JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA
B.TECH V SEMESTER (2025-26)

MINOR PROJECT-1
(15B19CI591) PROJECT SYNOPSIS

TITLE OF PROJECT:

"Rhythm-Aware Human Motion Diffusion Model for Dance Generation"

Under the Supervision of: Dr. Alka Singhal

Submitted by:

23103391 Aditya Chaturvedi (B16)
23103397 Aditya Bajaj (B16)
23803014 Lovish kumar(B13)

ABSTRACT

Creating realistic dance moves that is a difficult problem in computer graphics and AI.
Most old models fail because their dance motions look unnatural, repetitive, or not in sync with the music beats.

In this project, we will build a pipeline using the **Motion Diffusion Model (MDM)** for **dance generation**.

MDM works by using a diffusion process: it starts with random noise and gradually removes it to create smooth, natural, and consistent 3D dance motions.

Our model will generate dance clips that are:

- **Visually natural** (look realistic)
- **Style-specific** (match dance style)

These dance clips can be useful for many applications such as:

- Animation and movies
- Gaming and VR characters
- Entertainment videos
- Social media creative tools

INTRODUCTION

Traditional deep learning models like RNNs, VAEs, and GANs are not very good for motion generation. They usually have two main problems:

1. They cannot keep movements smooth for a long time (poor long-term consistency).
2. They often create dance steps that look repetitive or unnatural.

The Motion Diffusion Model (MDM) solves these problems. It works step by step like a cleaning process:

- First, it starts with random noisy motion.
- Then, it slowly removes the noise.
- At the end, it creates a smooth and realistic motion sequence.

MDM can also use conditions (extra information) to control the output. For example:

- Text prompts → "make the character dance hip-hop style"
- Past motion → continue the same dance style smoothly

Because of this, MDM can generate different styles of dance that look realistic and fit the context.
In this project, we will use both:

- Benchmark motion datasets (real data already available)
- Synthetic data (artificially generated for experiments)

We will train and test an MDM model that can create short dance clips. These dance clips can be used for:

- Virtual characters in games or VR
- Animation and film-making tools
- Interactive systems like dance learning apps

MDM

- Smooth & Diverse Outputs: Diffusion-based models avoid mode collapse and repetitive outputs common in GANs/VAEs.

- Conditional Flexibility: Supports conditioning on non-audio features (e.g., text prompts, past frames) for style-aware synthesis.

- Temporal Consistency: Maintains realistic joint transitions across frames.

- Geometry & Shape Awareness: Can incorporate geometry-aware losses to enforce realistic poses and foot contacts.

- State-of-the-Art: Diffusion approaches match or surpass prior methods on realism and diversity metrics on motion benchmark.

- MDM frames human motion generation as a denoising diffusion probabilistic model (DDPM):

- Forward Process: Gradually add Gaussian noise to real motion data over T timesteps.

- Reverse Process: Train a neural network to iteratively remove noise, reconstructing motion from noise.

- Conditional Generation: Text or semantic features (encoded tokens) are provided to the denoiser (via cross-attention) so generated motion follows the requested style/description.

FORMULAS

1. Forward Process (Noise Addition)

The **forward diffusion process** gradually adds Gaussian noise to a clean data sample x_0 over T timesteps. The noisy sample at step t, x_t, is:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$$

where:

- x_0 = original clean motion data
- x_t = noisy motion at time t
- $\epsilon \sim \mathcal{N}(0, I)$ = Gaussian noise
- $\alpha_t = 1 - \beta_t$ = noise retention factor at step t
- $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ = cumulative noise retention up to step t
- $\beta_t \in (0, 1)$ = variance schedule (small positive values)

This equation lets you directly sample x_t from x_0 **in one step** without simulating all $t - 1$ steps.

---

2. Reverse Process (Denoising)

The **reverse diffusion process** removes noise step by step, predicting x_{t-1} from x_t:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\, \epsilon_\theta(x_t, t) \right) + \sigma_t z$$

where:

- $\epsilon_\theta(x_t, t)$ = model's prediction of the noise at step t
- $z \sim \mathcal{N}(0, I)$ = fresh Gaussian noise (used only for stochastic sampling)
- $\sigma_t^2 = \tilde{\beta}_t$ = variance of the reverse step,
  where $\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$

ALGORITHMS AVAILABLE FOR GENERATIVE MOTION

RNN/LSTM-based Models: Sequential prediction; often accumulate errors over long sequences.

Variational Autoencoders (VAEs): Compact latent representations but can produce averaged/blurred motions.

Generative Adversarial Networks (GANs): Can create realistic frames but suffer from temporal instability and mode collapse.

Transformers: Capture long-range temporal dependencies; computationally intensive but powerful for sequence modeling.

Diffusion Models (e.g., MDM): Progressive denoising from noise to motion — strong diversity and temporal coherence.

METHODOLOGY

1. Problem Statement

Generate realistic human dance motion sequences conditioned on semantic inputs (text prompts) and past motion context.

Preserve temporal consistency and natural joint movement throughout the clip.
Support style-specific generation (hip-hop, ballet, contemporary, funky, etc.).
Reduce mode collapse or overfitting that occurs in GAN-based methods.

2. Dataset Preparation

Datasets Used

AIST++ (3D dance motion sequences, used here as motion-only data).

Mixamo (animation / retargeting support).

Human3.6M (general motion capture for augmenting pose diversity).

Synthetic procedural dataset (for controlled experiments and ablation studies).

Data Preprocessing Steps

Motion Normalization: Standardize skeleton positions, root translation, and orientation; represent motions as root-relative positions or joint rotations.

Sequence Segmentation: Clip motions into short segments (e.g., 5–10 s) with consistent frame rate.

Text Tagging / Labels: Associate each motion segment with text prompts or style labels (for conditional training).

3. Model & Training Methodology

Model Used: Motion Diffusion Model (MDM) with a Transformer-based denoiser and a frozen text encoder (e.g., DistilBERT) for semantic conditioning.

Training Steps

Model Initialization: Start from scratch or load pretrained backbone weights if available.

Conditioning: Provide text-token embeddings (or null tokens for classifier-free guidance) alongside motion input.

Losses:

Denoising Loss: MSE between predicted and target noise.

Geometry Losses: Joint-angle regularization, foot contact consistency, and other kinematic constraints.

Optimization: AdamW optimizer, gradient clipping, checkpointing and early stopping based on validation metrics (e.g., FID or motion quality).

4. Inference Pipeline

Condition Encoding: Encode the text prompt / style tokens and prepare past motion context if used.

Diffusion Sampling: Start from Gaussian noise and iteratively denoise using the trained MDM conditioned on text tokens.

Post-processing: Optionally smooth joint rotations, enforce foot contacts, and retarget motion to specific skeleton rigs (Mixamo).

Output: Exported motion arrays (e.g., .npy) and rendered 3D animation sequence.

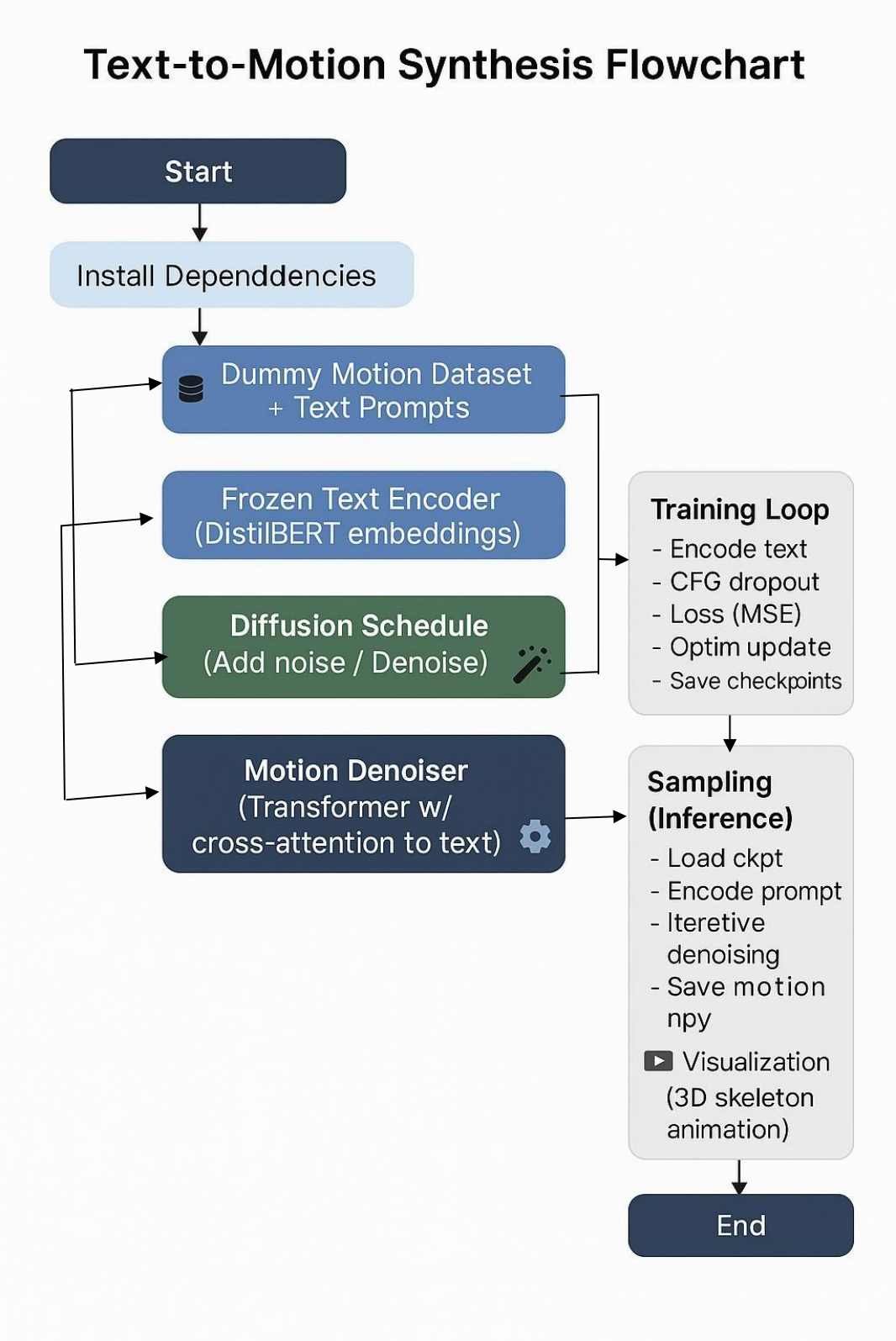
5. Expected Output

A trained MDM capable of producing natural, diverse, and style-aware dance motions conditioned on text or past motion.

Generated clips suitable for avatar animation, game engines, or visualization tools.

Quantitative comparison against baselines (RNN, VAE, GAN, Transformer) showing improved temporal consistency and realism.

FLOWCHART


Text-to-Motion Synthesis Flowchart
Start
Install Dependddencies
Dummy Motion Dataset + Text Prompts
Frozen Text Encoder (DistilBERT embeddings)
Diffusion Schedule (Add noise / Denoise)
Motion Denoiser (Transformer w/ cross-attention to text)
Training Loop
- Encode text
- CFG dropout
- Loss (MSE)
- Optim update
- Save checkpoints
Sampling (Inference)
- Load ckpt
- Encode prompt
- Iteretive denoising
- Save motion npy
Visualization (3D skeleton animation)
End

KEY FEATURES

- Diffusion-based Generation: Smooth, high-quality motion synthesis with strong diversity.

- Text/Semantic-Conditioning: Generate motions responsive to text prompts or style descriptors.

- Geometry & Kinematic Awareness: Integrate losses that maintain believable joint angles and foot contact.

- Generalization: Adaptable to unseen styles and motions via conditioning and data augmentation.

- Application-Ready: Retargetable outputs for virtual avatars and animation pipelines.

REFERENCES

1. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., & Bermano, A. H. (2022). Human motion diffusion model. *arXiv preprint arXiv:2209.14916*.
2. Zhou, W., Dou, Z., Cao, Z., Liao, Z., Wang, J., Wang, W., ... & Liu, L. (2024, September). Emdm: Efficient motion diffusion model for fast and high-quality motion generation. In *European Conference on Computer Vision* (pp. 18-38). Cham: Springer Nature Switzerland.
3. Han, X., Xu, H., & Sun, H. (2024, October). Motion Diffusion Model for Long Motion Generation. In *2024 5th International Conference on Machine Learning and Computer Application (ICMLCA)* (pp. 603-608). IEEE.
4. Chen, X. (2024). Text-driven human motion generation with motion masked diffusion model. *arXiv preprint arXiv:2409.19686*.