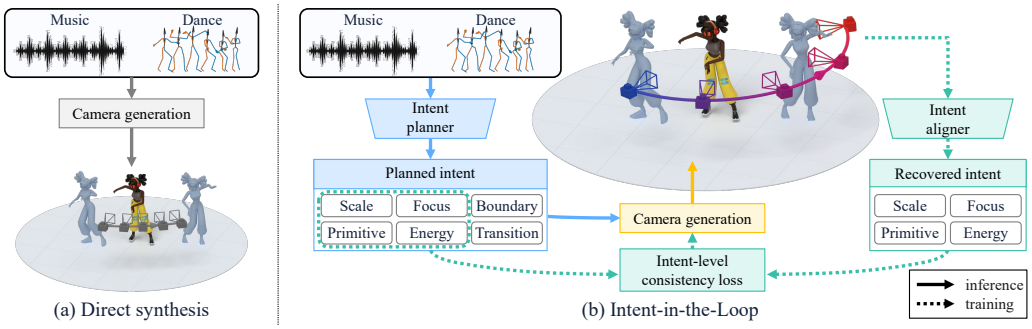

Figure 1: Formulations of music-and-dance-driven camera synthesis. **(a) Direct synthesis:** without explicit cinematographic intent. **(b) Intent-in-the-Loop:** intent-conditioned generation with alignment of recovered and planned intent. Dashed arrows indicate intent recovery and alignment used only during training.

motion energy, transition type, and keyframe. In **generate**, the camera trajectory is synthesized in a coarse-to-fine manner. The coarse trajectory is constructed by combining shot-level and frame-level camera motions decoded from the planned intent. We then refine the coarse trajectory using an intent-conditioned residual diffusion model to produce a fine-grained camera trajectory. In **align**, we recover the cinematographic intent from the generated camera trajectory. Here, we introduce an intent alignment loss that optimizes the camera generator by aligning the intent predicted in the **plan** stage with that recovered in the **align** stage. By explicitly feeding the consistency back into the generate process, this closed-loop formulation encourages the model to capture high-level cinematographic decisions beyond low-level motion plausibility, leading to camera movements that better reflect the intended shot structure, visual emphasis, and dynamic style.

To demonstrate the effectiveness of our method, we perform extensive experiments on the standard DCM benchmark dataset (Wang et al., 2024a). Both objective and subjective evaluations demonstrate that our method consistently outperforms existing approaches. In summary, our main contributions are as follows:

- Motivated by the importance of cinematographic intent, we integrate and formalize explicit shot-level intent into the dance camera synthesis process, enabling more plausible and intent-aware camera synthesis.
- We propose Intent-in-the-Loop (IITL), a novel **plan–generate–align** framework for camera synthesis that explicitly models cinematographic intent, enabling the generation of plausible camera trajectories that faithfully reflect the intended cinematographic decisions.
- Through extensive experiments, we demonstrate that the proposed method better reflects cinematographic intent than existing approaches and enables more plausible camera synthesis in subjective evaluations.

2 Related Work

Music-and-dance-conditioned camera synthesis. Rule-based virtual cinematography uses shot conventions, geometric objectives, continuity constraints, and trajectory smoothing (He et al., 1996; Christie et al., 2005; Galvane et al., 2015; Litteneker & Terzopoulos, 2017; Lee et al., 2026), but relies on hand-designed rules and scene abstractions. Learning-based camera synthesis estimates trajectories from data (Dehghanian et al., 2026); YOTO incorporates learned dance-camera synthesis into an end-to-end personalized 3D music-video system (Lee et al., 2025). DanceCamera3D (Wang et al., 2024a) uses transformer-based diffusion conditioned on music and dance; DanceCamAnimator (Wang et al., 2024b) separates keyframes from in-between motions and employs tween functions; TemMEGA (Xu et al., 2025) uses discrete tokens and a temporal conditional masked transformer. These learning-based methods leave framing scale, visual focus, camera motion primitives, and camera motion energy implicit; IITL explicitly models cinematographic intent.

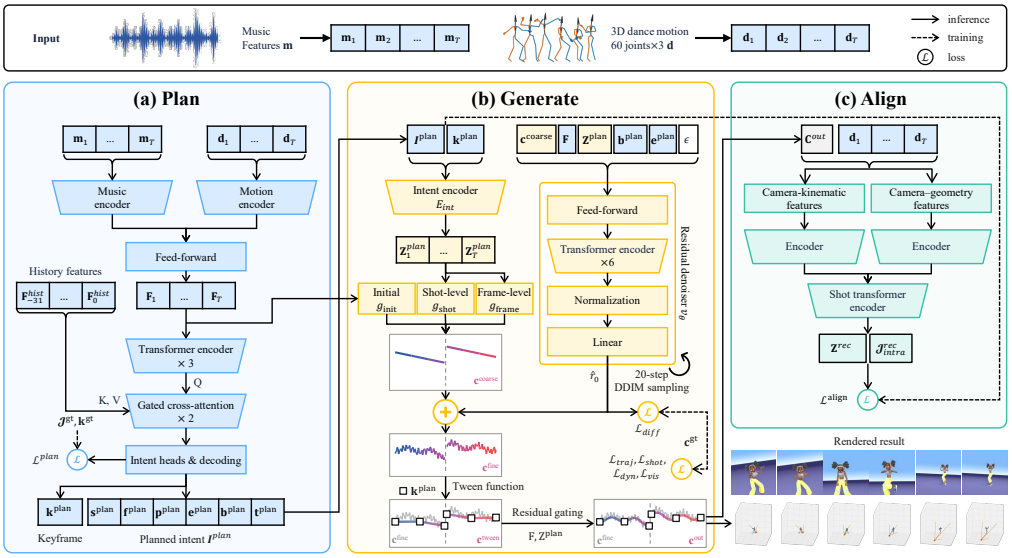

Figure 2: Overview of Intent-in-the-Loop (IITL). **(a) Plan:** intra- and inter-shot intent planning, and keyframe prediction. **(b) Generate:** coarse-to-fine camera generation with intent-conditioned residual diffusion and refinement using planned keyframes. **(c) Align:** alignment of planned intent with intent estimated from the generated camera trajectory with dance motion during training.

Controllable camera synthesis. The Cinematographic Camera Diffusion Model (Jiang et al., 2024) conditions camera trajectories on descriptions and optional control frames. E.T. (Courant et al., 2025) distinguishes camera-only from camera-character captions and reports higher controllability; GenDoP (Zhang et al., 2025) distinguishes motion captions from directorial captions describing scene interaction and directorial intent, reporting better controllability and motion stability across both caption types. Director3D (Li et al., 2024) predicts text-conditioned trajectories for open-world 3D generation. Inspired by these explicit controls, IITL models cinematographic intent to guide dance-camera trajectory generation.

Planning feedback and intent consistency. Cycle consistency (Zhu et al., 2017) constrains forward and reverse mappings through reconstruction. InfoNCE (van den Oord et al., 2018) learns representations by contrasting matched pairs against negatives. In human motion generation, CLoSD (Tevet et al., 2025) feeds physically executed motions back to its diffusion-based planner to improve physical plausibility and task success. In text-to-camera generation, VERTIGO (Li et al., 2026) uses a cinematography-aware vision-language model to score rendered previews and preference feedback to improve condition adherence, framing quality, and perceptual realism. IITL introduces an intent-level cycle-consistency loss to align planned intent with intent recovered from generated camera trajectories.

3 Camera Synthesis with Intent-in-the-Loop

3.1 Problem Formulation

Camera synthesis. As shown in Fig. 2, given synchronized music features $\mathbf{m} \in \mathbb{R}^{T\times 35}$ and 3D dance motion $\mathbf{d} \in \mathbb{R}^{T\times 60\times 3}$ over T frames, we synthesize a frame-aligned camera trajectory $\mathbf{c}^{\text{out}} = (\mathbf{c}_1^{\text{out}}, \ldots, \mathbf{c}_T^{\text{out}}) \in \mathbb{R}^{T\times d_C}$. Each $\mathbf{c}_j^{\text{out}} \in \mathbb{R}^{d_C}$ is the camera-parameter vector at frame j, and their sequence forms the camera trajectory. Following DanceCamera3D (Wang et al., 2024a), each vector contains $d_C = 8$ camera parameters: camera-to-target distance, 3D target position, Euler rotation, and field of view (FOV). The 3D camera position is computed from these parameters. $\mathbf{c}^{\text{gt}}$ is the ground truth trajectory.

Intent Representation. Following shot-level cinematographic attributes and camera control (Jiang et al., 2022; 2024), we define intra-shot intent $\mathcal{I}_{\text{intra}}$ and inter-shot intent $\mathcal{I}_{\text{inter}}$, with

$\mathcal{I} = \mathcal{I}_{\text{intra}} \cup \mathcal{I}_{\text{inter}}$. Let S be the number of shots in the T-frame window. For shot $i \in \{1, \ldots, S\}$, framing scale $\mathbf{s}_i \in \{0,1\}^4$ and visual focus $\mathbf{f}_i \in \{0,1\}^4$ are one-hot vectors over (close-up, medium, full, long) and (head, upper body, full body, lower body), respectively. Camera motion primitives $\mathbf{p}_i \in \{0,1\}^7$ are multi-label (static, pan, tilt, dolly, orbit, track, crane); camera motion energy $e_i \in [0,1]$ is normalized motion intensity. We define the intra-shot intent $\mathcal{I}_{\text{intra}} = \{\mathbf{s}, \mathbf{f}, \mathbf{p}, \mathbf{e}\}$ using the within shot attributes. The framewise binary boundary mask $\mathbf{b} \in \{0,1\}^T$ marks the start and end frames of each shot. The framewise binary transition mask $\mathbf{t} \in \{0,1\}^T$ indicates which shot boundaries correspond to hard cuts. We define these attributes between shots as inter-shot intent $\mathcal{I}_{\text{inter}} = \{\mathbf{b}, \mathbf{t}\}$. Beside the intent, we define the framewise binary keyframe mask $\mathbf{k} \in \{0,1\}^T$. The keyframe set $\mathcal{K} = \{k_1, \ldots, k_{N_K}\}$ is formed from the frames for which the keyframe mask is positive, with $1 \le k_1 < \cdots < k_{N_K} \le T$.

3.2 Intent Planner: Encoding and Intent Planning

The intent planner first encodes music and dance motion into frame-aligned features. We extract music features $\mathbf{m}$ using Librosa (McFee et al., 2015) with FACT's configuration (Li et al., 2021). The music feature $\mathbf{m}$ and dance motion $\mathbf{d}$ are separately fed to four-layer transformer encoders (Vaswani et al., 2017) and then fused into a multimodal representation $\mathbf{F} \in \mathbb{R}^{T \times 256}$. For a history window of $T^{\text{hist}} = 32$ strictly before the current window, the same encoders and fusion layer are used to produce history representation $\mathbf{F}^{\text{hist}} \in \mathbb{R}^{T^{\text{hist}} \times 256}$.

For intent prediction, intent planner E_{plan} embeds $\mathbf{F}$ with a three-layer transformer encoder, then applies two-layer cross-attention using these hidden states as queries and $\mathbf{F}^{\text{hist}}$ as keys and values. Inspired by gated attention (Qiu et al., 2025), a sigmoid gate fuses the current hidden states with the cross-attention output. Seven transformer heads predict framewise intents and keyframe as

$$\mathbf{b}^{\text{plan}}, \mathbf{t}^{\text{plan}}, \mathbf{s}^{\text{plan}}, \mathbf{f}^{\text{plan}}, \mathbf{p}^{\text{plan}}, \mathbf{e}^{\text{plan}}, \mathbf{k}^{\text{plan}} = E_{\text{plan}}(\mathbf{F}, \mathbf{F}^{\text{hist}}). \tag{1}$$

For each shot i, intra-shot intent attributes $\mathbf{s}_i^{\text{plan}}, \mathbf{f}_i^{\text{plan}}, \mathbf{p}_i^{\text{plan}}, e_i^{\text{plan}}$ are obtained by mean pooling the corresponding framewise predictions within the shot.

To train the intent planner, we first define the shot-level intra-shot intent loss as follows:

$$\mathcal{L}_{\text{intra}}^{\text{plan}} = \text{WCE}(\mathbf{s}_i^{\text{plan}}, \mathbf{s}_i^{\text{gt}}) + \text{WCE}(\mathbf{f}_i^{\text{plan}}, \mathbf{f}_i^{\text{gt}}) + \text{WBCE}(\mathbf{p}_i^{\text{plan}}, \mathbf{p}_i^{\text{gt}}) + \text{SL1}(e_i^{\text{plan}}, e_i^{\text{gt}}), \tag{2}$$

where i denotes the shot index, and WCE, WBCE, and SL1 denote weighted cross-entropy, weighted binary cross-entropy, and smooth L1, respectively. Here, since camera motion primitive $\mathbf{p}_i^{\text{gt}}$ is represented as a multi-label vector rather than a one-hot vector, we apply weighted binary cross-entropy independently to each dimension and average the resulting losses. In addition, we define losses for inter-shot intent, keyframe, and temporal consistency as follows:

$$\mathcal{L}_{\text{inter}}^{\text{plan}} = \text{FBCE}(\mathbf{b}^{\text{plan}}, \mathbf{b}^{\text{gt}}) + \text{WBCE}(\mathbf{t}^{\text{plan}}, \mathbf{t}^{\text{gt}}) \tag{3}$$

$$\mathcal{L}_{\mathcal{K}}^{\text{plan}} = \text{WBCE}\left(\mathbf{k}^{\text{plan}}, \mathbf{k}^{\text{gt}}\right), \tag{4}$$

$$\mathcal{L}_{\text{temp}}^{\text{plan}} = \frac{\sum_{j=2}^{T} m_j^{\text{temp}} (|b_j^{\text{plan}} - b_{j-1}^{\text{plan}}| + \sum_{\mathbf{x} \in \{\mathbf{s},\mathbf{f},\mathbf{p}\}} \|\mathbf{x}_j^{\text{plan}} - \mathbf{x}_{j-1}^{\text{plan}}\|_2)}{\max\left(1, \sum_{j=2}^{T} m_j^{\text{temp}}\right)}, \tag{5}$$

where FBCE denotes focal binary cross-entropy (Lin et al., 2017), and the binary mask m_j^{temp} indicates whether the current frame j and previous frame belong to the same shot according to planned boundary $\mathbf{b}^{\text{plan}}$. Therefore, training objective $\mathcal{L}^{\text{plan}}$ for intent planner is defined as:

$$\mathcal{L}^{\text{plan}} = \mathcal{L}_{\text{intra}}^{\text{plan}} + \mathcal{L}_{\text{inter}}^{\text{plan}} + \mathcal{L}_{\mathcal{K}}^{\text{plan}} + \mathcal{L}_{\text{temp}}^{\text{plan}}. \tag{6}$$

3.3 Camera Generator: Coarse-to-Fine Generation

The camera generator produces the trajectory in a coarse-to-fine manner. To construct the coarse camera trajectory, we first encode the planned intent as

$$\mathbf{Z}_j^{\text{plan}} = E_{\text{int}}\left(b_j^{\text{plan}}, \mathbf{s}_i^{\text{plan}}, \mathbf{f}_i^{\text{plan}}, \mathbf{p}_i^{\text{plan}}, e_i^{\text{plan}}\right), \tag{7}$$

where E_{int} is a transformer encoder within the generator, j indexes frames, and i indexes the shot containing frame j. From $\mathbf{Z}^{\text{plan}} = (\mathbf{Z}_1^{\text{plan}}, \ldots, \mathbf{Z}_T^{\text{plan}})$, we generate shot-level $\mathbf{c}_i^{\text{shot}}$ and frame-level $\mathbf{c}_j^{\text{frame}}$ camera trajectories that respectively model hard-cut transitions and continuous motion:

$$\mathbf{c}_1^{\text{shot}} = g_{\text{init}}\left(\frac{1}{T}\sum_{k=1}^{T}\mathbf{F}_k\right), \tag{8}$$

$$\mathbf{c}_i^{\text{shot}} = \mathbf{c}_{i-1}^{\text{shot}} + t_j^{\text{plan}} g_{\text{shot}}(\mathbf{Z}_j^{\text{plan}}), \qquad i > 1,\ b_j^{\text{plan}} = 1, \tag{9}$$

$$\mathbf{c}_j^{\text{frame}} = e_i^{\text{plan}} \cdot \left(\frac{\sum_{r=1}^{7} p_{i,r}^{\text{plan}} \mathbf{V}_r}{\sum_{r=1}^{7} p_{i,r}^{\text{plan}}} + g_{\text{frame}}(\mathbf{Z}_j^{\text{plan}})\right), \tag{10}$$

where $g_{\text{init}} : \mathbb{R}^{|\mathbf{F}_k|} \to \mathbb{R}^{d_C}$ and $g_{\text{shot}}, g_{\text{frame}} : \mathbb{R}^{|\mathbf{Z}_j^{\text{plan}}|} \to \mathbb{R}^{d_C}$ are linear projection layers into camera-parameter space, and r indexes camera motion primitives. The projection g_{init} initializes $\mathbf{c}_1^{\text{shot}}$ from the temporal mean of $\mathbf{F}$. Each $\mathbf{V}_r \in \mathbb{R}^{d_C}$ denotes a trainable camera-motion basis for primitive r, and the primitive-dependent component is constructed by taking their weighted average according to the predicted primitive probabilities.

The coarse camera-parameter $\mathbf{c}_j^{\text{coarse}}$ at j^{th} frame is obtained by combining the shot-level camera-parameter with the per-frame increments accumulated since the most recent hard cut:

$$\mathbf{c}_j^{\text{coarse}} = \mathbf{c}_i^{\text{shot}} + \sum_{k=j_{\text{cut}}}^{j} \mathbf{c}_k^{\text{frame}}, \tag{11}$$

where $j_{\text{cut}} = \max\left(\{1\} \cup \left\{k \mid 1 < k \le j,\ b_k^{\text{plan}} = t_k^{\text{plan}} = 1\right\}\right)$ indexes the most recent hard-cut frame at or before j.

Inspired by residual diffusion (Shang et al., 2024), we refine the coarse trajectory using residual trajectory $\mathbf{r}_0 = \mathbf{c}^{\text{gt}} - \mathbf{c}^{\text{coarse}}$ and forward diffusion process $q(\mathbf{r}_\tau \mid \mathbf{r}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_\tau}\mathbf{r}_0, (1-\bar{\alpha}_\tau)\mathbf{I})$, where $\bar{\alpha}_\tau$ is the cumulative noise-schedule coefficient and τ the diffusion step. With standard v-parameterization (Salimans & Ho, 2022), the generator's denoising function v_θ predicts $\widehat{\mathbf{v}}_\tau$ to match target $\mathbf{v}_\tau$:

$$\mathbf{v}_\tau = \sqrt{\bar{\alpha}_\tau}\boldsymbol{\epsilon} - \sqrt{1-\bar{\alpha}_\tau}\mathbf{r}_0, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{12}$$

$$\widehat{\mathbf{v}}_\tau = v_\theta\left(\mathbf{r}_\tau, \tau, \mathbf{F}, \mathbf{Z}^{\text{plan}}, \mathbf{c}^{\text{coarse}}, \mathbf{b}^{\text{plan}}, \mathbf{e}^{\text{plan}}\right), \tag{13}$$

where $\mathbf{v}_{\tau,j}$ and $\widehat{\mathbf{v}}_{\tau,j}$ are framewise values of $\mathbf{v}_\tau$ and $\widehat{\mathbf{v}}_\tau$, and $\mathbf{c}^{\text{coarse}} = (\mathbf{c}_1^{\text{coarse}}, \ldots, \mathbf{c}_T^{\text{coarse}})$ is the coarse camera trajectory. At inference, DDIM sampling (Song et al., 2021) estimates the clean residual $\widehat{\mathbf{r}}_0$, giving $\mathbf{c}^{\text{fine}} = \mathbf{c}^{\text{coarse}} + \widehat{\mathbf{r}}_0$.

After residual diffusion denoising, the generator further refines the camera trajectory using the keyframe set $\mathcal{K}^{\text{plan}}$. The keyframes serve as temporal anchors for smooth tween refinement while preserving bounded fine-grained details from the generated trajectory. To do this, we introduce an intent-conditioned residual gating mechanism that adaptively controls how much fine-grained detail from the generated trajectory is retained during tween refinement. A linear projection layer g_{gate} predicts framewise modulation parameters $\delta_j, \gamma_j \in \mathbb{R}^{d_C}$ via $[\delta_j; \gamma_j] = g_{\text{gate}}([\mathbf{F}_j; \mathbf{Z}_j^{\text{plan}}])$. These parameters respectively modulate the tween function between keyframes and the amount of fine-grained trajectory detail retained during refinement:

$$\mathbf{c}_j^{\text{tween}} = \left(1 - \frac{\sum_{k=k_\ell+1}^{j}\delta_k}{\sum_{k=k_\ell+1}^{k_{\ell+1}}\delta_k}\right)\mathbf{c}_{k_\ell}^{\text{fine}} + \left(\frac{\sum_{k=k_\ell+1}^{j}\delta_k}{\sum_{k=k_\ell+1}^{k_{\ell+1}}\delta_k}\right)\mathbf{c}_{k_{\ell+1}}^{\text{fine}}, \tag{14}$$

$$\mathbf{c}_j^{\text{out}} = \mathbf{c}_j^{\text{tween}} + \boldsymbol{\gamma}_j \odot \left(\mathbf{c}_j^{\text{fine}} - \mathbf{c}_j^{\text{tween}}\right), \tag{15}$$

where $k_\ell, k_{\ell+1} \in \mathcal{K}^{\text{plan}}$ are adjacent keyframes and $\odot$ denotes channel-wise multiplication. The fine camera parameters are preserved at keyframes to ensure trajectory continuity across adjacent interpolation intervals, while interpolation is prevented from crossing hard cuts.

3.4 INTENT ALIGNER: INTENT RECOVERY AND ALIGNMENT

The intent aligner E_{align} recovers intra-shot intent from $(\mathbf{c}^{\text{out}}, \mathbf{d})$ to provide alignment supervision during generator training. Motivated by the video-based shot attribute analysis (Li et al., 2023b), we extract camera-kinematic and camera-geometry features to recover the intra-shot intent. At each frame, the camera-kinematic features comprise the generated camera parameter and its first- and second-order temporal differences, which provide discrete velocity and acceleration cues in camera parameter space. The camera-geometry features are computed from $(\mathbf{c}^{\text{out}}, \mathbf{d})$ and represent the dancer's position and pose relative to the camera. These features are fed to transformer encoders, whose outputs are fused to predict framewise intra-shot intent attributes and the framewise intent embedding $\mathbf{Z}^{\text{rec}}$, as follows:

$$\mathbf{s}^{\text{rec}}, \mathbf{f}^{\text{rec}}, \mathbf{p}^{\text{rec}}, \mathbf{e}^{\text{rec}}, \mathbf{Z}^{\text{rec}} = E_{\text{align}}(\mathbf{c}^{\text{out}}, \mathbf{d}), \tag{16}$$

where $\mathbf{Z}^{\text{rec}}$ is aligned with $\mathbf{Z}^{\text{plan}}$ in equation 7 after shot pooling. After mean pooling within planned shot boundaries, we recover the intra-shot intent $\mathcal{I}^{\text{rec}}_{\text{intra}} = (\mathbf{s}^{\text{rec}}, \mathbf{f}^{\text{rec}}, \mathbf{p}^{\text{rec}}, \mathbf{e}^{\text{rec}})$.

Alignment enforces intent-level consistency through intra-shot intent agreement, shot-level contrastive alignment, and within-shot temporal consistency. The intra-shot intent alignment loss matches recovered intra-shot intent to the planned counterparts using CE, BCE, and SL1 losses:

$$\mathcal{L}^{\text{align}}_{\text{intra}} = \text{CE}(\mathbf{s}^{\text{rec}}, \mathbf{s}^{\text{plan}}) + \text{CE}(\mathbf{f}^{\text{rec}}, \mathbf{f}^{\text{plan}}) + \text{BCE}(\mathbf{p}^{\text{rec}}, \mathbf{p}^{\text{plan}}) + \text{SL1}(\mathbf{e}^{\text{rec}}, \mathbf{e}^{\text{plan}}). \tag{17}$$

The shot-level contrastive alignment loss $\mathcal{L}^{\text{align}}_{\text{con}}$ uses symmetric InfoNCE (van den Oord et al., 2018), with matching shot-pooled $\mathbf{Z}^{\text{plan}}$ and $\mathbf{Z}^{\text{rec}}$ as positives and other shots as negatives:

$$\mathcal{L}^{\text{align}}_{\text{con}} = -\frac{1}{2N}\sum_{i=1}^{N}\left[\log\frac{\exp(a_{ii})}{\sum_{k=1}^{N}\exp(a_{ik})} + \log\frac{\exp(a_{ii})}{\sum_{k=1}^{N}\exp(a_{ki})}\right], \tag{18}$$

where N is the number of paired shots and $a_{ik} = (\mathbf{Z}^{\text{plan}}_i)^\top \mathbf{Z}^{\text{rec}}_k / (0.1 \cdot \|\mathbf{Z}^{\text{plan}}_i\|_2 \cdot \|\mathbf{Z}^{\text{rec}}_k\|_2)$ denotes cosine similarity scaled by temperature 0.1. The within-shot temporal consistency loss $\mathcal{L}^{\text{align}}_{\text{temp}}$ penalizes frame-to-frame changes in recovered scale, focus, and motion primitive before shot pooling:

$$\mathcal{L}^{\text{align}}_{\text{temp}} = \frac{\sum_{j=2}^{T} m_j^{\text{temp}} \sum_{\mathbf{x}\in\{\mathbf{s},\mathbf{f},\mathbf{p}\}} \|\mathbf{x}^{\text{rec}}_j - \mathbf{x}^{\text{rec}}_{j-1}\|_1}{\max\left(1, \sum_{j=2}^{T} m_j^{\text{temp}}\right)}, \tag{19}$$

The mask m_j^{temp} indicates adjacent frames within the same planned shot, as in Section 3.2. Therefore, the complete intent alignment loss $\mathcal{L}^{\text{align}}$ is defined as:

$$\mathcal{L}^{\text{align}} = \mathcal{L}^{\text{align}}_{\text{intra}} + \mathcal{L}^{\text{align}}_{\text{con}} + \mathcal{L}^{\text{align}}_{\text{temp}}. \tag{20}$$

3.5 CAMERA GENERATOR TRAINING AND INFERENCE

Before camera generator training, we pretrain the intent planner and intent aligner using ground truth intent labels. During intent aligner training, the ground truth camera trajectory is used as input to equation 16 instead of generated camera trajectory. After pretraining, we freeze the intent planner and intent aligner and train the camera generator using diffusion, trajectory reconstruction, shot-structure, dynamics, visibility, and intent alignment losses. The diffusion loss $\mathcal{L}_{\text{diff}}$ supervises residual denoising, and the trajectory reconstruction loss $\mathcal{L}_{\text{traj}}$ matches $\mathbf{c}^{\text{out}}$ to $\mathbf{c}^{\text{gt}}$:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I),\mathbf{r}_\tau,\tau,j}\left[\|\widehat{\mathbf{v}}_{\tau,j} - \mathbf{v}_{\tau,j}\|_2^2\right], \tag{21}$$

$$\mathcal{L}_{\text{traj}} = \text{SL1}\left(\mathbf{c}^{\text{out}}, \mathbf{c}^{\text{gt}}\right). \tag{22}$$

The shot-structure loss $\mathcal{L}_{\text{shot}}$ and dynamics loss $\mathcal{L}_{\text{dyn}}$ are defined as:

$$\mathcal{L}_{\text{shot}} = \text{SL1}(\Delta\mathbf{c}^{\text{out}}, \Delta\mathbf{c}^{\text{gt}}; \mathbf{t}^{\text{gt}}) + \text{SL1}(\mathbf{c}^{\text{tween}}, \widetilde{\mathbf{c}}^{\text{gt}}), \tag{23}$$

$$\mathcal{L}_{\text{dyn}} = \text{SL1}(\Delta\mathbf{c}^{\text{out}}, \Delta\mathbf{c}^{\text{gt}}; \mathbf{m}^{(1)}) + \text{SL1}(\Delta^3\mathbf{c}^{\text{out}}, \Delta^3\mathbf{c}^{\text{gt}}; \mathbf{m}^{(3)}), \tag{24}$$

where $\Delta\mathbf{c}_j = \mathbf{c}_j - \mathbf{c}_{j-1}$ and Δ^3 applies Δ three times to measure camera jerk (i.e., temporal change in acceleration). The binary masks $\mathbf{m}^{(1)}$ and $\mathbf{m}^{(3)}$ select differences that do not cross ground truth

hard cuts, with entries $m_j^{(1)} = 1 - t_j^{\mathrm{gt}}$ and $m_j^{(3)} = (1 - t_j^{\mathrm{gt}})(1 - t_{j-1}^{\mathrm{gt}})(1 - t_{j-2}^{\mathrm{gt}})$, respectively. First- and third-order difference terms are evaluated over $j = 2, \ldots, T$ and $j = 4, \ldots, T$, respectively. $\widetilde{\mathbf{c}}^{\mathrm{gt}}$ is obtained by interpolating between consecutive ground truth keyframes using interpolation phases derived from the ground truth trajectory, without crossing its hard cuts.

We also match output and ground truth dancer visibility during training. A differentiable camera-dance projection maps 3D joints $\mathbf{d}$ into the camera view defined by camera parameters $\mathbf{c}$, giving a framewise visible-joint ratio $\mathbf{r}_{\mathrm{vis}}(\mathbf{c}, \mathbf{d}) \in [0, 1]^T$. Thus, the visibility loss is defined as:

$$\mathcal{L}_{\mathrm{vis}} = \mathrm{SL1}\left(\mathbf{r}_{\mathrm{vis}}(\mathbf{c}^{\mathrm{out}}, \mathbf{d}), \mathbf{r}_{\mathrm{vis}}(\mathbf{c}^{\mathrm{gt}}, \mathbf{d})\right). \tag{25}$$

The generator training objective $\mathcal{L}_{\mathrm{gen}}$ combines camera generation losses with intent alignment loss:

$$\mathcal{L}_{\mathrm{gen}} = \mathcal{L}^{\mathrm{align}} + \mathcal{L}_{\mathrm{diff}} + \mathcal{L}_{\mathrm{traj}} + \mathcal{L}_{\mathrm{shot}} + \mathcal{L}_{\mathrm{dyn}} + \mathcal{L}_{\mathrm{vis}}. \tag{26}$$

Additional method and training details are provided in Appendix A.

4 Experiments

4.1 Implementation details

Datasets. Our experiments are conducted on the DCM dataset (Wang et al., 2024a), which provides paired music, dance motion, and animator-authored camera trajectories. For a fair comparison, we use the train and test splits provided with the original dataset. To obtain ground truth cinematographic intent from DCM, we employ rule-based intent extraction methods (Jiang et al., 2024; Wang et al., 2024b). We additionally obtain ground truth keyframes using path simplification methods (Ramer, 1972; Douglas & Peucker, 1973). Note that, because these rule-based extraction procedures are non-differentiable, we introduce the intent aligner to establish a differentiable feedback loop during camera generator training.

Evaluation metrics. Following the evaluation protocol of DCM (Wang et al., 2024a), we assess the synthesized camera trajectories in terms of motion quality, diversity, and dancer fidelity. To evaluate quality, we measure the Fréchet inception distance (FID) (Heusel et al., 2017) between generated and test-set trajectories in both kinetic and shot feature spaces, reported as FID_k and FID_s. Diversity is evaluated by computing the mean pairwise Euclidean distance of generated samples in the corresponding kinetic and shot feature spaces, denoted as Dist_k and Dist_s. We further assess dancer fidelity using dancer missing rate (DMR) and limbs capture difference (LCD). DMR quantifies how often the dancer is outside the camera view, whereas LCD measures the discrepancy in visible body parts between synthesized and ground truth camera trajectories.

To evaluate the intent planner and intent aligner, we measure prediction performance for both intra-shot and inter-shot intents. Inter-shot intent prediction is evaluated using the F1 score for shot-boundary detection, allowing a tolerance of up to three frames. For intra-shot intent prediction, we use F1 scores for shot scale and visual focus, mean average precision (mAP) for camera motion primitives, and mean absolute error (MAE) for camera motion energy.

4.2 Evaluation results

Quantitative comparison. Table 1 compares IITL with state-of-the-art dance-camera synthesis methods, including DanceCamera3D (Wang et al., 2024a), DanceCamAnimator (Wang et al., 2024b), and offline TemMEGA (Xu et al., 2025). These results demonstrate that IITL consistently outperforms the other methods overall. In particular, IITL achieves the lowest FID_k and FID_s scores and the highest Dist_s, indicating closer agreement with the ground truth camera motion and framing distributions while maintaining greater framing diversity. Although IITL yields a lower Dist_k than DanceCamAnimator and TemMEGA, it achieves the best FID_k and substantially higher Dist_s. This indicates that IITL does not maximize raw kinetic variation but instead generates camera motions that better match the ground truth kinetic distribution while maintaining diverse shot-level compositions.

To assess the perceptual quality of IITL, we conduct a user study comparing videos generated by IITL, DanceCamera3D, DanceCamAnimator, and the ground truth. TemMEGA is excluded from

Table 1: Quantitative comparison of IITL with state-of-the-art dance-camera synthesis methods on the DCM dataset. IITL win rate is the percentage of pairwise preferences for IITL against the method in each row, with standard error.

	Quality		Diversity		Dancer fidelity		User Study
Method	$\text{FID}_k \downarrow$	$\text{FID}_s \downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_s \uparrow$	DMR $\downarrow$	LCD $\downarrow$	IITL Win Rate (%)$\uparrow$
Ground truth	–	–	3.275	1.731	0.0014	–	41.7 ± 3.1
DanceCamera3D (Wang et al., 2024a)	3.749	0.280	1.631	1.326	0.0025	0.147	79.6 ± 2.6
DanceCamAnimator (Wang et al., 2024b)	3.453	0.268	**3.140**	1.293	0.0022	0.152	70.9 ± 3.0
TemMEGA (offline) (Xu et al., 2025)	3.237	0.255	1.961	1.347	**0.0020**	**0.141**	–
IITL (ours)	**3.181**	**0.103**	1.872	**1.460**	0.0021	0.145	–

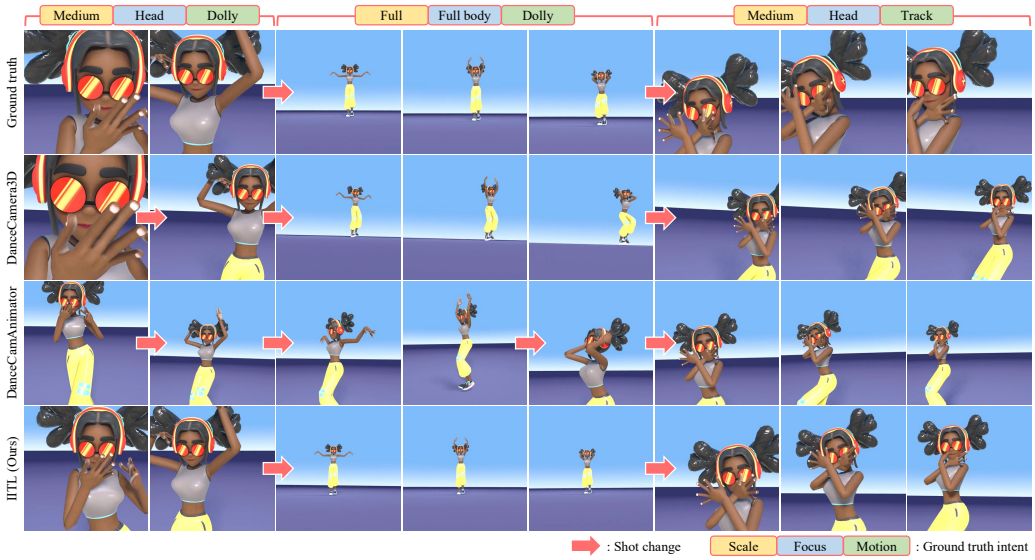

Figure 3: Qualitative comparison of IITL with state-of-the-art dance-camera synthesis methods on the DCM dataset. Red arrows indicate shot changes and labels show intra-shot ground truth intent: scale, visual focus, and camera motion primitive.

this evaluation because neither its source code nor pretrained model is publicly available. We randomly select 10 synchronized music–dance–camera sequences from the test set and render all methods under identical input and rendering conditions. For each sequence, the IITL result is paired separately with the outputs of DanceCamera3D, DanceCamAnimator, and the ground truth, resulting in 30 pairwise comparisons. Twenty-three participants evaluate all pairs in randomized order and choose the video whose camera motion better matches the music and presents the dance more expressively. Table 1 summarizes the pairwise preference rates for IITL together with their standard errors. The result shows that IITL is preferred over DanceCamera3D and DanceCamAnimator in 79.6% and 70.9% of the comparisons, respectively, with participants frequently noting visible shaking and unnatural camera rotations in the baseline results. Moreover, although the ground truth videos contain well-designed camera shots, IITL is preferred in 41.7% of the comparisons against the ground truth, suggesting that the synthesized results are often perceptually competitive with animator-designed camera work. These results indicate that IITL produces smooth camera motion and diverse shot transitions that closely resemble those observed in the ground truth sequences.

Qualitative comparison. Fig. 3 demonstrates the qualitative comparison of our IITL with DanceCam3D and DanceCamAnimator, and ground truth. For an accurate comparison, ground truth intent is denoted. The result shows that DanceCamera3D and DanceCamAnimator introduce additional shot changes and generate videos that are poorly aligned with the ground truth intent. In contrast, IITL not only closely follows the ground truth shot changes but also synthesizes videos that are semantically well aligned with the ground truth intent. Additional qualitative comparisons are provided in the supplementary video.

Table 2: Evaluation of the **(a)** intent planner and **(b)** intent aligner with different backbone networks.

(a) Intent planner	Input	B-F1 ↑	Scale F1 ↑	Focus F1 ↑	Prim. mAP ↑	Energy MAE ↓
AtomicDance (Cai et al., 2026)	Music	–	0.2617	0.2416	0.4509	0.1693
MotionBERT (Zhu et al., 2023)	Dance	–	0.2886	0.2990	0.4646	0.1708
DanceMVP (Zhong & Demiris, 2024)	Music+dance	0.0478	0.3000	0.4064	**0.5800**	0.1692
UniMuMo (Yang et al., 2025)	Music+dance	–	0.3213	**0.4140**	0.5618	0.1681
Intent planner (ours)	Music+dance	**0.1663**	**0.3364**	0.3821	0.5702	**0.1679**

(b) Intent aligner	Scale F1 ↑	Focus F1 ↑	Prim. mAP ↑	Energy MAE ↓
ASFormer (Yi et al., 2021)	0.452	0.618	0.606	0.159
MS-TCN++ (Li et al., 2023a)	0.459	0.604	0.611	0.162
Intent aligner (ours)	**0.492**	**0.664**	**0.631**	**0.155**

Table 3: Ablation study on intent conditioning and the loop mechanism.

	Quality		Diversity		Dancer fidelity	
Method	$\mathrm{FID}_k \downarrow$	$\mathrm{FID}_s \downarrow$	$\mathrm{Dist}_k \uparrow$	$\mathrm{Dist}_s \uparrow$	DMR ↓	LCD ↓
Direct synthesis	7.796	0.125	0.625	1.401	0.0026	0.173
Explicit intent conditioning (Open-loop)	3.204	0.107	1.793	**1.468**	0.0073	0.181
IITL w/o $\mathcal{L}_{\mathrm{con}}^{\mathrm{align}}$ (Closed-loop)	3.227	0.106	1.788	1.449	0.0055	0.177
IITL (Closed-loop)	**3.181**	**0.103**	**1.872**	1.460	**0.0021**	**0.145**

Intent planner and intent aligner evaluation. To validate the effectiveness of the network architectures used for our intent planner and intent aligner, we conduct comparative experiments in Tables 2(a) and 2(b), respectively, by replacing each module with alternative backbone networks. For a fair comparison, all backbone models are retrained on the DCM dataset before evaluation. The results show that the proposed architectures for the intent planner and intent aligner consistently achieve better overall intent prediction performance than the alternative backbones.

Ablation studies. We conduct an ablation study to evaluate the effectiveness of the proposed IITL framework, which explicitly encodes intent and applies a closed-loop mechanism during camera synthesis. As shown in Table 3, we compare following formulations: direct synthesis without explicit intent encoding, synthesis with explicit intent conditioning, and the full loop framework with the intent aligner. We further ablate the proposed shot-level contrastive alignment loss, $\mathcal{L}_{\mathrm{con}}^{\mathrm{align}}$, to assess its contribution within the closed-loop formulation. The result shows that explicit intent conditioning alone improves both quality and diversity metrics over direct synthesis, but degrades DMR and LCD, indicating that conditioning by itself does not necessarily improve dancer fidelity. In contrast, applying the loop with intent aligner improves dancer fidelity while largely preserving the quality and diversity performance. Furthermore, adding the contrastive alignment loss yields additional overall performance gains. These results support the effectiveness of combining explicit intent conditioning with closed-loop framework to improve both trajectory quality and dancer fidelity.

5 Conclusion

We presented IITL, a closed-loop framework for music-and-dance-driven 3D camera synthesis through *plan*, *generate*, and *align*. By explicitly modeling cinematographic intent, IITL enables camera trajectories to better reflect high-level cinematographic decisions and provides shot-level interpretability. In addition, we introduce a closed-loop framework that aligns the intent recovered from the generated camera trajectories with the planned intent during training. This closed-loop formulation encourages the model to capture high-level cinematographic decisions beyond low-level motion plausibility, enabling camera movements that more faithfully reflect the intended shot structure, visual emphasis, and dynamic style. Extensive experiments demonstrate the effectiveness of IITL in both objective and subjective evaluations, while ablation studies further validate the benefits of explicit intent conditioning and the proposed closed-loop framework. We believe that the proposed method has strong practical potential in applications such as film and music video production. In future work, we plan to extend our framework to multi-dancer scenes by addressing group composition, attention shifts, and interactions among multiple dancers.

AI USE STATEMENT

In this work, we used generative AI tools, including OpenAI ChatGPT and Codex, to assist with experiment design and method implementation. We have not used these tools to generate experimental data or mathematical proofs. Additionally, we used them for code debugging, literature review, and manuscript editing. We have reviewed all AI-assisted work. Specifically, all AI-assisted code, results, citations, and text were reviewed and verified by the authors. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Xinhao Cai, Yixuan Sun, Minghang Zheng, Qingchao Chen, Xin Jin, Song-chun Zhu, and Yang Liu. Music-to-dance generation via atomic movements. *arXiv preprint arXiv:2607.13978*, 2026. URL https://arxiv.org/abs/2607.13978.

Marc Christie, Rumesh Machap, Jean-Marie Normand, Patrick Olivier, and Jonathan Pickering. Virtual camera planning: A survey. In *Smart Graphics*, volume 3638 of *Lecture Notes in Computer Science*, pp. 40–52. Springer, 2005. doi: 10.1007/11536482_4. URL https://doi.org/10.1007/11536482_4.

Robin Courant, Nicolas Dufour, Xi Wang, Marc Christie, and Vicky Kalogeiton. E.T. the exceptional trajectories: Text-to-camera-trajectory generation with character awareness. In *Computer Vision – ECCV 2024*, volume 15062 of *Lecture Notes in Computer Science*, pp. 464–480. Springer, 2025. doi: 10.1007/978-3-031-73235-5_26. URL https://doi.org/10.1007/978-3-031-73235-5_26.

Zahra Dehghanian, Pouya Ardehkhani, Amir Vahedi, Hamid Beigy, and Hamid R. Rabiee. Camera trajectory generation: A comprehensive survey of methods, metrics, and future directions. *Computer Vision and Image Understanding*, 271:104872, September 2026. doi: 10.1016/j.cviu.2026.104872. URL https://doi.org/10.1016/j.cviu.2026.104872.

David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973. doi: 10.3138/FM57-6770-U75U-7727. URL https://doi.org/10.3138/FM57-6770-U75U-7727.

Quentin Galvane, Rémi Ronfard, Christophe Lino, and Marc Christie. Continuity editing for 3D animation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. URL https://ojs.aaai.org/index.php/AAAI/article/view/9288.

Li-wei He, Michael F. Cohen, and David H. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 217–224, 1996. doi: 10.1145/237170.237259. URL https://doi.org/10.1145/237170.237259.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, pp. 6626–6637, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Neville Hogan and Dagmar Sternad. Sensitivity of smoothness measures to movement duration, amplitude, and arrests. *Journal of Motor Behavior*, 41(6):529–534, 2009. doi: 10.3200/35-09-004-RC.

Hongda Jiang, Xi Wang, Marc Christie, Libin Liu, and Baoquan Chen. Cinematographic camera diffusion model. *Computer Graphics Forum*, 43(2):e15055, 2024. doi: 10.1111/cgf.15055. URL https://doi.org/10.1111/cgf.15055.

Xuekun Jiang, Libiao Jin, Anyi Rao, Linning Xu, and Dahua Lin. Jointly learning the attributes and composition of shots for boundary detection in videos. *IEEE Transactions on Multimedia*, 24: 3049–3059, 2022. doi: 10.1109/TMM.2021.3092143. URL https://doi.org/10.1109/TMM.2021.3092143.

Kyungjune Lee, Youngjin Shin, Jungwoo Huh, and Sanghoon Lee. You only touch once: One-touch system for personalized 3D music video generation. In *Proceedings of the 2025 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 2259–2263, October 2025. doi: 10.1109/APSIPAASC65261.2025.11249334. URL https://doi.org/10.1109/APSIPAASC65261.2025.11249334.

Kyungjune Lee, Beom Kwon, and Sanghoon Lee. Camera trajectory planning in immersive 3-D human motion-based entertainment contents. *IEEE Access*, 14:50779–50790, 2026. doi: 10.1109/ACCESS.2026.3676487. URL https://doi.org/10.1109/ACCESS.2026.3676487.

Mengtian Li, Yuwei Lu, Feifei Li, Chenqi Gan, Zhifeng Xie, and Xi Wang. VERTIGO: Visual preference optimization for cinematic camera trajectory generation. *arXiv preprint arXiv:2604.02467*, 2026. doi: 10.48550/arXiv.2604.02467. URL https://arxiv.org/abs/2604.02467.

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. AI choreographer: Music conditioned 3D dance generation with AIST++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13401–13412, October 2021. doi: 10.1109/ICCV48922.2021.01315. URL https://openaccess.thecvf.com/content/ICCV2021/html/Li_AI_Choreographer_Music_Conditioned_3D_Dance_Generation_With_AIST_ICCV_2021_paper.html.

Shi-Jie Li, Yazan Abu Farha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. MS-TCN++: Multi-stage temporal convolutional network for action segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6647–6658, 2023a. doi: 10.1109/TPAMI.2020.3021756. URL https://doi.org/10.1109/TPAMI.2020.3021756.

Xinyang Li, Zhangyu Lai, Linning Xu, Yansong Qu, Liujuan Cao, Shengchuan Zhang, Bo Dai, and Rongrong Ji. Director3D: Real-world camera trajectory and 3D scene generation from text. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2391. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/89566c18d5b3e9836e8e16fde010b41d-Abstract-Conference.html.

Yuzhi Li, Feng Tian, Haojun Xu, and Tianfeng Lu. Toward unified and quantitative cinematic shot attribute analysis. *Electronics*, 12(19):4174, 2023b. doi: 10.3390/electronics12194174. URL https://www.mdpi.com/2079-9292/12/19/4174.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, October 2017. doi: 10.1109/ICCV.2017.324. URL https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html.

Alan Litteneker and Demetri Terzopoulos. Virtual cinematography using optimization and temporal smoothing. In *Proceedings of the 10th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2017. doi: 10.1145/3136457.3136467. URL https://doi.org/10.1145/3136457.3136467.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in Python. In *Proceedings of the 14th Python in Science Conference*, pp. 18–24, 2015. doi: 10.25080/Majora-7b98e3ed-003. URL https://doi.org/10.25080/Majora-7b98e3ed-003.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. In *Advances in Neural Information Processing Systems*, volume 38, pp. 100092–100118, 2025. doi: 10.52202/085713-3345. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/904e89bb4e632e75fb47f093b620b257-Abstract-Conference.html.

Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972. doi: 10.1016/S0146-664X(72)80017-0. URL https://doi.org/10.1016/S0146-664X(72)80017-0.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Shuyao Shang, Zhengyang Shan, Guangxing Liu, LunQian Wang, XingHua Wang, Zekai Zhang, and Jinglin Zhang. ResDiff: Combining CNN and diffusion model for image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8975–8983, 2024. doi: 10.1609/aaai.v38i8.28746. URL https://doi.org/10.1609/aaai.v38i8.28746.

Joan Solà, Jeremie Deray, and Dinesh Atchuthan. A micro Lie theory for state estimation in robotics. *arXiv preprint arXiv:1812.01537*, 2018. URL https://arxiv.org/abs/1812.01537.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit H. Bermano, and Michiel van de Panne. CLoSD: Closing the loop between simulation and diffusion for multi-task character control. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZISppZSTv.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Zixuan Wang, Jia Jia, Shikun Sun, Haozhe Wu, Rong Han, Zhenyu Li, Di Tang, Jiaqing Zhou, and Jiebo Luo. DanceCamera3D: 3D camera movement synthesis with music and dance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7892–7901, June 2024a. doi: 10.1109/CVPR52733.2024.00754. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wang_DanceCamera3D_3D_Camera_Movement_Synthesis_with_Music_and_Dance_CVPR_2024_paper.html.

Zixuan Wang, Jiayi Li, Xiaoyu Qin, Shikun Sun, Songtao Zhou, Jia Jia, and Jiebo Luo. DanceCamAnimator: Keyframe-based controllable 3D dance camera synthesis. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10200–10209, October 2024b. doi: 10.1145/3664647.3680980. URL https://doi.org/10.1145/3664647.3680980.

Peter D. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, 1967. doi: 10.1109/TAU.1967.1161901.

Chenghao Xu, Guangtao Lyu, Jiexi Yan, Muli Yang, and Cheng Deng. Smooth and flexible camera movement synthesis via temporal masked generative modeling.

In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/13becb43369ef8c50eb00da399bd1488-Abstract-Conference.html.

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, 2025. doi: 10.1609/aaai.v39i24.34752. URL https://ojs.aaai.org/index.php/AAAI/article/view/34752.

Fangqiu Yi, Hongyu Wen, and Tingting Jiang. ASFormer: Transformer for action segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 236, 2021. doi: 10.5244/C.35.49. URL https://www.bmva-archive.org.uk/bmvc/2021/assets/papers/0183.pdf.

Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (PIP): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13167–13178, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Yi_Physical_Inertial_Poser_PIP_Physics-Aware_Real-Time_Human_Motion_Tracking_From_CVPR_2022_paper.html.

Mengchen Zhang, Tong Wu, Jing Tan, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. GenDoP: Auto-regressive camera trajectory generation as a director of photography. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18229–18239, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_GenDoP_Auto-regressive_Camera_Trajectory_Generation_as_a_Director_of_Photography_ICCV_2025_paper.html.

Yun Zhong and Yiannis Demiris. DanceMVP: Self-supervised learning for multi-task primitive-based dance performance assessment via transformer text prompting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10270–10278, 2024. doi: 10.1609/aaai.v38i9.28893. URL https://ojs.aaai.org/index.php/AAAI/article/view/28893.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2223–2232, October 2017. doi: 10.1109/ICCV.2017.244. URL https://openaccess.thecvf.com/content_ICCV_2017/html/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.html.

Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. MotionBERT: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15085–15099, October 2023. doi: 10.1109/ICCV51070.2023.01385. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhu_MotionBERT_A_Unified_Perspective_on_Learning_Human_Motion_Representations_ICCV_2023_paper.html.

A APPENDIX

The architectural descriptions follow the *plan–generate–align* order of Section 3. This appendix supplies architectural details, decoding rules, target construction, training and inference details, and additional evaluation of ground truth intent fidelity, shot changes, camera rotations, within-shot dynamics, and loss and intent aligner ablations.

A.1 INTENT PLANNER

Multimodal encoding and history representation. Within the intent planner, the music encoder receives the 35-D music features $\mathbf{m}$ extracted using Librosa (McFee et al., 2015), following the feature configuration of FACT (Li et al., 2021). The dance encoder receives $\mathrm{vec}(\mathbf{d}) \in \mathbb{R}^{T\times 180}$, where vec flattens the 60 joint positions into 180 coordinates per frame. Each is a four-layer temporal transformer encoder (Vaswani et al., 2017). Their outputs are fused by the feed-forward layer

$$\mathbf{F} = \mathrm{LN}\big(\mathrm{GELU}\big([E_{\mathrm{music}}(\mathbf{m}); E_{\mathrm{dance}}(\mathrm{vec}(\mathbf{d}))]\mathbf{W}_{\mathrm{fuse}}^{\top}\big)\big) \in \mathbb{R}^{T\times 256}. \tag{27}$$

Here $[\cdot;\cdot]$ denotes concatenation along the feature dimension, $\mathbf{W}_{\mathrm{fuse}}$ is the learned projection matrix applied to each frame, GELU is the Gaussian error linear unit activation, and LN denotes layer normalization. E_{music} and E_{dance} denote the two temporal encoders. Framewise features are stored as rows. The same music and dance encoders and fusion layer produce $\mathbf{F}^{\mathrm{hist}} \in \mathbb{R}^{T^{\mathrm{hist}}\times 256}$ from the preceding history window. If fewer than $T^{\mathrm{hist}} = 32$ frames are available, the available frames are right-aligned after left zero-padding. The current window never enters the history window.

The intent planner E_{plan} first produces current hidden states $\mathbf{H}$ from $\mathbf{F}$ using a three-layer transformer encoder. A two-layer sigmoid gated cross-attention block uses these states as queries and $\mathbf{F}^{\mathrm{hist}}$ as keys and values:

$$\begin{aligned}
\mathbf{H}^{\mathrm{mem}} &= \mathrm{CrossAttn}(\mathbf{H}, \mathbf{F}^{\mathrm{hist}}, \mathbf{F}^{\mathrm{hist}}),\\
\mathbf{g}^{\mathrm{mem}} &= \sigma\big([\mathbf{H}; \mathbf{H}^{\mathrm{mem}}]\mathbf{W}_{\mathrm{gate}}^{\top}\big),\\
\mathbf{H}^{\mathrm{plan}} &= \mathrm{LN}(\mathbf{H} + \mathbf{g}^{\mathrm{mem}} \odot (\mathbf{H}^{\mathrm{mem}} - \mathbf{H})).
\end{aligned} \tag{28}$$

CrossAttn denotes the cross-attention block, and $\mathbf{H}, \mathbf{H}^{\mathrm{mem}}, \mathbf{H}^{\mathrm{plan}} \in \mathbb{R}^{T\times 256}$ denote the states before cross-attention, after cross-attention, and after gating. $\mathbf{W}_{\mathrm{gate}}$ is the learned gate projection matrix, $\mathbf{g}^{\mathrm{mem}} \in [0,1]^{T\times 256}$ contains the numerical gate values, σ applies the sigmoid to each entry, and $\odot$ denotes channel-wise multiplication. For a sample without preceding history, we set $\mathbf{H}^{\mathrm{plan}} = \mathbf{H}$.

Intent and keyframe decoding. The seven task-specific heads of E_{plan} operate on $\mathbf{H}^{\mathrm{plan}}$. Six intent heads produce a binary shot-boundary logit, 4-way framing-scale and visual-focus logits, 7-D multi-label camera motion primitive logits, a scalar camera motion energy logit, and a binary transition-type logit. The remaining head produces a binary keyframe logit. Sigmoid converts the binary logits to probabilities. Before decoding, $\mathbf{b}^{\mathrm{plan}}$, $\mathbf{t}^{\mathrm{plan}}$, and $\mathbf{k}^{\mathrm{plan}}$ denote framewise probabilities in $[0,1]^T$, as in the intent planner losses. After decoding, the same symbols denote binary masks. Local maxima above 0.5 in boundary probabilities determine the shot segmentation, with the first frame designated as a shot start. The decoded mask $\mathbf{b}^{\mathrm{plan}}$ marks the start and end frames of each shot, as in Section 3.1. At noninitial shot starts, transition probabilities at least 0.5 indicate hard cuts, and lower probabilities indicate continuous transitions. Softmax gives framewise scale and focus probabilities, while sigmoid gives primitive probabilities and camera motion energy in $[0,1]$. These framewise predictions are mean pooled within each decoded shot, and the resulting attributes are shared by the shot's frames.

Local maxima above 0.5 in the keyframe probabilities are selected as keyframes. The first and last frames of each decoded shot are also included, and additional keyframes are inserted so that no interval exceeds 60 frames. The resulting ordered set $\mathcal{K}^{\mathrm{plan}} = \{k_1, \ldots, k_{N_K}\}$ specifies the temporal anchors used for trajectory refinement, where N_K is the number of keyframes and $1 \le k_1 < \cdots < k_{N_K} \le T$. The mask $\mathbf{k}^{\mathrm{plan}}$ marks these frame indices, while $\mathcal{K}^{\mathrm{plan}}$ lists them.

Intent planner targets derived from ground truth. The intent pseudo-labels derived using the fixed geometry- and motion-based rules Q in Section 4.1 provide intent planner supervision. The

mask $\mathbf{b}^{\mathrm{gt}}$ marks ground truth shot starts and ends, while scale, focus, primitive, and energy form $\mathcal{I}^{\mathrm{gt}}_{\mathrm{intra}}$ and remain constant within each ground truth shot. At each noninitial shot start, the transition label $\mathbf{t}^{\mathrm{gt}}$ identifies a hard cut when the camera discontinuity exceeds a fixed threshold. Together, $\mathbf{b}^{\mathrm{gt}}$ and $\mathbf{t}^{\mathrm{gt}}$ form $\mathcal{I}^{\mathrm{gt}}_{\mathrm{inter}}$. The keyframe set $\mathcal{K}^{\mathrm{gt}}$ is obtained by applying Ramer–Douglas–Peucker simplification (Ramer, 1972; Douglas & Peucker, 1973) to a smoothed, normalized ground truth camera path. The first and last frames of each ground truth shot and any additional frames required to keep keyframe intervals within 60 frames are then included. The binary mask $\mathbf{k}^{\mathrm{gt}} \in \{0,1\}^T$ marks the resulting keyframe locations for intent planner supervision. Between consecutive ground truth keyframes that do not cross a hard cut, cumulative camera-parameter distance normalized to $[0,1]$ defines the ground truth interpolation phases. Interpolating the endpoint camera parameters with these phases produces the ground truth tween path $\widetilde{\mathbf{c}}^{\mathrm{gt}}$ used in equation 23.

A.2 Camera generator

Intent encoding and coarse trajectory. As defined in equation 7, the generator's transformer encoder E_{int} maps the planned boundary and shot-pooled scale, focus, primitive, and energy predictions to framewise intent embeddings $\mathbf{Z}^{\mathrm{plan}}_j$. Neither transition-type predictions nor keyframe probabilities enter this embedding. Decoded hard cuts govern shot-level updates and accumulation resets, while $\mathcal{K}^{\mathrm{plan}}$ provides refinement locations.

In equation 10, the probability $p^{\mathrm{plan}}_{i,r}$ for primitive r in shot i is divided by the sum across the seven primitives. The weighted sum of the trainable bases $\mathbf{V}_r \in \mathbb{R}^{d_C}$ is added to $g_{\mathrm{frame}}(\mathbf{Z}^{\mathrm{plan}}_j)$ and scaled by e^{plan}_i to form the frame-level camera-parameter increment $\mathbf{c}^{\mathrm{frame}}_j \in \mathbb{R}^{d_C}$. Both g_{frame} and g_{shot} are linear layers with bias mapping intent embeddings into camera-parameter space. The linear projection g_{init} maps the temporal mean of $\mathbf{F}$ from 256 to d_C dimensions, with bias. The shot-level camera-parameter vector $\mathbf{c}^{\mathrm{shot}}_i \in \mathbb{R}^{d_C}$, shared across shot i, is initialized and updated as

$$\begin{aligned}\overline{\mathbf{F}} &= \frac{1}{T}\sum_{k=1}^{T}\mathbf{F}_k, \qquad \mathbf{c}^{\mathrm{shot}}_1 = g_{\mathrm{init}}(\overline{\mathbf{F}}),\\ \mathbf{c}^{\mathrm{shot}}_i &= \mathbf{c}^{\mathrm{shot}}_{i-1} + t^{\mathrm{plan}}_j g_{\mathrm{shot}}(\mathbf{Z}^{\mathrm{plan}}_j), \qquad i>1,\ b^{\mathrm{plan}}_j = 1.\end{aligned} \tag{29}$$

Here j is the first frame of shot i in the update, and boundary and transition conditions use decoded binary values. In equation 11, the frame-level camera-parameter increments are accumulated from j_{cut} through j and added to $\mathbf{c}^{\mathrm{shot}}_i$ to obtain the camera-parameter vector $\mathbf{c}^{\mathrm{coarse}}_j \in \mathbb{R}^{d_C}$ at frame j. Stacking these vectors gives the coarse trajectory $\mathbf{c}^{\mathrm{coarse}} \in \mathbb{R}^{T\times d_C}$. Hard cuts update the shared shot-level camera-parameter vector and restart accumulation. Continuous transitions retain both. The current-frame increment contributes in both cases. These generator components are trained jointly through $\mathcal{L}_{\mathrm{gen}}$.

Fine trajectory and residual diffusion. For diffusion step $\tau \in \{1,\dots,N_d\}$, where N_d is the number of training diffusion steps, the forward process and v-parameterization (Ho et al., 2020; Salimans & Ho, 2022) give

$$\begin{aligned}\bar{\alpha}_\tau &= \prod_{\tau'=1}^{\tau}(1-\beta_{\tau'}),\\ \mathbf{r}_\tau &= \sqrt{\bar{\alpha}_\tau}\mathbf{r}_0 + \sqrt{1-\bar{\alpha}_\tau}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\\ \mathbf{v}_\tau &= \sqrt{\bar{\alpha}_\tau}\boldsymbol{\epsilon} - \sqrt{1-\bar{\alpha}_\tau}\mathbf{r}_0,\\ \widehat{\mathbf{r}}_{0,\tau} &= \sqrt{\bar{\alpha}_\tau}\mathbf{r}_\tau - \sqrt{1-\bar{\alpha}_\tau}\widehat{\mathbf{v}}_\tau,\end{aligned} \tag{30}$$

where β_τ is the noise variance at step τ, $\mathbf{I}$ is the identity covariance, and $\widehat{\mathbf{r}}_{0,\tau}$ is the clean-residual estimate obtained from $\widehat{\mathbf{v}}_\tau$. We set $\bar{\alpha}_0 = 1$ for the clean endpoint. For consecutive selected inference steps $\tau > \tau' \geq 0$, deterministic DDIM sampling uses

$$\begin{aligned}\widehat{\boldsymbol{\epsilon}}_\tau &= \sqrt{1-\bar{\alpha}_\tau}\mathbf{r}_\tau + \sqrt{\bar{\alpha}_\tau}\widehat{\mathbf{v}}_\tau,\\ \mathbf{r}_{\tau'} &= \sqrt{\bar{\alpha}_{\tau'}}\widehat{\mathbf{r}}_{0,\tau} + \sqrt{1-\bar{\alpha}_{\tau'}}\widehat{\boldsymbol{\epsilon}}_\tau.\end{aligned} \tag{31}$$

Here $\widehat{\epsilon}_\tau$ is the noise estimate at step τ. The generator's six-layer transformer denoising function v_θ, with parameters θ, receives the inputs specified in Section 3.3, including $\mathbf{b}^{\text{plan}}$ and framewise camera motion energy. These values condition diffusion, while decoded boundaries determine shot pooling and only hard cuts reset coarse integration. Training uses $N_d = 1000$ steps, and inference uses 20-step deterministic DDIM sampling (Song et al., 2021). The final clean estimate gives $\mathbf{c}^{\text{fine}} = \mathbf{c}^{\text{coarse}} + \widehat{\mathbf{r}}_0$.

Trajectory refinement. The camera generator then refines the fine trajectory. The intent-conditioned residual gating uses the linear projection g_{gate} on concatenated music-dance features and planned intent embeddings to predict two framewise modulation vectors:

$$[\delta_j; \gamma_j] = g_{\text{gate}}([\mathbf{F}_j; \mathbf{Z}_j^{\text{plan}}]), \qquad \delta_j, \gamma_j \in \mathbb{R}^{d_C}. \tag{32}$$

Here, δ_j modulates the tween function and γ_j modulates the retained fine-trajectory detail. The cumulative ratios and products with endpoint camera parameters in the tween function are evaluated channel-wise. For nonzero cumulative denominators, the ratios equal 0 and 1 at the first and last keyframes, respectively.

The refinement in equation 15 preserves the fine camera parameters at each keyframe:

$$\mathbf{c}_{k_\ell}^{\text{out}} = \mathbf{c}_{k_\ell}^{\text{fine}}. \tag{33}$$

This preserves trajectory continuity across adjacent interpolation intervals, while no interpolation crosses a hard cut. The ground truth tween function described in Section A.1 constructs only the ground truth tween path during training.

A.3 Intent aligner

The intent aligner E_{align} receives the camera-kinematic input

$$\mathbf{x}_j^{\text{cam}} = [\mathbf{c}_j^{\text{out}}; \Delta\mathbf{c}_j^{\text{out}}; \Delta^2\mathbf{c}_j^{\text{out}}] \in \mathbb{R}^{3d_C} \tag{34}$$

where $\Delta\mathbf{c}_j^{\text{out}} = \mathbf{c}_j^{\text{out}} - \mathbf{c}_{j-1}^{\text{out}}$ and $\Delta^2\mathbf{c}_j^{\text{out}} = \Delta\mathbf{c}_j^{\text{out}} - \Delta\mathbf{c}_{j-1}^{\text{out}}$, with unavailable initial differences padded. One transformer encoder maps $\mathbf{x}_j^{\text{cam}}$ to framewise feature vectors encoding camera configuration and motion. The differences provide discrete velocity and acceleration cues in camera-parameter space. Framing scale and visual focus also depend on camera–dancer geometry. Another transformer encoder processes eight framewise camera-geometry features computed from $(\mathbf{c}_j^{\text{out}}, \mathbf{d}_j)$: visible-joint ratio, screen occupancy, projected dancer height, a height feature for visual focus, relative camera distance, horizontal and vertical coordinates of the projected dancer position, and relative camera height. Its framewise feature vectors encode camera–dancer spatial relations and the dancer's framing. The encoded features are concatenated and passed to a shot transformer encoder. The fused features recover framing scale, visual focus, camera motion primitives, and the framewise intent embedding $\mathbf{Z}^{\text{rec}} \in \mathbb{R}^{T\times 128}$, while camera motion energy is predicted from the first encoder's output alone. During generator training, the framewise attribute predictions are mean pooled within planned shot boundaries, after which $\mathbf{s}^{\text{rec}}, \mathbf{f}^{\text{rec}}, \mathbf{p}^{\text{rec}}, \mathbf{e}^{\text{rec}}$ denote shot-level attributes, as in Section 3.4. The complete intent aligner input is $(\mathbf{c}^{\text{out}}, \mathbf{d})$ as defined in equation 16. During intent aligner pretraining, $\mathbf{c}^{\text{out}}$ is replaced by $\mathbf{c}^{\text{gt}}$, and the fixed rules Q derive the intent pseudo-labels from the same ground truth pair $(\mathbf{c}^{\text{gt}}, \mathbf{d})$.

During generator training, the planned intra-shot intent is detached. Scale and focus targets use the argmax of shot-pooled predictions, while primitive and energy targets retain the planned probabilities and scalar values. Detached planned boundaries define shot pooling and the temporal mask m_j^{temp}, but are not prediction targets for the intent aligner. The intent embeddings are mean pooled within these shots for contrastive alignment, using at most 128 paired shots. The alignment losses are defined in Section 3.4.

A.4 Camera generator training and inference

The generator training objective in equation 26 combines camera generation losses with intent alignment. We train the three models sequentially. First, the intent aligner is trained from $(\mathbf{c}^{\text{gt}}, \mathbf{d})$ using the intra-shot intent labels derived from ground truth for scale, focus, primitives, and energy described in the Datasets paragraph of Section 4.1. Second, the intent planner, including its music and



Figure 4: Qualitative comparison of IITL, DanceCamera3D (Wang et al., 2024a), and DanceCamAnimator (Wang et al., 2024b) on DCM under identical music-dance inputs and rendering conditions. Red arrows indicate shot changes.

dance encoders and fusion layer, is trained with equation 6. During intent planner training, intra-shot predictions are pooled over ground truth shots. Scale and focus use 0.05 label smoothing. Finally, the intent planner and intent aligner are frozen, while the camera generator, including E_{int}, v_θ, and the residual gating and trajectory refinement operations, is trained with equation 26. Gradients pass through the frozen E_{align} to the camera generator, while the intent planner and intent aligner receive no parameter updates. Generator training uses 150-frame windows, batch size 12, learning rate 10^{-4}, and weight decay 0.01.

At inference, the intent planner encodes music and dance and produces the planned intent and keyframe set. The camera generator computes the planned intent embeddings, constructs the coarse trajectory, obtains $\mathbf{c}^{\text{fine}}$ using 20-step DDIM sampling, and refines it to produce $\mathbf{c}^{\text{out}}$. The intent aligner and training losses are omitted.

A.5 Ground Truth Intent Fidelity and Qualitative Comparisons

Ground truth intent fidelity. For generator g with trajectory $\mathbf{c}^g$, intent $\mathcal{I}^{\text{gt}}$ derived from ground truth and trajectory-inferred intent $\mathcal{I}_g^{\text{traj}} = Q(\mathbf{c}^g, \mathbf{d})$ define ground truth intent fidelity $M_{\text{gt}}(g) = M\big(\mathcal{I}_g^{\text{traj}}, \mathcal{I}^{\text{gt}}\big)$, where M is the corresponding boundary, scale, focus, primitive, or energy metric. Ground truth intent fidelity uses the same fixed rules Q, not the learned intent aligner, to measure agreement with intent derived from ground truth rather than with planned intent. Boundary F1 (B-F1) matches inter-shot boundaries within ± 3 frames. Scale and focus use macro-F1, motion primitives use mAP, and camera motion energy uses MAE. Table 4 evaluates intent recovered from generated camera trajectories using the same fixed rules Q for all methods. IITL leads four of the five ground truth intent metrics: boundary F1, focus F1, primitive mAP, and energy MAE. The higher ground truth B-F1 and ground truth focus F1 support better agreement with ground truth shot changes and visual focus. Although IITL also achieves the lowest FID_s, its ground truth scale F1 is below DanceCamera3D (0.346 vs. 0.360). This distinction shows that distribution-level framing quality and agreement with individual ground truth attributes capture complementary aspects of camera generation. TemMEGA trajectories were unavailable for this evaluation.

Table 4: End-to-end ground truth intent fidelity on DCM, evaluated using the same fixed rules Q for all generators. Best and second-best results are **bold** and underlined.

Method	$FID_s \downarrow$	B-F1 ↑	Scale F1 ↑	Focus F1 ↑	Prim. mAP ↑	Energy MAE ↓
DanceCamera3D (Wang et al., 2024a)	0.280	0.153	**0.360**	0.327	0.446	0.216
DanceCamAnimator (Wang et al., 2024b)	0.268	0.109	0.320	0.324	0.444	0.232
IITL (ours)	**0.103**	**0.166**	0.346	**0.352**	**0.461**	**0.209**



Figure 5: Unnatural camera rotations in baseline results on DCM under identical music-dance inputs and rendering conditions.

Shot changes. Fig. 4 compares corresponding dance frames across ground truth shot changes. The first change, between the second and third columns, introduces an intermediate shot before the second change between the fourth and fifth columns. IITL follows both marked ground truth shot changes, whereas DanceCamera3D and DanceCamAnimator show only the second marked change in this example. At the second change, IITL also produces tighter framing than either baseline, more closely following the head-focused ground truth view. This example illustrates better preservation of the ground truth inter-shot structure and is consistent with the higher ground truth B-F1 in Table 4.

Unnatural camera rotations. Fig. 5 provides a qualitative example of the unnatural camera rotations discussed in the user study in Section 4.2. Compared with the ground truth views, DanceCamera3D and DanceCamAnimator produce large rotations that turn the view sideways or upside down. IITL avoids these extreme orientations in the shown frames.

A.6 Within-shot dynamics

Evaluation protocol. Jerk-based evaluation is used to characterize motion smoothness in prior 3D motion work (Yi et al., 2022), while the choice of normalization affects its dependence on movement amplitude and duration (Hogan & Sternad, 2009). We report normalized translational jerk (NTJ), angular jerk on $\mathrm{SO}(3)$ (AJ), zoom jerk (ZoomJ), and RMS angular oscillation (OscRMS). These are complementary measurements of within-shot camera dynamics rather than a single measure of cinematographic quality. All methods use the same shot boundaries derived from ground truth

and a sampling interval $h = 1/30$ seconds. In this subsection, T denotes the number of frames in the evaluated clip. The definitions below specify our camera-specific normalization, temporal differences, and aggregation. The references motivate jerk-based analysis rather than define these particular evaluation choices.

Translational jerk. Let $\mathbf{x}_j \in \mathbb{R}^3$ denote the camera position at frame j in the evaluated trajectory $\mathbf{c}$, and let $\mathbf{x}_j^{\mathrm{gt}}$ denote its ground truth counterpart. For a scalar or vector signal $\mathbf{u}_j$, we use the backward-difference notation of Section 3:

$$\Delta^3 \mathbf{u}_j = \mathbf{u}_j - 3\mathbf{u}_{j-1} + 3\mathbf{u}_{j-2} - \mathbf{u}_{j-3}, \qquad j = 4, \dots, T. \tag{35}$$

Let d_j^{gt} be the signed ground truth camera-to-target distance, and let ℓ_{ref} be the median of its finite absolute values greater than 10^{-6} within the clip. The framewise normalized translational jerk is

$$J_j^{\mathrm{trans}} = \frac{\|\Delta^3 \mathbf{x}_j\|_2}{h^3 \ell_{\mathrm{ref}}}. \tag{36}$$

The same ℓ_{ref} derived from ground truth is used for all methods on a clip. If no valid distance is available, the implementation uses $\ell_{\mathrm{ref}} = \max\{\mathrm{P}_{90}(\|\mathbf{x}_j^{\mathrm{gt}} - \mathrm{median}_k\, \mathbf{x}_k^{\mathrm{gt}}\|_2), 1\}$, where the position median is taken coordinate-wise and P_{90} denotes the 90th percentile over frames. This normalization uses a shared ground truth length scale without normalizing movement duration, so J_j^{trans} has units of s^{-3} and is not a dimensionless jerk measure.

Angular jerk. Let $\mathbf{a}_{x,j}, \mathbf{a}_{y,j}, \mathbf{a}_{z,j}$ be the camera axes associated with the evaluated trajectory. The stored axes form a left-handed triad, so we negate the look axis and project the resulting matrix onto the group of 3D rotation matrices $\mathrm{SO}(3)$:

$$\begin{aligned} \mathbf{A}_j &= [\mathbf{a}_{x,j}, \mathbf{a}_{y,j}, -\mathbf{a}_{z,j}] = \mathbf{U}_j \mathbf{\Sigma}_j \mathbf{V}_j^\top, \\ \mathbf{R}_j &= \mathbf{U}_j \,\mathrm{diag}\big(1, 1, \det(\mathbf{U}_j \mathbf{V}_j^\top)\big) \mathbf{V}_j^\top. \end{aligned} \tag{37}$$

Here the first line is a singular value decomposition, with orthogonal matrices $\mathbf{U}_j, \mathbf{V}_j$ and a diagonal matrix $\mathbf{\Sigma}_j$ of singular values in descending order. The rotation $\mathbf{R}_j$ maps camera coordinates to world coordinates. Using the rotation logarithm (Solà et al., 2018), we estimate angular velocity in world coordinates and obtain angular jerk from its second-order temporal difference:

$$\begin{aligned} \boldsymbol{\omega}_j &= \frac{\log(\mathbf{R}_j \mathbf{R}_{j-1}^\top)^\vee}{h}, \qquad j = 2, \dots, T, \\ J_j^{\mathrm{ang}} &= \frac{\|\boldsymbol{\omega}_j - 2\boldsymbol{\omega}_{j-1} + \boldsymbol{\omega}_{j-2}\|_2}{h^2}, \qquad j = 4, \dots, T. \end{aligned} \tag{38}$$

$\log(\cdot)$ denotes the matrix logarithm on $\mathrm{SO}(3)$, and $(\cdot)^\vee$ maps a skew-symmetric matrix to its corresponding vector. We use the principal rotation vector in radians. Thus, AJ uses relative rotations rather than differences of Euler angles, and J_j^{ang} has units of $\mathrm{rad\,s}^{-3}$.

Zoom jerk. Let ϕ_j be the FOV after clipping to $[1^\circ, 179^\circ]$ and conversion to radians. The log tangent of the half FOV and its third difference define

$$z_j = \log \tan\left(\frac{\phi_j}{2}\right), \qquad J_j^{\mathrm{zoom}} = \frac{|\Delta^3 z_j|}{h^3}. \tag{39}$$

J_j^{zoom} has units of s^{-3}. The factor h^{-3} in the jerk quantities converts third-order frame differences into estimates of third-order time derivatives.

Boundary exclusion and percentile aggregation. For a clip, let $\mathcal{B}^{\mathrm{gt}}$ contain the noninitial shot-start indices derived from ground truth and used by the evaluator. A jerk sample is retained only when none of its four supporting frames lies within three frames of any such boundary:

$$\begin{aligned} \mathcal{V} = \big\{ j \in \{4, \dots, T\} : |k - b| > 3 \\ \text{for all } k \in \{j-3, \dots, j\},\ b \in \mathcal{B}^{\mathrm{gt}} \big\}. \end{aligned} \tag{40}$$

The initial frame does not introduce an exclusion region. For each metric, we pool the finite retained samples from all test clips and then take a single 95th percentile:

$$\begin{aligned}\mathrm{NTJ} &= \mathrm{P}_{95}\big(\{J^{\mathrm{trans}}_{c,j}\}_{1\le c\le N_{\mathrm{clip}},\, j\in\mathcal{V}_c}\big)\,,\\ \mathrm{AJ} &= \mathrm{P}_{95}\big(\{J^{\mathrm{ang}}_{c,j}\}_{1\le c\le N_{\mathrm{clip}},\, j\in\mathcal{V}_c}\big)\,,\\ \mathrm{ZoomJ} &= \mathrm{P}_{95}\big(\{J^{\mathrm{zoom}}_{c,j}\}_{1\le c\le N_{\mathrm{clip}},\, j\in\mathcal{V}_c}\big)\,.\end{aligned} \tag{41}$$

Here c indexes the $N_{\mathrm{clip}} = 43$ test clips, $\mathcal{V}_c$ is the retained index set for clip c, and $J_{c,j}$ is its corresponding framewise quantity. The operator P_{95} denotes the 95th percentile, and the indexed braces collect samples with repetitions retained. Each percentile is computed over the pooled finite samples using linear interpolation between ordered values, without averaging per-clip percentiles.

Angular oscillation. OscRMS measures band-limited variation of the same world-coordinate angular velocity $\boldsymbol{\omega}_j$. For each ground truth shot, we trim three frames from every edge adjacent to another shot, leaving the outer clip edges untrimmed, and retain angular-velocity intervals whose two endpoints lie inside the trimmed shot. Only shots with at least 90 retained angular-velocity samples are used. For shot i, let n_i be this sample count and $P_i(f)$ the sum of the three coordinate-wise, one-sided power spectral densities estimated using Welch's method (Welch, 1967), where f is frequency in Hz. We use a periodic Hann window of length $\min(256, n_i)$, overlap equal to half the window length rounded down, linear detrending of each window, arithmetic averaging of window spectra, and a sampling rate of 30 Hz. For each shot, let $f_1 < \cdots < f_K$ denote its K frequency bins in $[0.3, 2.5]$ Hz. Trapezoidal integration gives the band power

$$B_i = \sum_{k=1}^{K-1} \frac{P_i(f_k) + P_i(f_{k+1})}{2}(f_{k+1} - f_k). \tag{42}$$

The integration uses only bins inside this band, without interpolating at its endpoints. For each clip c, let $\mathcal{S}_c$ be its eligible shots and let $\mathcal{C}$ be the set of clips with at least one eligible shot. The reported statistic is

$$\mathrm{OscRMS} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\left(\frac{\sum_{i\in\mathcal{S}_c} n_i B_i}{\sum_{i\in\mathcal{S}_c} n_i}\right)^{1/2}. \tag{43}$$

Thus, band powers are weighted by retained sample count within each clip before taking the square root, and the resulting clip values are averaged with equal weight. OscRMS has units of $\mathrm{rad\,s^{-1}}$. Clips without an eligible shot are omitted from this average.

Results. Table 5 reports the within-shot dynamics results. IITL has lower NTJ, AJ, and OscRMS than both synthesized baselines and lower ZoomJ than DanceCamAnimator, but higher ZoomJ than DanceCamera3D. Thus, the measurements support reduced translational and angular jerk while identifying zoom dynamics as a remaining limitation relative to DanceCamera3D. IITL's OscRMS is below that of ground truth, which may indicate over-smoothing rather than improved cinematographic quality. These within-shot measurements exclude boundary neighborhoods and therefore do not evaluate shot-change fidelity itself. TemMEGA trajectories were unavailable for this evaluation.

Table 5: Within-shot trajectory dynamics on DCM. Best and second-best synthesized results are **bold** and underlined.

	Jerk			Oscillation
Method	NTJ ↓	AJ ↓	ZoomJ ↓	OscRMS ↓
Ground truth	78.206	22.091	1.280	0.374
DanceCamera3D (Wang et al., 2024a)	<u>1940.245</u>	<u>1184.488</u>	**284.183**	<u>0.387</u>
DanceCamAnimator (Wang et al., 2024b)	9011.724	4853.383	1794.253	0.606
IITL (ours)	**518.399**	**310.113**	<u>790.978</u>	**0.3211**

A.7 Loss ablation

Experimental settings. Table 6 compares full IITL with seven variants that remove a generator loss group in equation 26 or an alignment term in equation 20. Removing $\mathcal{L}^{\text{align}}$ excludes all three alignment terms, while removing an individual term retains the other two. All variants retain diffusion and trajectory reconstruction losses and planned intent conditioning.

Table 6: Generator loss ablation on DCM. Each variant removes the indicated loss from full IITL. Best and second-best results are **bold** and underlined, with ties indicated at the reported precision.

	Quality		Diversity		Dancer fidelity	
Method	$\text{FID}_k \downarrow$	$\text{FID}_s \downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_s \uparrow$	DMR $\downarrow$	LCD $\downarrow$
IITL (ours)	**3.181**	**0.103**	**1.872**	1.460	**0.0021**	**0.145**
w/o $\mathcal{L}^{\text{align}}$	3.204	0.107	1.793	**1.468**	0.0073	0.181
w/o $\mathcal{L}^{\text{align}}_{\text{con}}$	3.227	0.106	1.788	1.449	0.0055	0.177
w/o $\mathcal{L}^{\text{align}}_{\text{intra}}$	3.200	0.104	1.731	1.455	0.0031	0.148
w/o $\mathcal{L}^{\text{align}}_{\text{temp}}$	3.205	0.104	1.729	1.455	0.0024	0.148
w/o $\mathcal{L}_{\text{shot}}$	3.855	0.119	1.553	1.423	0.0022	0.146
w/o $\mathcal{L}_{\text{dyn}}$	3.270	0.104	1.769	1.457	0.0025	0.148
w/o $\mathcal{L}_{\text{vis}}$	3.203	0.104	1.729	1.456	0.0024	0.148

Results. The results show that full IITL achieves the lowest FID_k, FID_s, DMR, and LCD and the highest Dist_k, while the variant without $\mathcal{L}^{\text{align}}$ has the highest Dist_s (1.468 vs. 1.460). Removing $\mathcal{L}_{\text{shot}}$ produces the largest increase in both FID scores (3.181 to 3.855 and 0.103 to 0.119) and the largest decrease in both diversity metrics. This indicates poorer agreement with ground truth camera motion and framing distributions and reduced diversity, supporting the use of shot-structure supervision.

Removing any individual alignment term worsens all six metrics relative to full IITL. Adding the complete intent alignment loss improves Dist_k from 1.793 to 1.872 and reduces LCD from 0.181 to 0.145, while Dist_s decreases slightly from 1.468 to 1.460. These results support combining intra-shot intent agreement, shot-level contrastive alignment, and within-shot temporal consistency. Removing $\mathcal{L}_{\text{dyn}}$ or $\mathcal{L}_{\text{vis}}$ also worsens all six metrics, supporting their inclusion in the generator objective.

A.8 Intent aligner ablation

Experimental settings. Complementing the backbone comparison in Table 2(b), Table 7 evaluates the feature inputs and fusion strategy of the intent aligner under the same DCM evaluation protocol. Camera-kinematic only and Camera-geometry only retain their respective feature inputs. Early fusion concatenates both feature types before a single encoder, whereas the full intent aligner fuses separately encoded features, as described in Section A.3.

Table 7: Intent aligner ablation on DCM. Best and second-best results are **bold** and underlined. – denotes an unavailable result.

Method	Scale F1 $\uparrow$	Focus F1 $\uparrow$	Prim. mAP $\uparrow$	Energy MAE $\downarrow$
Camera-kinematic only	0.440	0.575	0.603	0.161
Camera-geometry only	0.397	0.616	0.559	–
Early fusion	0.457	0.629	0.544	0.162
Intent aligner (ours)	**0.492**	**0.664**	**0.631**	**0.155**

Results. Fusing separately encoded camera-kinematic and camera-geometry features improves scale, focus, and primitive prediction over variants using either feature type alone. Camera-geometry features alone yield higher focus F1 (0.616 vs. 0.575), whereas camera-kinematic features alone yield higher scale F1 and primitive mAP. The full intent aligner also outperforms early fusion on all four metrics, supporting separate encoding before feature fusion.