HAW HAMBURG

**BACHELOR THESIS**

Tim Spulak

# A Diffusion-Based Approach to 3D Human Motion Synthesis

Faculty of Engineering and Computer Science
Department Computer Science

**HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG**
**Hamburg University of Applied Sciences**

Tim Spulak

# A Diffusion-Based Approach to 3D Human Motion Synthesis

Bachelor thesis submitted for examination in Bachelor´s degree
in the study course *Bachelor of Science Angewandte Informatik*
at the Department Computer Science
at the Faculty of Engineering and Computer Science
at University of Applied Science Hamburg

Supervisor: Prof. Dr. Kai von Luck
Supervisor: Dr. Jan Schwarzer

Submitted on: November 27, 2025

**Tim Spulak**

**Title of Thesis**

A Diffusion-Based Approach to 3D Human Motion Synthesis

**Keywords**

3D Human Motion, Action-Conditioned Synthesis, Denoising Diffusion Probabilistic Model, Exponential Moving Average, Importance Sampling

**Abstract**

This thesis investigates the impact of two generic optimization strategies—importance sampling and Exponential Moving Average—on a diffusion model for action-conditioned 3D human motion synthesis. Therefore, the Human Motion Diffusion Model is adapted and applied to the UESTC dataset—comprising complex movements. The experiments are conducted in three phases: (i) analyzing the effect of reducing the diffusion process from 1,000 to 100 steps and comparing the results to state-of-the-art methods, (ii) assessing the isolated and combined influence of importance sampling and Exponential Moving Average under a reduced number of training steps, and (iii) running a full-scale training using the best configuration from the second phase. Reducing diffusion steps significantly decreases computational cost while results show competitive motion quality. In contrast, both importance sampling and Exponential Moving Average produce marginal improvements in quantitative metrics and do not meaningfully alter convergence in the long term. Moreover, visual inspection of generated samples reveals significant artifacts such as foot sliding, and difficulties in reflecting fine-grained movements. These results indicate that, for this task and dataset, domain-specific optimizations dominate performance, while generic optimization strategies yield negligible improvements.

**Tim Spulak**

**Thema der Arbeit**

Ein diffusionsbasierter Ansatz zur Synthese menschlicher 3D-Bewegungsmuster

**Stichworte**

Menschliche 3D-Bewegungsmuster, Aktionskonditionierte Synthese, Denoising Diffusion Probabilistic Model, Exponential Moving Average, Importance Sampling

**Kurzzusammenfassung**

In der vorliegenden Arbeit wird der Einfluss zweier generischer Optimierungsstrategien–importance sampling und Exponential Moving Average–auf ein Diffusionsmodell zur aktionsbasierten Generierung dreidimensionaler menschlicher Bewegungsmuster untersucht. Hierfür wird das Human Motion Diffusion Model angepasst und auf den UESTC-Datensatz angewendet, der komplexe Bewegungsabläufe umfasst. Die Experimente gliedern sich in drei Phasen: (i) Analyse der Auswirkungen der Reduktion des Diffusionsprozesses von 1.000 auf 100 Schritte, (ii) Bewertung des isolierten und kombinierten Einflusses von importance sampling und Exponential Moving Average unter reduzierter Trainingsdauer, und (iii) Durchführung eines vollständigen Trainings mit der vielversprechendsten Konfiguration aus Phase zwei. Die Reduzierung der Diffusionsschritte führt zu einer deutlichen Verringerung des Rechenaufwands bei vergleichbarer Bewegungsqualität. Im Gegensatz dazu bewirken sowohl importance sampling als auch Exponential Moving Average nur marginale Verbesserungen der quantitativen Metriken und zeigen über den beobachteten Zeitraum hinweg keinen nennenswerten Einfluss auf die Konvergenz. Die visuelle Analyse der generierten Sequenzen zeigt zudem signifikante Artefakte wie Fußrutschen sowie Schwierigkeiten bei der Darstellung feingranularer Bewegungen. Insgesamt deuten die Ergebnisse darauf hin, dass für diese Aufgabe und diesen Datensatz primär domänenspezifische Optimierungen die Modellleistung bestimmen, während generische Optimierungsstrategien lediglich vernachlässigbare Vorteile bieten.

# Acknowledgments

I would like to thank Michael J. Black and the SMPL licensing team for granting permission to use the SMPL model to generate meshes for the purposes of this thesis.

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **DDIM** | Denoising Diffusion Implicit Model |
| **DDPM** | Denoising Diffusion Probabilistic Model |
| **EMA** | Exponential Moving Average |
| **FID** | Fréchet Inception Distance |
| **FK** | Forward Kinematics |
| **GAN** | Generative Adversarial Network |
| **IK** | Inverse Kinematics |
| **KL** | Kullback-Leibler |
| **MDM** | Human Motion Diffusion Model |
| **SMPL** | Skinned Multi-Person Linear |
| **VAE** | Variational Auto-Encoder |
| **VLB** | Variational Lower Bound |

# 1 Introduction

*3D Human motion* synthesis seeks to generate natural and diverse human movements, playing a crucial role in a variety of fields such as the film and video game industry, robotics, and human behavior analysis [17, 29, 35, 37]. Other research also explores the use of synthetic motion data to augment datasets to improve model training [17, 30].

Typically, motion generation is contextualized on different types of signals, including scene context, audio, or textual descriptions. Among these, textual signals are the preferred modality for many researchers [3, 37]. While the non-linearity and physical plausibility of human motion are challenging, conditioning on signals further increases the complexity of the task. Achieving realism in generated motion now depends not only on the movement itself but also on its semantic alignment with the conditioning signal. Moreover, the realism of human motion heavily relies on human perception, which is highly sensitive to subtle inconsistencies in human movements [37].

The generation of motion without an initial pose or sequence has historically been under-explored [17]. Early work relied on statistical models to generate basic movements such as walking. In contrast, subsequent research shifted toward more constrained settings, most notably motion prediction, where future frames are generated from a given pose or sequence of poses [17]. In recent years, the focus of research has broadened again. Advances in deep generative modeling—spanning Variational Auto-Encoders (VAEs) [10], Generative Adversarial Networks (GANs) [5], and Normalizing Flows [22]—combined with improved human body models such as the *SMPL* model [14], have enabled the construction of larger datasets and renewed interest in human motion synthesis. These developments made the task of unconstrained human motion generation increasingly feasible [37].

More recently, *Denoising Diffusion Probabilistic Models (DDPMs)* [7] have gained widespread attention due to advances in the image synthesis domain [3], where they have

ultimately outperformed GANs [4]. These developments have motivated their application to the domain of human motion synthesis as well [35, 3], leveraging their ability to efficiently capture the underlying data distribution [29]. Building on these developments, Tevet et al. [29] have developed the *Human Motion Diffusion Model (MDM)*, a lightweight diffusion-based framework capable of generating motion from diverse modalities and achieving state-of-the-art results.

## 1.1 Research Aim

The aim of this work is to investigate how well a diffusion-based model can generate realistic 3D human motion—particularly in settings where synthetic data may later be used to extend existing datasets. Therefore, the MDM framework is adapted and applied to the *UESTC* dataset [8], which contains complex actions and fine-grained movements that go beyond simple movements such as walking.

While MDM supports multiple conditioning modalities, this study restricts the setup to *action-conditioned* motion generation. Action labels provide a simple conditioning signal, avoiding the complex mapping inherent to natural language processing. This setup allows the evaluation to focus on the effectiveness and efficiency of generic training dynamics and optimization strategies, in particular *importance sampling* and *EMA*. Additionally, the results are compared to the original MDM model trained on the UESTC dataset.

## 1.2 Thesis Organization

The thesis comprises five chapters. The subsequent Chapter 2 introduces the theoretical background of diffusion models and action-conditioned motion generation, and formulates the research question. Chapter 3 outlines the methodological setup, including the experimental setup and measures taken to ensure reproducibility. Chapter 4 presents and analyzes the results, accompanied by a discussion of their implications and limitations. Finally, Chapter 5 concludes the key findings and outlines potential directions for future research.

# 2 Related Work

This chapter provides an overview of the relevant research on diffusion models and their application to action-conditioned motion synthesis. It highlights existing gaps in the literature and outlines the research question addressed in this thesis. Section 2.1 discusses the mathematical foundations and advancements in diffusion models, including optimization strategies (EMA and importance sampling), which are central to this thesis' experimental contributions. Section 2.2 reviews state-of-the-art methods for action-conditioned motion synthesis, emphasizing current challenges and limitations, and introduces the MDM, the foundational model of this thesis. Section 2.3 formulates the research question based on identified gaps in the literature.

## 2.1 Diffusion-Based Generative Models

This chapter presents the fundamentals and mathematical background of diffusion models. It begins with an introduction to the forward and reverse diffusion processes in Section 2.1.1, followed by a discussion of advances in training objectives over time in Section 2.1.2. Section 2.1.3 then covers the basics of sampling, including methods that enhance efficiency and quality. Finally, Section 2.1.4 presents optimization techniques aimed at improving training dynamics and stability.

### 2.1.1 Diffusion Process

Diffusion models are based on a two-stage process: the forward process, wherein noise is gradually added to the data, and the reverse process, wherein a neural network is trained to eliminate the added noise. Together, these define a Markov chain that transforms clean data into noise and vice versa [7, 25, 32].

Figure 2.1: Illustration of variance schedules. Visual comparison of latent samples produced under a linear schedule (top) and a cosine schedule (bottom) at linearly spaced timesteps $t \in [0, T]$. The figure illustrates how different noise schedules affect the progressive corruption as $t$ increases (adapted from Figure 3 in [16]).

In the forward process, fractions of Gaussian noise are incrementally added to the input data $x_0$ over a fixed number of timesteps $T$. The result of this progressive noise addition is a series of increasingly distorted samples, denoted as $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. It is noteworthy that $\mathbf{x}_T$ approaches an isotropic Gaussian distribution. By the final step, denoted by $T$, the original data is effectively destroyed [32].
The application of noise per timestep is defined by a variance schedule (cf. Equation (2) in [7]) $\{\beta_t \in (0,1)\}_{t=1}^{T}$

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) \coloneqq \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \quad q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \coloneqq \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t \mathbf{I}). \tag{2.1}$$

As illustrated in Figure 2.1, two prevalent variance schedules are depicted: a linear schedule (upper) and a cosine schedule (lower). These schedules have been proposed in the literature and are discussed in detail in the works of Ho et al. [7] and Nichol and Dhariwal [16], respectively. The iterative calculation of noise can result in suboptimal efficiency. However, a closed-form expression for $q(\mathbf{x}_t \mid \mathbf{x}_0)$ can be derived through the application of the reparameterization trick [7, 32]. Defining $\alpha_t \coloneqq 1 - \beta_t$ and the cumulative product $\bar{\alpha}_t \coloneqq \prod_{i=1}^{t} \alpha_i$, the forward process can be expressed as described in [32]:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\,\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\,\boldsymbol{\epsilon}_{t-1}, && \text{;where } \boldsymbol{\epsilon}_{t-1}, \boldsymbol{\epsilon}_{t-2} \ldots \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
&= \sqrt{\alpha_t \alpha_{t-1}}\,\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\,\bar{\boldsymbol{\epsilon}}_{t-2}, && \text{;where } \bar{\boldsymbol{\epsilon}}_{t-2} \text{ merges two Gaussians} \\
&\vdots \\
&= \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}, \\
q(\mathbf{x}_t \mid \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\,\mathbf{I}).
\end{aligned} \tag{2.2}$$

The closed form enables the direct sampling of a noisy version of $x_0$ at any timestep $t$ without the necessity of computing all intermediate steps [7, 32]. The reverse process aims to recover data by gradually removing the noise added during the forward diffusion. Although it is possible in theory, the reconstruction of an original sample by sampling from $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ at any step of the reverse process is a process that requires knowledge of the entire dataset. Therefore, the posterior distribution is intractable in practice [25, 32].

Thus, a neural network with parameters $\theta$ is trained to approximate these probabilities. In particular, the model forecasts the mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ and variance $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ of a Gaussian distribution, modeling the reverse Markov process as (cf. Equation (1) in [7])

$$p_\theta(\mathbf{x}_{0:T}) \coloneqq p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t), \quad p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \coloneqq \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)). \quad (2.3)$$

While $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ is intractable, it becomes analytically tractable when conditioned on the original clean input $x_0$ (cf. Equation (6) in [7]):

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}). \quad (2.4)$$

The mean $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)$ and variance $\tilde{\beta}_t$ are expressed as a function of the standard Gaussian density function.
In order to estimate the original data from noisy inputs at any given timestep, it is necessary to use a reparameterization of $x_0$ in terms of $x_t$ and known noise. [7, 16, 32].

### 2.1.2 Training

Training diffusion models involves maximizing the model log-likelihood. However, direct calculation of the likelihood is intractable. To address this issue, Jensen's inequality is employed to derive a Variational Lower Bound (VLB) on the log-likelihood [25, 32]. As derived in in [25]), the VLB can be expressed as a sum of Kullback-Leibler (KL) divergence terms and a negative log-likelihood term. These terms can be efficiently

evaluated and optimized (cf. Equation (4-7) [16]):

$$L_{\text{VLB}} := L_0 + L_1 + \cdots + L_{T-1} + L_T \tag{2.5}$$

$$L_0 := -\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1) \tag{2.6}$$

$$L_{t-1} := D_{\text{KL}}\big(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\big) \tag{2.7}$$

$$L_T := D_{\text{KL}}\big(q(\mathbf{x}_T \mid \mathbf{x}_0) \,\|\, p(\mathbf{x}_T)\big) \tag{2.8}$$

Each term $L_t$, with the exception of $L_0$, involves a comparison between two Gaussian distributions and can be computed in closed form. The term $L_0$ corresponds to the reconstruction error between the predicted and true data distributions. In order to optimize this bound in an efficient manner, the training process samples a timestep $t$ uniformly [16] and uses the closed-form expression for $q(x_t|x_0)$ to estimate the corresponding $L_{t-1}$ term of the VLB [16].

The process of approximating the reverse Markov transitions $p_\theta(x_{t-1}|x_t)$ requires the reparameterization of the mean $\mu_\theta(x_t, t)$ of the Gaussian at each step. Common parameterizations include the direct prediction of $\mu_\theta$, the prediction of the original data $x_0$, or the prediction of the noise $\epsilon$ added at each step $t$ [7]. According to the findings of Ho et al. [7], the most optimal result is achieved when predicting epsilon, thereby ensuring the highest attainable quality of the sample and the greatest possible training stability. Under the parameterization, the mean can be expressed as (cf. Equation (11) in [7]):

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big). \tag{2.9}$$

Consequently, the loss $L_t$ per timestep reduces to a weighted mean squared error between the true noise $\boldsymbol{\epsilon}$ and the predicted noise $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ (cf. Equation (12) in [7]):

$$\mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right]. \tag{2.10}$$

where the weighting factor $\dfrac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}$ depends on the variance schedule [7, 16].

Ho et al. further proposed a simplified training objective $L_{simple}$ (cf. Equation (14) in [7]) that discards this weighting, making the implementation easier and leading to improved sample quality:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2\Big]. \tag{2.11}$$

Removing the weight factor reduces the relative influence of inputs with low amounts of noise. Consequently, this results in an empirical shift in learning towards harder denoising samples with high-noise cases [7].

### 2.1.3 Sampling

In the original DDPM proposed by Ho et al. [7], the process of sampling begins from pure Gaussian noise, $x_T \sim p(x_T)$. Subsequently, a neural network iteratively reverses the aforementioned process to recover the original input $x_0$. The quality of the generated samples depends on the number of diffusion steps, denoted by $T$. While larger values of $T$ bring the reverse process distribution closer to Gaussian and thus provide a better approximation of the forward process, they also lead to slower sampling, as the denoising process is sequential. Thus, while the simple objective (Equation (2.11)) significantly improves visual quality, the original DDPM still especially lags behind state-of-the-art GANs in sampling speed [7, 26].

Nichol and Dhariwal's [16] Improved DDPM introduced several changes to address these issues: (i) a cosine-based noise schedule (see Figure 2.1), (ii) a learned variance parameterization in combination with a new objective $L_{hybrid} = L_{simple} + \lambda L_{VLB}$ to improve log-likelihoods, and (iii) a strided sampling strategy that reduces the number of steps from $T$ to $S \ll T$ by evaluating the reverse process only at a subset $\{\tau_1, \ldots, \tau_S\}$ of timesteps. These modifications enable faster sampling and close the gap to the sample quality of GAN while outperforming them in mode coverage.

Song et al.'s [26] Denoising Diffusion Implicit Model (DDIM) approaches the problem of slow inference from a different perspective. Since the DDIM training objective depends only on the marginal distributions $q(x_t|x_0)$ rather than the joint distribution over all timesteps, the reverse process no longer needs to be strictly stochastic. This allows the model to utilize a non-Markovian formulation and sample over a subset of timesteps, denoted $\{\tau_1, \ldots, \tau_S\}$, where $S$ can be much smaller than the full chain length $T$ [32]. Therefore, DDIM generalizes the variance used in the reverse step (as described in [32])

$$\tilde{\beta}_t = \sigma_t^2 = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t. \tag{2.12}$$

DDIM introduces a stochasticity parameter $\eta \in [0, 1]$ such that the sampling variance $\sigma_t^2$ is defined as $\eta \cdot \tilde{\beta}_t$. By setting $\eta = 0$, DDIM allows for substantially fewer timesteps without sacrificing sample quality [32].

In summary, Improved DDPM accelerates sampling by reducing the number of reverse process evaluations via timestep sub-sampling, while DDIM achieves speedups by reformulating the reverse process to allow partially or fully deterministic sampling. Both approaches reduce the number of required denoising steps, with the DDIM formulation also enabling deterministic generation of identical samples given a fixed random seed. These advances are particularly relevant to the motion synthesis domain, where long sequences and high-dimensional representations make efficient sampling crucial.

### 2.1.4 Optimization Strategies

Beyond the fundamental optimizations in diffusion models discussed above, several optimization techniques have been proposed to further improve training efficiency and stability. Among these methods are importance sampling of timesteps [16], and EMA [9, 18].

#### Importance Sampling

In standard diffusion training, timesteps $t \in [0, T-1]$ are sampled uniformly, when computing the training loss. However, different timesteps contribute unequally to the overall objective, and some may produce noisier gradients. Importance sampling addresses this by weighting timesteps according to their estimated loss magniture (cf. Equation (18) [16]):

$$L_{vlb} = \mathbb{E}_{t\sim p_t}\left[\frac{L_t}{p_t}\right], \text{where } p_t \propto \sqrt{\mathbb{E}\left[L_t^2\right]}, \text{and } \sum p_t = 1. \tag{2.13}$$

Since the expected squared loss $\mathbb{E}[L_t^2]$ is unknown and may change during training, it can be approximated using a moving history of previous loss values. This approach focuses training on the timesteps that contribute most to instability, reducing noise and improving convergence when optimizing the variational lower bound $L_{vlb}$. While importance sampling is most effective for objectives with high variance across timesteps, it may be less impactful for more stable or hybrid objectives [16].

#### Exponential Moving Average

Loss functions in diffusion models are stochastic, as the training signal is derived from multiple aspects, such as random noise, and conditioning variables. This stochasticity

can destabilize the training dynamics, making optimization sensitive to hyperparameters [9]. To mitigate such instability, EMA has become a common technique in deep learning, especially in the context of generative models. Initially utilized in GANs, EMA has since become standard in diffusion models [13].
While there are different implementations, traditional EMA is a copied model, maintaining a weighted average $\hat{\theta}_\beta$ of the original model parameters $\theta$ over the course of training. At each training step $t$, the average is updated as described in [9]:

$$\hat{\theta}_\beta(t) = \beta\hat{\theta}_\beta(t-1) + (1-\beta)\theta(t), \tag{2.14}$$

where the decay constant $\beta \in (0, 1)$ is typically set close to one. This smooths out parameter updates by providing an exponential decay of past values in which recent updates have higher weight while earlier updates contribute less [9].
The benefits of EMA are mainly empirical across supervised, unsupervised, and generative training [1]. It significantly improves training stability by suppressing gradient noise [13], and its slowly evolving parameter averages. Furthermore, EMA reduces overfitting due to wider minima, and is inexpensive because of its simple update rule [1]. Even though its effects can be compared to other regularizations, such as learning rate scheduling [13], which directly impact optimization dynamics, EMA instead works by smoothing the parameter trajectory throughout training.

## 2.2 Action-Conditioned Motion Synthesis

This chapter provides an overview of action-conditioned human motion synthesis, highlighting the key components involved in generating realistic human motion sequences corresponding to specific actions. It begins by discussing the challenges and considerations specific to datasets, introducing the UESTC dataset in Section 2.2.1. Section 2.2.2 presents the evaluation metrics commonly used to assess performance. The advantages and limitations of DDPMs in the motion synthesis domain are then discussed, along with an introduction to the MDM framework in Section 2.2.3. Finally, Section 2.2.4 examines domain-specific optimizations, such as pose representations and geometric loss functions, and their role in supporting architectural approaches.

### 2.2.1 UESTC Dataset

The action-conditioned motion synthesis task relies on datasets that provide paired action labels and corresponding motion sequences [34]. However, there remains a scarcity of motion capture data specifically designed for this purpose, which has led researchers such as Petrovich et al. [17] to employ monocular motion estimation techniques for generating motion sequences. While these methods expand data availability, they often introduce noisy artifacts into the resulting sequences [17]. Moreover, many existing datasets lack standardized train/test splits [2], which limits the ability to reliably evaluate model generalization and compare performance across approaches. Among the most commonly used datasets for this task are HumanAct12 [6], NTU RGB+D [23], and UESTC [8].

The UESTC dataset [8] is the only one among them that provides predefined train/test splits. Originally created for human-robot interaction tasks, it contains 40 action categories—primarily exercises and cyclic movements—across 25,600 motion sequences performed by 118 subjects. Fifteen of these actions are executed in both standing and sitting positions. The recordings were captured using a Microsoft Kinect V2 camera and include RGB, depth, and skeleton sequences. In total, the dataset comprises 83 hours of video footage recorded from eight viewpoints. Sequences with fixed viewpoints last between 6 and 7 seconds (approximately 200 frames), whereas those with varying viewpoints range from 55 to 65 seconds in duration, corresponding to roughly 1,730-2,000 frames.

The dataset provides multiple evaluation protocols, including Cross-subject recognition, two variants of Cross-view recognition, and Arbitrary-view recognition. Most studies on action-conditioned motion synthesis adopt the Cross-subject split [2, 17, 29, 35]. In this protocol, the training split consists of all action categories performed by a fixed subset of 51 subjects, comprising 10,650 sequences, while the remaining subjects form the test split with 13,350 sequences.

While Ji et al. [8] emphasize the dataset's diversity—covering both large and fine-grained movements that result in complex motion patterns—it is important to note that the use of Kinect sensors inherently limits motion quality. As these sensors capture only monocular depth information, motion sequences must be reconstructed using estimation techniques as described above, leading to noisy artifacts [17].

### 2.2.2 Evaluation Metrics

Evaluating generated motions is inherently challenging due to the one-to-many nature of the task, the subjective perception of motion quality, and the need to assess both motion fidelity and semantic consistency [37]. While automated metrics allow large-scale, reproducible evaluation, human judgment ultimately determines how natural a motion appears. Human studies can complement quantitative results by capturing perceptual and cultural nuances, but they are time-consuming and less scalable [27]. This thesis therefore focuses on automated metrics. For the action-conditioned motion synthesis task, four key evaluation metrics are commonly employed: Fréchet Inception Distance (FID), Diversity, Multimodality, and Recognition Accuracy. To compute these metrics, a separate recognition model must first be trained [17].

Fidelity metrics evaluate generated motions based on naturalness, smoothness, and plausibility [37]. FID measures the difference between the feature distributions of generated and real motion data [6, 35]. It is widely used to assess the overall quality of generated motions, where lower FID values indicate a higher similarity to real data [6, 17, 34].

Diversity measures a model's ability to generate a wide range of motions while avoiding repetitions and frozen outputs [37]. It is computed by sampling two subsets $\{v_1, ..., v_{S_d}\}$ and $\{v'1, ..., v'S_d\}$ of size $S_d$ from all generated motions and is defined as (cf. Equation (6) in [29]):

$$\text{Diversity} = \frac{1}{S_d} \sum_{i=1}^{S_d} \left\| v_i - v'_i \right\|_2 . \tag{2.15}$$

Multimodality evaluates the diversity of generated motions conditioned on the same action [29, 35]. Similar to Diversity, it is computed by sampling two subsets of size $S_d$, belonging to the same action class $c$, $\{v_{c,1}, ..., v_{c,S_d}\}$ and $\{v'c, 1, ..., v'c, S_d\}$, and is defined across all action classes $C$ as (cf. Equation (7) in [29]):

$$\text{Multimodality} = \frac{1}{C \times S_d} \sum_{c=1}^{C} \sum_{i=1}^{S_d} \left\| v_{c,i} - v'_{c,i} \right\|_2 . \tag{2.16}$$

Optimal Diversity and Multimodality values are achieved when they approximate those computed from real motion data [29].

The Recognition Accuracy metric assesses the overall correlation between generated motions and their conditioning signals [3, 29]. Combining these metrics provides a compre-

hensive measure of both realism and diversity [29]. While Diversity and Multimodality complement FID and Recognition Accuracy, FID is most commonly reported as the principal metric for motion quality and therefore serves as the primary optimization target [6, 17, 34]. Notably, FID remains underexplored (cf. Table 4 in [29]) in the context of the UESTC dataset for current state-of-the-art architectures such as [2, 3, 29, 35], leaving room for further improvement.

### 2.2.3 Diffusion in Motion Synthesis

Human motion synthesis is a complex task that requires capturing the inherent diversity of possible movements [29], while simultaneously maintaining perceptual realism and physical plausibility [37]. With the advent of deep learning, a wide range of model architectures has been explored to address these challenges, including VAEs, GANs, Normalizing Flow Networks, and Implicit Neural Representations [24] [34, 37]. Despite the progress achieved by these approaches, they often remain limited in quality, and expressiveness [29].

More recently, DDPMs have emerged as a promising alternative due to several advantages over earlier architectures. Tevet et al. [29] argue that DDPMs are particularly well suited for the motion synthesis task because they inherently model the many-to-many nature of the underlying problem. Another advantage of the DDPM framework lies in its ability to retain the original motion sequence throughout the diffusion process, allowing additional constraints to be applied during denoising to improve motion consistency [34]. Furthermore, DDPMs tend to produce more diverse samples compared to previous generative approaches [34].
Despite these advantages, DDPMs also come with certain drawbacks. Yuan et al. [33] argue that DDPMs tend to generate physically unrealistic motion sequences, due to the absence of physical constraints during training. Moreover, they tend to be computationally intensive, and challenging to control [29, 35].

Tevet et al. (2023) [29] tackle these drawbacks and utilize the advantages by proposing MDM, a classifier-free diffusion model capable of solving multiple generation tasks, such as text-conditioned synthesis, and motion editing. MDM currently represents the state-of-the-art in action-conditioned motion synthesis and the UESTC dataset (cf. Table 4 in [29]).

It uses a transformer encoder-only [31] backbone, which facilitates generating arbitrary-length motion sequences, and offers a temporally-aware structure beneficial for the motion generation task. During training, Gaussian noise is applied to each motion sequence. Each frame in these sequences serves as an input token, concatenated with a positional embedding. The final input is enriched by an additional token representing the condition and the current diffusion timestep. While a Contrastive Language-Image Pre-training model [21] is utilized for the text-conditioned motion synthesis task, a simple learned embedding suffices for each action in the action-conditioned task [29].
The generated motion sequence can be described as $x^{1:N} = \{x^i\}_{i=1}^{N}$. The underlying pose representation—rotations or joint positions—is denoted as $x^i \in \mathbb{R}^{JxD}$, where $J$ is the number of joints and $D$ the dimensionality of each joint. Moreover, MDM comprises geometric losses, which require the model to predict the original motion sequence $x_0$ at any timestep $t$ from the noisy input $x_t$ [29], which will be discussed in more detail in the next section.

### 2.2.4 Domain-Specific Optimizations

As discussed in Section 2.2.3, DDPMs provide substantial flexibility in architectural design. Consequently, several studies have focused on architectural innovations—for instance, enhancing sampling efficiency by integrating a GAN discriminator [35], performing diffusion in latent space [3], or incorporating a physics simulator into the diffusion process [33]. In contrast, this section focuses on domain-specific optimizations that address the unique characteristics of human motion, particularly pose representation and geometric loss functions, which are crucial for ensuring structural coherence and perceptual realism.

#### Pose Representation

Human motion can be described as a temporal sequence of body poses (see Figure 2.2). Pose representations are commonly divided into keypoint-based and rotation-based categories (see Figure 2.3). While Forward Kinematics (FK) enables the conversion of rotation-based representations into keypoint-based representations, Inverse Kinematics (IK) allows reconstructing rotations from keypoints [37]. These conversions are especially useful for animations. By utilizing rotations, it is possible to apply the SMPL model [14], which generates a triangular 3D mesh. SMPL models the body with $K = 24$

joints. Each joint has a set of parameters $\theta \in \mathbb{R}^{K\times 3}$, calculated with respect to the parent joint [37].

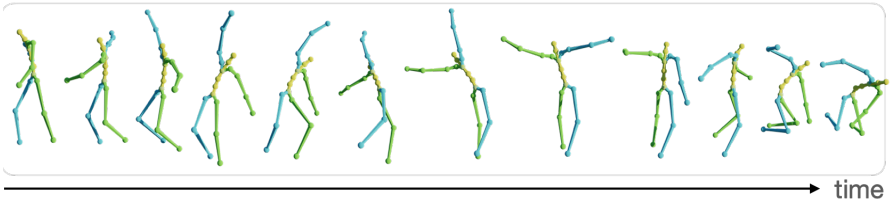



Figure 2.2: Illustration of a sequence of body poses across 12 frames with respect to the temporal axis (left to right) (adapted from Figure 11 in [20]).

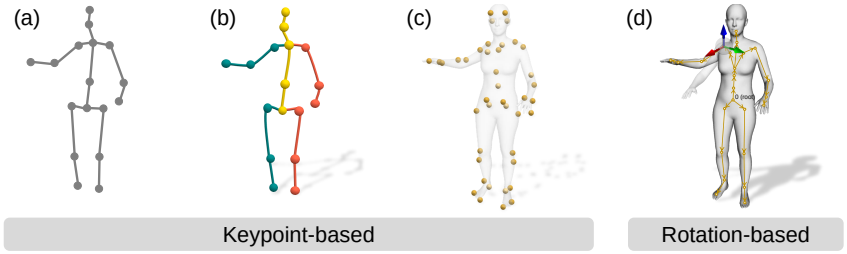



Figure 2.3: Typical human pose and shape representations with the same pose in (a) 2D keypoints, (b) 3D keypoints, (c) 3D marker keypoints, and (d) rotation-based model. (reproduced from Figure 3 in [37]).

Although 3D rotations can be parameterized using representation spaces [37] such as Euler angles, axis-angles, or quaternions, Zhou et al. [36] demonstrated that low-dimensional representations ($< 5$D) introduce discontinuities that harm neural network learning. To address this, they propose a continuous 6D rotation representation, derived from the first two columns of a rotation matrix and orthogonalized through a cross product. This representation has become a de-facto standard in recent motion diffusion works [17, 29, 34, 35], as it ensures smooth interpolation, stability, and compatibility with the application of FK and IK. A detailed explanation of the mathematical formulation and continuity advantages of the 6D rotation representation can be found in Appendix A.

### Geometric Loss Functions

One key component of the training process in MDM is the use of geometric loss functions, which are applied in addition to the simple objective (see Section 2.1.2). Since these losses measure geometric differences between the ground truth and the predicted samples, the

model must predict the original data $x_0$ at every timestep $t$ during the reverse diffusion process.

Geometric losses are motivated by prior work in motion synthesis, and address common issues such as jitter by regulating the velocity [17] and foot sliding [27] by implementing a dedicated loss term. Tevet et al. [29] propose three geometric losses:
position loss

$$\mathcal{L}_{\text{pos}} = \frac{1}{N} \sum_{i=1}^{N} \left\| FK(x_0^i) - FK(\hat{x}_0^i) \right\|_2^2, \tag{2.17}$$

velocity loss

$$\mathcal{L}_{\text{vel}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \left\| (x_0^{i+1} - x_0^i) - (\hat{x}_0^{i+1} - \hat{x}_0^i) \right\|_2^2, \tag{2.18}$$

and foot contact loss

$$\mathcal{L}_{\text{foot}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \left\| (FK(\hat{x}_0^{i+1}) - FK(\hat{x}_0^i)) * f_i \right\|_2^2. \tag{2.19}$$

For the position loss and foot contact loss, it is necessary to transform the model's predicted joint rotations into joint positions (denoted as FK($\cdot$)), since distances are calculated in 3D space. In the case of the foot contact loss, $f_i$ denotes the binary foot contact mask, which indicates whether each foot joint is in contact with the ground (cf. Equations 3–5 in [29]). Taken together, these losses form the overall loss function (cf. Equation (6) in [29]):

$$\mathcal{L} = \mathcal{L}_{\text{simple}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{vel}}\mathcal{L}_{\text{vel}} + \lambda_{\text{foot}}\mathcal{L}_{\text{foot}}. \tag{2.20}$$

Although these optimizations are sophisticated and require domain knowledge, their contribution to improving motion quality can be modest in some cases (cf. Table 4 in [29]).

## 2.3 Summary and Research Question

This chapter has highlighted advances in diffusion-based models, spanning from fundamental improvements in training and sampling to more advanced optimization tech-

niques, such as importance sampling and EMA, which contribute to training stability and enhanced training dynamics. Over the years, these improvements have led to significant performance acceleration, making diffusion models the state-of-the-art generative approach, outperforming GANs [4].

Simultaneously, researchers in 3D human motion synthesis have explored various architectural designs (see Table B.1 for a comprehensive overview of methods), adopted more suitable pose representations, and tailored specific loss functions to generate high-quality samples. These models are commonly evaluated using metrics such as Diversity, FID, Multimodality, and Recognition Accuracy. With the rise of diffusion models, Tevet et al. [29] introduced a lightweight framework that achieves state-of-the-art results across multiple conditional modes.

However, the action-conditioned motion synthesis task lacks sufficient datasets, with UESTC being the only large-scale dataset with a train/test split. This scarcity of data along with the issue of UESTC providing monocular videos, which have to be converted using estimation methods, leaves room for improvement in the quality of generated motions.

While the majority of research focuses on specialized architectural solutions, complex loss functions (e.g., geometric losses), and conditional modalities, more general optimization techniques—such as importance sampling and EMA—remain underrepresented. Notably, only Chen et al. [3] mention EMA, but it is not explicitly integrated into their experiments.

Given that importance sampling and EMA are task-agnostic and relatively easy to integrate into any diffusion-based model, this thesis investigates their influence on action-conditioned motion synthesis. Building on the lightweight MDM framework proposed by Tevet et al. [29], the corresponding research question is formulated as follows:

> To what extent do generic training dynamics and optimization strategies—particularly importance sampling and EMA—affect the effectiveness and efficiency of action-conditioned motion diffusion models?

To examine this research question, experiments are conducted on a modified baseline model[1] derived from MDM simplified for the action-conditioned task, as well as on versions integrating importance sampling and EMA.

[1] The term baseline model in this thesis is used to refer to the MDM model adapted for the action-conditioned task, without the application of any optimization strategies.

# 3 Methodology

This chapter presents the methodology used to conduct experiments on the proposed model, focusing on evaluating the effectiveness of EMA and importance sampling. Section 3.1 describes the architectural design and workflow, outlining how MDM was adapted and simplified for the action-conditioned motion synthesis task. Since this work employs the UESTC dataset introduced in Section 2.2.1, Section 3.2 details the preprocessing pipeline used to convert the raw video data into sequences represented in the 6D rotation format.
Building upon these foundations, Section 3.3 explains the model's training procedure, including implementation details, hyperparameter configurations derived from an initial pilot phase, and the applied settings for importance sampling and EMA. Section 3.4 describes the sampling process of the proposed model, justifies the chosen denoising procedure, and the weighting of the utilized evaluation metrics.
Finally, Section 3.5 outlines the experimental setup, which is structured around three phases of experiments, and explains the workflow connecting them.

## 3.1 Model Architecture

Building upon the architectural foundations and domain-specific design considerations discussed in Section 2.2, this section details the implementation of the model used in this thesis. The design is derived from the state-of-the-art MDM architecture [29], adapted and simplified to focus exclusively on action-conditioned motion synthesis.

The backbone of the model is a transformer encoder, originally introduced by Vaswani et al. [31]. The input to the transformer is twofold and comprises a sequence of motion frames $x_t^{1:N}$ of length $N$, which are first passed through a linear layer to match the model dimension. Additionally, an action label $c$ (i.e., the conditioning signal) and the current diffusion timestep $t$ are embedded and concatenated to form the $z_{tk}$ token

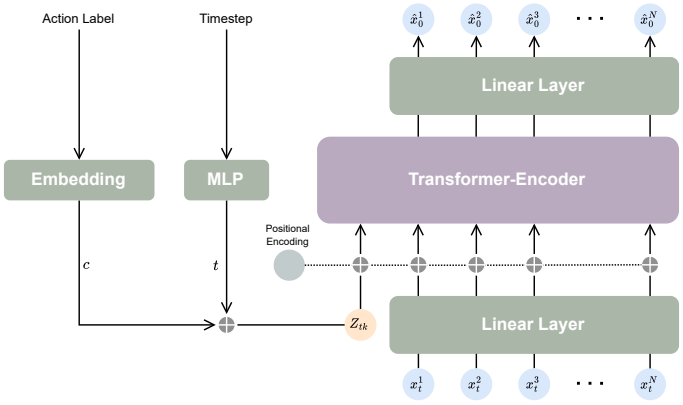



Figure 3.1: Overview of the modified model architecture (adapted from Figure 2 in [29]).

(see Figure 3.1). The sequence tokens, together with the $z_{tk}$ token, are then augmented with positional encodings and passed to the transformer encoder. The output of the transformer is subsequently processed by a final linear layer to match the output dimension. Note that the number of output tokens equals the number of sequence tokens, as the $z_{tk}$ token only serves as a conditioning token and is not reconstructed at the output. Ultimately, the only modification to the model architecture compared to the text-conditioned version of MDM is the replacement of the text encoder with an action embedding layer.

The hyperparameters of the model architecture are set to match those of the original MDM model [29], comprising 8 attention heads, 8 layers, a latent dimension of 512, and a feedforward dimension of 1,024. As a result, the model consists of approximately $17 \times 10^6$ parameters. An extensive overview of all hyperparameters can be found in Appendix C.2.

## 3.2 Dataset Preprocessing

This research utilizes the UESTC dataset [8]. Unlike many other datasets, UESTC provides predefined train/test splits [17] and emphasizes more complex and dynamic movement patterns. It is of particular relevance to this thesis due to its widespread adoption in recent motion synthesis research (see Table B.1). Moreover, state-of-the-art performance on this dataset remains open for improvement—especially regarding the

FID metric, where the best reported test score to date is 12.81, achieved by the MDM model [29]. As a reference point, the FID score of the ground truth data is 2.79 (see Table 4 in [29]), indicating significant room for advancement.

The preprocessing pipeline follows the approach introduced by Petrovich et al. [17] and subsequently adopted by Tevet et al. [29]. All video sequences are converted into 3D motion representations using the Video Inference for Body Pose and Shape Estimation model [11], which estimates SMPL parameters from monocular RGB videos. In scenes containing multiple subjects, the track most closely corresponding to the provided Kinect skeleton is selected to ensure consistency. Only sequences recorded from the eight static camera viewpoints are retained, while those captured by the rotating camera are excluded. To maintain viewpoint consistency across samples, all motions are canonicalized to a frontal orientation. Following the official Cross-subject protocol, subjects are divided into training and testing sets. This protocol is preferred over the Cross-view split, as viewpoint variations can be easily simulated by the model [17].

In addition to data preprocessing, evaluation features are extracted using a recognition model provided by the MDM implementation [29], which follows the design introduced by Petrovich et al. [17]. The model operates on pose parameters expressed as 6D rotations, offering greater stability with respect to viewpoint variations than joint-based representations. Using this pretrained model ensures consistent and comparable feature embeddings for computing evaluation metrics such as FID and recognition accuracy.

The final dataset used for training comprises approximately 10,650 sequences (roughly 33 sequences per action on average when normalized by viewpoint), with 13,350 sequences reserved for testing. By applying the same preprocessing steps and maintaining the same train/test split as prior works, this thesis ensures a fair comparison of results with existing state-of-the-art methods.

## 3.3 Training Procedure

The training pipeline of the proposed model largely follows the original MDM framework [29], with several adaptations introduced to improve training stability and sample efficiency. This section describes the main components of the training process and the overall optimization setup, including the learning rate schedule, loss weighting strategy, and stabilization techniques such as EMA and importance sampling.

As an initial step, all hyperparameters were set to match those of the original MDM model. However, since a detailed hyperparameter list for training on the UESTC dataset was not available, an initial phase of pilot experiments was conducted to identify a suitable configuration. Due to the complexity of the model and the resulting large number of hyperparameters involved, an exhaustive overview of hyperparameter settings is provided in Appendix C. The following discussion focuses on the key differences and adaptations introduced in the training procedure.

A key component of the training process is the use of geometric loss functions as discussed in Section 2.2.4. In the original formulation [29], the weights of each geometric loss are set to either 0 or 1. However, pilot experiments revealed that the model's geometric losses were very small in magnitude, and using the foot contact loss did not noticeably improve performance. Moreover, including the position loss appeared to conflict with the primary mean squared error objective, so it was set to $\lambda_{\text{pos}} = 0$. The remaining weights were adjusted empirically to $\lambda_{\text{vel}} = 25.0$ and $\lambda_{\text{foot}} = 45.0$, to achieve a meaningful contribution to the overall loss landscape.

To further enhance convergence, a learning rate schedule combining a linear warmup and cosine annealing was employed. The learning rate linearly increases from 0.0 to $1.2 \times 10^{-4}$ during the first 5,000 steps and subsequently decays following a cosine schedule to a minimum value of $10^{-5}$ over the full training duration. Although this adjustment is not strictly required for training stability, it was found empirically to improve generalization and convergence speed.

Another observation during the pilot phase was that the dropout parameter, set to 0.1 in the original implementation [29], appeared to degrade performance. Empirically, reducing it to 0.0 resulted in slightly improved results. As these hyperparameter adjustments are not the main focus of this thesis, the observed improvement was considered sufficient without conducting an extensive ablation study, thereby keeping the experimental scope manageable.

Following the approach proposed in [29], the model was trained for a total of $2 \times 10^6$ training steps. Moreover, the diffusion process was configured with 1,000 timesteps. However, subsequent experiments (see Section 3.5) also evaluate adjusted training and diffusion steps. Checkpoints to save the model and EMA parameters, as well as the optimizer state are recurrently done after $2 \times 10^5$ training steps. Following directly after checkpointing, the current model is evaluated.

The utilization of importance sampling follows the implementation of Nichol and Dhariwal [16]. During the forward propagation, importance sampling produces a tuple of sampled timesteps and their associated weights for each batch element. The model loss is then evaluated at these sampled timesteps and weighted accordingly during the backward propagation. This implementation is straightforward, as the sampling and weighting operations can be easily replaced with uniform sampling assigning equal weights for all timesteps.

EMA[1] is instantiated with the model parameters and a decay rate, which is typically set close to 1.0 [16]. The exact decay rate is treated as a tunable hyperparameter and is discussed further in Section 3.5. During training, the EMA parameters are updated after each optimization step in the backward propagation phase. These averaged parameters are saved alongside the model checkpoints to be utilized during evaluation and inference. Evaluation with EMA is done after training, as it does not require retraining, utilizing the saved checkpoints.

Finally, all models were trained on a server equipped with ten NVIDIA Quadro P6000 GPUs, two 18-core CPUs, and 384 GB of system memory. In practice, 8 GPUs were utilized concurrently using Distributed Data Parallel[2], with an effective global batch size of 64.

## 3.4 Evaluation Procedure

To assess the performance of the proposed model, this section outlines the procedure used for generating motion sequences and evaluating their quality. Sampling from the diffusion model is an integral part of the evaluation, as the quality of the generated sequences directly impacts the metrics used to measure model performance. In the following the sampling process, the evaluation metrics utilized and the according hyperparameters are discussed.

Sampling from the diffusion model (as depicted in Figure 3.2) is performed by the reverse process described in Section 2.1. Based on a condition $c$ and random noise $x_T$ sampled from a Gaussian distribution along with the desired dimension of the output sequence, the model iteratively denoises the input. At each step $t$, the model predicts the original

[1] https://github.com/fadel/pytorch_ema (accessed 10/29/2025)
[2] https://docs.pytorch.org/docs/stable/generated/torch.nn.parallel.DistributedDataParallel.html (accessed 10/29/2025)

sequence $\hat{x}_0$ from the noisy input $x_t$ and applies noise according to the next timestep $t-1$. Although, as described in Section 2.1.2 predicting $\epsilon$ has been shown to yield better results in image synthesis, MDM chooses to predict $\hat{x}_0$ (i.e., the desired motion sequence) directly, due to the application of geometric losses.

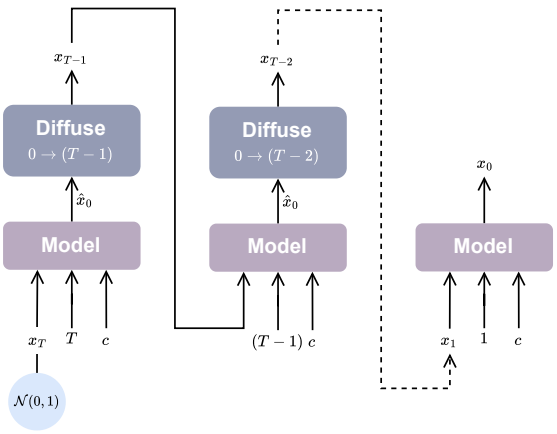



Figure 3.2: Overview of the samplng process (adapted from Figure 2 in [29]).

While Section 2.1.3 introduced alternative sampling strategies, this thesis employs the original DDPM denoising process. In particular, DDIM [26] is not adopted, following the findings of Zhou et al. [35], who report that DDIM tends to degrade sample quality for motion data. Furthermore, the Improved DDPM variant [16] reduces the number of diffusion steps, however, since Tevet et al. [29] already provided additional experiments[3] with fewer diffusion steps—which will be discussed in Section 3.5—after the initial submission, sampling time can be reduced accordingly without deviating from the original DDPM framework.

Following the evaluation protocols provided by Petrovich et al. [17] and Tevet et al. [29], the generation of motion sequences is restricted to a fixed length of 60 frames. This constraint ensures a fair and consistent comparison across models by aligning the temporal duration of the generated motions with the reference implementation.

To assess generated samples the four metrics introduced in Section 2.2.2—namely FID, Recognition Accuracy, Diversity, and Multimodality—are utilized. Since, most studies

[3] https://openreview.net/forum?id=SJ1kSyO2jwu (accessed 10/29/2025)

consider FID as their principal metric and due to its capability to indicate overall motion quality in terms of smoothness and plausibility, this thesis treats it the same way. Consistent with the evaluation procedure of Tevet et al. [29] for the UESTC dataset, this thesis generates 1,000 samples for each of 20 distinct seeds (seed values 1 to 20) to ensure a fair comparison of results with existing methods.

## 3.5 Experimental Setup

The experimental setup is designed specifically to investigate the research question outlined in Section 2.3. To evaluate the impact of importance sampling and EMA thoroughly, the experiments are conducted in three different phases, which will be introduced in the following.

#### Phase 1: Baseline Model and Diffusion Steps Comparison

This phase serves to validate the performance of the baseline model by comparing it to MDM [29]. Since Tevet et al. [29] have provided additional experiments reducing diffusion steps down to 100, the comparison will comprise evaluations of trainings with 1,000 and 100 diffusion steps. The 100-step setting is of particular interest because Tevet et al. [29] did not evaluate diffusion step reduction on the UESTC dataset. Moreover, it significantly reduces computation, which might be beneficial for the following phases. Hence, if the performance is close to the 1,000-step setting, the 100-step setting will be preferred for the other phases. Following Tevet et al. [29], experiments are run for $2 \times 10^6$ training steps, using the same hyperparameters except for those discussed in Section 3.3.

#### Phase 2: Importance Sampling and EMA Hyperparameter Exploration

This phase evaluates different decay rate settings for EMA, the addition of importance sampling to the baseline model, and their combination. To reduce computation, training steps are limited to $1 \times 10^6$, and the cosine annealing schedule is adjusted accordingly. Since EMA is evaluated on checkpoints created during training, these evaluations can be compared directly to the baseline model performance at $1 \times 10^6$ steps without retraining.

The EMA decay rates are set to 0.9999 and 0.9995, following the recommendations of [16].

**Phase 3: Final Evaluation with Optimized Settings**

In this final phase, the most promising configuration identified in Phase 2 is selected to perform a full-scale evaluation over $2 \times 10^6$ training steps. The primary goal is to assess the cumulative impact of importance sampling and EMA on the final model performance. The selection of the best configuration is based primarily on the FID metric, as it provides a comprehensive measure of motion quality in terms of smoothness and plausibility.
All other hyperparameters are kept consistent with the baseline training protocol described in Section 3.3, ensuring a fair comparison with both the original baseline model and the intermediate results from Phase 2. This phase serves as the definitive evaluation to demonstrate the effectiveness of the optimized training strategies and to quantify improvements achievable with the selected settings.

# 4 Evaluation

This chapter presents the evaluation of the conducted experiments across the phases described in Section 3.5. All experiments involve different configurations of the baseline model, including variations in the number of diffusion steps and the application of optimization techniques such as EMA and importance sampling.

Section 4.1 presents the results obtained across these phases, using quantitative metrics (tables) alongside qualitative visualizations (plots and heatmaps). Subsequently, Section 4.2 interprets the results and discusses their implications.

## 4.1 Results

Section 4.1.1 presents the results of Phase 1, validating the performance of the proposed model. Section 4.1.2 reports the effects of applying EMA and importance sampling individually and combined, while Section 4.1.3 presents the final experiment, using the most promising configuration identified in Phase 2.

### 4.1.1 Phase 1: Baseline Model and Diffusion Steps Comparison

This phase aims to validate the baseline model's performance and examine the influence of the number of diffusion steps on generation quality. Table 4.1 summarizes the results, including ground truth metrics (Real) from the UESTC dataset. The proposed model is denoted as *Proposed* with subscripts indicating the diffusion steps used during training.

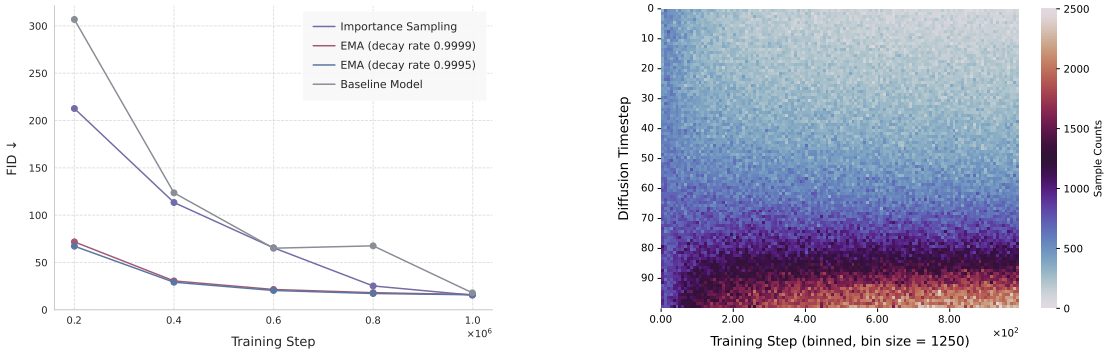
The model trained with $1,000$ diffusion steps achieves comparable results to MDM on $\text{FID}_{test}$, Diversity, and Multimodality, with higher $\text{FID}_{train}$ (12.37 compared to MDM's 9.98) and lower Accuracy (0.91 compared to MDM's 0.95). The 100-step variant shows improved $\text{FID}_{train} = 10.30$ and Accuracy $= 0.94$, closely matching MDM.

| Method | $\text{FID}_{train} \downarrow$ | $\text{FID}_{test} \downarrow$ | Accuracy$\uparrow$ | Diversity$\rightarrow$ | Multimodality$\rightarrow$ |
|---|---|---|---|---|---|
| Real | $2.92^{\pm.26}$ | $2.79^{\pm.29}$ | $0.988^{\pm.001}$ | $33.34^{\pm.32}$ | $14.16^{\pm.06}$ |
| INR [2] | $\mathbf{9.55^{\pm.06}}$ | $15.00^{\pm.09}$ | $0.941^{\pm.001}$ | $31.59^{\pm.19}$ | $14.68^{\pm.07}$ |
| MDM [29] | $9.98^{\pm1.33}$ | $12.81^{\pm1.46}$ | $\mathbf{0.950^{\pm.000}}$ | $33.02^{\pm.28}$ | $\mathbf{14.26^{\pm.12}}$ |
| $\text{Proposed}_{1000}$ | $12.37^{\pm0.86}$ | $\mathbf{12.59^{\pm1.22}}$ | $0.91^{\pm0.01}$ | $32.24^{\pm0.51}$ | $14.89^{\pm0.35}$ |
| $\text{Proposed}_{100}$ | $10.30^{\pm0.98}$ | $13.31^{\pm0.93}$ | $0.94^{\pm0.01}$ | $\mathbf{33.49^{\pm0.55}}$ | $14.04^{\pm0.20}$ |

Table 4.1: Evaluation results of Phase 1 (Adapted from Table 4 in [29]). Validation of the baseline model trained with different numbers of diffusion steps. Lower values ($\downarrow$) indicate better performance for FID; higher values ($\uparrow$) are preferable for Accuracy; and values closer to the real data ($\rightarrow$) are desirable for Diversity and Multimodality. The results closest to the ground truth (Real) are highlighted in bold and reported with mean $\pm$ standard deviation.

### 4.1.2 Phase 2: Importance Sampling and Exponential Moving Average Hyperparameter Exploration

This phase evaluates the model's performance under the application of EMA, importance sampling, and their combination. All experiments in this phase use the 100 diffusion timestep setting established in Phase 1. Figure 4.1a shows the development of the FID metric over the course of training, comparing importance sampling and two EMA configurations with decay rates of 0.9999 and 0.9995 against the baseline model. While the baseline model exhibits the highest initial FID value, slightly above 300, the addition of importance sampling reduces it to just above 200. In contrast, both EMA variants start with substantially lower initial FID values, around 70. Both importance sampling and EMA lead to smoother curves throughout training. Although the baseline model's FID initially decreases, its improvement deteriorates slightly after approximately $6 \times 10^5$ training steps. Beyond this point, the FID values of all models converge to similar levels. Figure 4.1b presents a heatmap showing the relationship between diffusion timesteps and training steps, where the latter are grouped into bins of size 1250 for visualization purposes. The heatmap illustrates how frequently each diffusion timestep is sampled throughout the training process. After an initial warm-up phase, importance sampling begins to favor higher diffusion timesteps (approximately $\gtrsim 80$) more frequently than lower ones (around $\lesssim 20$). Around $1,5 \times 10^5$ training steps, timesteps above 80 are sampled roughly $\gtrsim 2000$ times, while intermediate timesteps between 80 and 90 reach counts of about 1250, and lower timesteps remain below 1000. This trend remains consistent throughout training, with timesteps near the maximum value of 100 eventually being sampled up to approximately 2500 times.




(a) Development of $\text{FID}_{test}$ during training ($1 \times 10^6$ steps) for EMA, importance sampling, and the baseline model.

(b) Heatmap displaying the frequency of diffusion timestep samples collected during training ($1 \times 10^6$ steps).

Figure 4.1: Comparison of training progression and sampling behavior across different methods (Phase 2).

Table 4.2 shows a detailed overview of the model performance applying EMA, importance sampling, and their combination compared against the baseline. EMA with a decay rate of 0.9999 is denoted as $\text{EMA}_A$, EMA with a decay rate of 0.9995 is denoted as $\text{EMA}_B$, and importance sampling is denoted as IS. Regarding the isolated application of EMA

| Method | $\text{FID}_{train} \downarrow$ | $\text{FID}_{test} \downarrow$ | Accuracy↑ | Diversity→ | Multimodality→ |
|---|---|---|---|---|---|
| Real | $2.92^{\pm.26}$ | $2.79^{\pm.29}$ | $0.988^{\pm.001}$ | $33.34^{\pm.32}$ | $14.16^{\pm.06}$ |
| Baseline | $18.287^{\pm1.081}$ | $17.904^{\pm1.318}$ | $0.886^{\pm0.010}$ | $31.884^{\pm0.550}$ | $15.089^{\pm0.308}$ |
| $\text{EMA}_A$ | $12.983^{\pm0.990}$ | $16.074^{\pm0.948}$ | $0.916^{\pm0.011}$ | $33.717^{\pm0.517}$ | $14.218^{\pm0.269}$ |
| $\text{EMA}_B$ | $12.688^{\pm1.075}$ | $15.885^{\pm0.929}$ | $0.913^{\pm0.012}$ | $33.592^{\pm0.526}$ | $14.377^{\pm0.266}$ |
| IS | $14.317^{\pm1.205}$ | $15.507^{\pm1.216}$ | $0.902^{\pm0.009}$ | $32.726^{\pm0.566}$ | $14.889^{\pm0.270}$ |
| $\text{EMA}_A$ + IS | $12.997^{\pm1.054}$ | $15.760^{\pm0.931}$ | $\mathbf{0.919^{\pm0.011}}$ | $33.675^{\pm0.533}$ | $14.039^{\pm0.280}$ |
| $\text{EMA}_B$ + IS | $\mathbf{12.535^{\pm1.083}}$ | $\mathbf{15.020^{\pm0.948}}$ | $\mathbf{0.919^{\pm0.011}}$ | $\mathbf{33.539^{\pm0.572}}$ | $\mathbf{14.204^{\pm0.258}}$ |

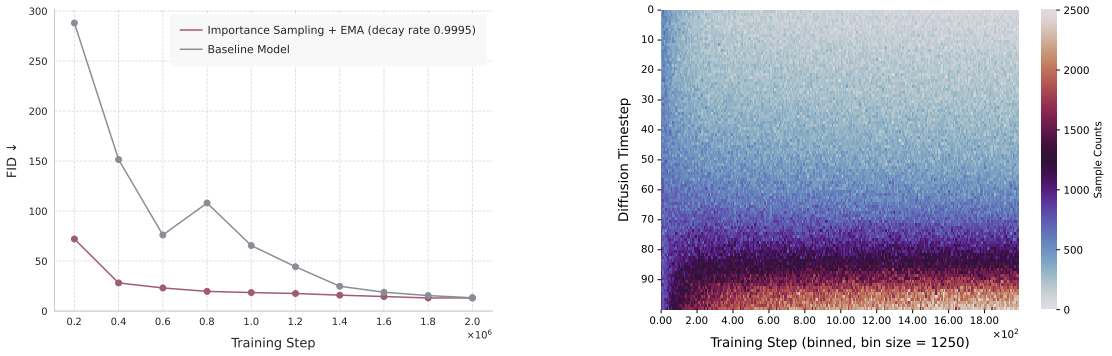
Table 4.2: Evaluation Results of Phase 2. Effect of introducing importance sampling (denoted as IS) and EMA on quantitative metrics with training steps reduced to $1 \times 10^6$.

and importance sampling, the latter achieves a slightly lower $\text{FID}_{test}$ value (15.507) than either EMA variant. All configurations yield substantially lower $\text{FID}_{test}$ values than the baseline, with an improvement of approximately $\gtrsim 1.8$. In terms of Diversity and Multimodality, EMA produces results closer to the ground truth than importance sampling. For example, $\text{EMA}_A$ exhibits a Multimodality gap of 0.058 compared to the real data, whereas importance sampling shows a larger gap of 0.729.

Regarding the combined application of EMA and importance sampling, the configuration using a decay rate of 0.9999 yields a slightly higher $\text{FID}_{test}$ value (15.760) than the isolated application of importance sampling (15.507). The combination utilizing a decay rate of 0.9995 achieves the closest results to the ground truth across all metrics, outperforming all other configurations.

### 4.1.3 Phase 3: Final Evaluation with Optimized Settings

This phase evaluates the model's performance using EMA in combination with importance sampling. The experiment in this phase uses the 100 diffusion timestep setting established in Phase 1, and is trained for $2 \times 10^6$ steps, with an EMA decay rate of 0.9995. Figure 4.1a shows the development of the FID metric over the course of training, comparing the combined application of EMA and importance sampling with the baseline model, which exhibits a regressive phase after $6 \times 10^5$ and its peak at $8 \times 10^5$ training steps, eventually converging to a value below approximately 20. The enhanced model follows a smooth curve without any regression or plateau. Eventually, both models converge to similar levels. Figure 4.2b presents a heatmap analogous to Figure 4.1b,

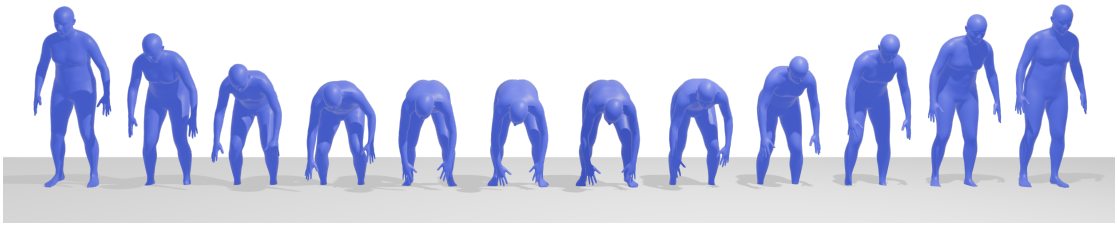



(a) Development of $\text{FID}_{test}$ during training ($2 \times 10^6$ steps) for EMA combined with importance sampling, and the baseline model.

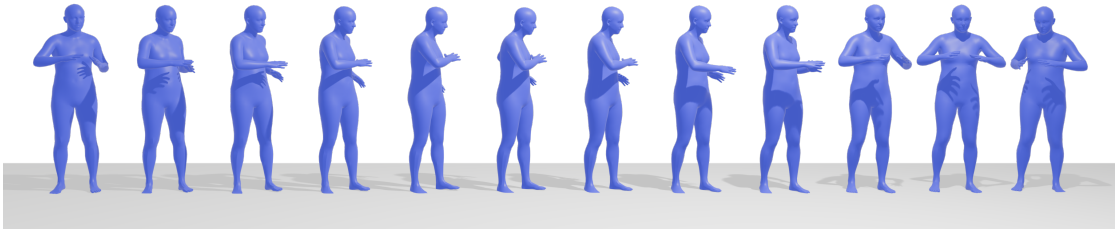
(b) Heatmap displaying the frequency of diffusion timestep samples collected during training ($2 \times 10^6$ steps).

Figure 4.2: Comparison of training progression and sampling behavior across different methods (Phase 3).

showing that the trend observed in Phase 2 continues consistently over the full $2 \times 10^6$ training steps. Table 4.3 presents an updated overview of model performance, extending the Phase 1 results with the conducted experiment in Phase 3. The combination of EMA

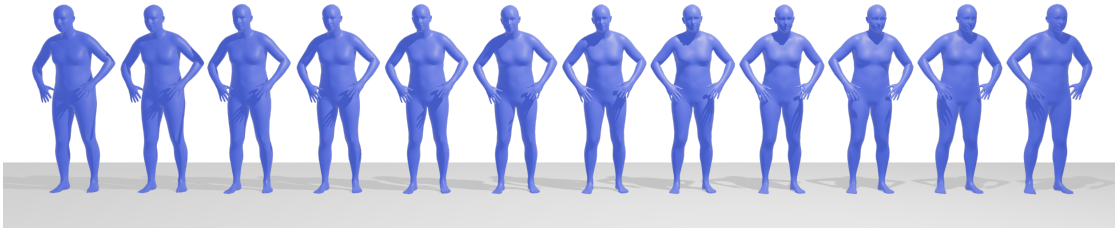


(a) Standing toe touches



(b) Standing rotation

(c) Left stretching



(d) Head anticlockwise circling

Figure 4.3: Visualization of four rendered pose sequences. Starting from frame 1 (leftmost), every third frame up to frame 34 (rightmost) is shown. For illustration purposes, the joint rotations generated by the model are rendered as SMPL meshes.

| Method | $\text{FID}_{train} \downarrow$ | $\text{FID}_{test} \downarrow$ | Accuracy↑ | Diversity→ | Multimodality→ |
|---|---|---|---|---|---|
| Real | $2.92^{\pm.26}$ | $2.79^{\pm.29}$ | $0.988^{\pm.001}$ | $33.34^{\pm.32}$ | $14.16^{\pm.06}$ |
| INR [2] | $9.55^{\pm.06}$ | $15.00^{\pm.09}$ | $0.941^{\pm.001}$ | $31.59^{\pm.19}$ | $14.68^{\pm.07}$ |
| MDM [29] | $9.98^{\pm1.33}$ | $12.81^{\pm1.46}$ | $\mathbf{0.950^{\pm.000}}$ | $33.02^{\pm.28}$ | $14.26^{\pm.12}$ |
| $\text{Proposed}_{1000}$ | $12.37^{\pm0.86}$ | $\mathbf{12.59^{\pm1.22}}$ | $0.91^{\pm0.01}$ | $32.24^{\pm0.51}$ | $14.89^{\pm0.35}$ |
| $\text{Proposed}_{100}$ | $10.30^{\pm0.98}$ | $13.31^{\pm0.93}$ | $0.94^{\pm0.01}$ | $\mathbf{33.49^{\pm0.55}}$ | $14.04^{\pm0.20}$ |
| $\text{Proposed}_{EMA+IS}$ | $\mathbf{9.468^{\pm0.819}}$ | $13.166^{\pm0.826}$ | $0.939^{\pm0.008}$ | $33.783^{\pm0.487}$ | $\mathbf{14.120^{\pm0.216}}$ |

Table 4.3: Evaluation results of Phase 3 (Adapted from Table 4 in [29]). Comparison of the proposed model trained with EMA and importance sampling (EMA+IS) against results of Phase 1.

and importance sampling over $2 \times 10^6$ training steps yields similar results compared to the proposed model trained with 100 diffusion steps, with an improvement in $\text{FID}_{train}$ from 10.30 to 9.468, and a reduction in the standard deviation across all metrics except for Multimodality.

Figure 4.3 visualizes four different pose sequences conditioned on actions standing toe touches, standing rotation, left stretching, and head anticlockwise circling. Each sequence displays 12 frames starting from frame 1 and showing every third frame up to frame 34. As illustrated in Figures 4.3a, 4.3b, and 4.3d, the model tends to generate sequences that initiate from a non-neutral initial pose. Moreover, in Figure 4.3a, the model reveals violations of physical plausibility, as the generated poses of frames 2 to 11 show the subject's feet penetrate the ground plane. This phenomenon can also be observed in some frames of Figures 4.3b and 4.3c. In addition, Figure 4.3b indicates the model's difficulty in performing rotations without foot sliding, a prevalent artifact in human motion synthesis [37]. As the subject's upper body rotates to the left, the feet exhibit a slight lateral movement in the same direction. Figure 4.3d represents an exemplary sequence of a fine-grained movement. The sequence exhibits the model's difficulty in performing a clear circular motion with the head, as the trajectory is hardly recognizable and most likely performed by the entire upper body instead of isolating the head movement. In contrast to those limitations, Figures 4.3a, 4.3b, and 4.3c exhibit smooth transitions of the upper body between poses. In particular, the arm movements in Figures 4.3a and 4.3c appear natural and fluid. Moreover, in case no movement of the legs is required, as in Figures 4.3b and 4.3d, the feet barely penetrate the ground plane. Overall, with exception of Figure 4.3d, the generated pose sequences appear consistent with the given action conditions.

## 4.2 Discussion

The results of Phase 1 illustrated in Table 4.1 indicate that the initial implementation, including the piloting phase optimizations, exhibited comparable performance to that of the reference model by Tevet et al. [29]. In particular, the 100-step model produced results closer to those of the reference model. Despite a slight drop in $\text{FID}_{test}$, it performed better than the 1000-step model. One possible explanation is that, unlike image-based generative tasks, DDPMs for motion synthesis do not necessarily benefit from a higher number of diffusion steps. The reduced dimensional space implies lower diversity, which potentially allows the model to learn the underlying distribution more easily. Additionally, motion data exhibits strong temporal correlations that transformer architectures can effectively leverage. Together, the observations from Phase 1 led to the selection of the 100-step model for Phases 2 and 3.

Phase 2 investigated the effects of EMA, importance sampling, and their combination. To enable comparison across multiple configurations, the total number of training steps was reduced to $1 \times 10^6$. Consequently, the results of Phase 2 are not directly comparable to those of Phase 1. As expected, the baseline model showed a notable drop in all quantitative metrics due to the reduced training duration (see Table 4.2).
Both EMA and importance sampling independently demonstrated significant improvements of $\text{FID}_{train}$ and $\text{FID}_{test}$. A comparison of FID development throughout training in Figure 4.1a exhibits the ability of EMA in not only stabilizing the training process, but also yielding significantly lower FID values than the baseline model early on during training. This can be explained by the smoothing effect of EMA, as it effectively averages the weights throughout training. Especially in the early stages of training, where the model traverses unstable loss landscapes, the benefit of EMA lies in preventing large weight updates from affecting its weights. Thus, the curve shows steady improvement, unaffected by instabilities and deterioration.
Ultimately, the combination of importance sampling and EMA, with a decay rate of 0.9995, achieved the best overall results across all quantitative metrics. This outcome suggests that the two optimizations complement each other effectively. While EMA enhances diversity and multimodality, both methods contribute to lowering the FID values. Furthermore, the focus on higher diffusion timesteps, which is induced by importance sampling, does not appear to have a negative impact on the benefits of EMA.

Building on these findings, Phase 3 aimed to compare the best-performing configuration from Phase 2, which is the combination of importance sampling and EMA with a decay

rate of 0.9995, with the reference model by Tevet et al. [29]. Unlike Phase 2, the training steps were set to $2 \times 10^6$. This ensured comparability with the results of Phase 1 and the reference model.
As demonstrated in Table 4.3 and Figure 4.2a, the proposed optimization has a negligible impact on the overall performance, as indicated by the quantitative and qualitative results, respectively. The heatmap in Figure 4.2b further confirms the trend observed in Phase 2. Thus, the model does not learn to effectively leverage insights provided by noisier samples, although it was expected to gradually shift its focus towards lower steps. In the long run, this might hinder the model from leveraging the full potential of the geometric losses. Especially the foot loss requires lower diffusion timesteps to effectively guide the model, as it operates on a fine-grained level. This observation suggests that the model's capacity to leverage these observations is reduced when importance sampling overemphasizes higher timesteps.
Furthermore, the impact of EMA expectedly vanishes over time. As demonstrated in Figure 4.2a, the curve exhibits a rapid convergence, yet EMA's role in smoothing becomes less significant as the training duration increases. This is due to the fact that EMA exponentially weights older model parameters less. Thus, as training progresses, the smoothed parameters are updated with weights of the baseline model that has already stabilized. As indicated by the curve of the baseline model, the training in motion synthesis appears to be relatively stable in general. Consequently, the effect of EMA becomes negligible in the long run.
Ultimately, the visualization of multiple samples in Figure 4.3 underlines the aforementioned conclusions. While coarse-grained movements, particularly those of the upper body, appear to be physically plausible, the finer-grained movements demonstrate the limitations of the model. Even with a significantly higher weighting than other losses, foot contact loss still leads to artifacts such as foot sliding and ground-plane penetration. Additionally, the anticlockwise circling sequence of the head shows an unclear movement. While Ji et al. [8] emphasize that the UESTC dataset contains challenging movements, the applied optimizations appear insufficient in addressing these challenges effectively.

To contextualize the findings across all phases, several limitations of this thesis must be acknowledged. Conditioning on action labels is a simple approach to guide motion, yet the lack of context may introduce the issue of ambiguous mappings from actions to motion. In particular, categories such as "kick" rely on the context to determine, whether to kick a ball, or perform a kick in combat sport. However, for this thesis, and the UESTC dataset in particular, this limitation is negligible, as each category is distinct.

The UESTC dataset [8] was selected for its predefined train/test splits and diverse set of complex action categories. However, state-of-the-art models such as MDM by Tevet et al. [29] have already demonstrated difficulties in producing physically plausible motions with respect to the FID. As a result, the observation of severe artifacts was expected throughout experimentation. Additionally, the dataset size may not have been sufficient for the model to effectively learn the complex motions it contains. The applied train/test split was somewhat unconventional, as the test set included more samples than the training set, while no separate evaluation set was provided. Nevertheless, the original split was retained to ensure comparability.

While following the suggestions of Tevet et al. [29] in reducing diffusion timesteps, it remains uncertain whether 100 diffusion steps are the best tradeoff between training and sampling acceleration and quantitative results. However, since the overall goal of this thesis was not to achieve state-of-the-art performance but rather to investigate the effects of EMA and importance sampling, and the effort of an extensive ablation study was beyond the scope of this work, this limitation was accepted.

Furthermore, importance sampling as implemented by Nichol and Dhariwal [16] was originally designed for image generation tasks. Its applicability to motion synthesis thus remains uncertain. The method also allows adaptive adjustment of sampling probabilities across diffusion timesteps, which was not explored in this thesis to maintain the scope manageable.

To employ geometric losses, the diffusion model was trained to predict the original sample $x_0$. While this approach enables the integration of losses introducing physical constraints, it may hinder the exploration of predicting the noise $\epsilon$ instead, which has been shown to improve sample quality in image generation tasks [16]. Moreover, these losses are empirically computationally expensive, as they require to transform the data representation from 6D rotation matrices to joint positions.

As the visualization of sequences in Figure 4.3 indicates, it is beneficial to evaluate generated motions qualitatively through user studies. While quantitative metrics provide objective measures, the absence of user studies may exclude the identification of disturbing artifacts and the determination of which aspects of motion could be improved.

# 5 Summary & Outlook

This thesis investigated the effectiveness and efficiency of generic training dynamics and optimization strategies—specifically importance sampling and EMA—in enhancing the performance of a diffusion-based motion synthesis model on the UESTC dataset. The proposed model, derived from the MDM framework, was restricted to the action-conditioned motion synthesis task and applied to the UESTC dataset, which features complex and fine-grained human movements. As the dataset continues to present challenges to achieving state-of-the-art performance including the original MDM implementation—it offers a suitable basis for evaluating potential optimizations. On a broader scale, the work also provides insights into the applicability of generated motions as a means to augment existing motion datasets.

The results showed that the proposed model, enhanced through the application of importance sampling and EMA with a decay rate of 0.9995, achieved performance comparable to the reference model MDM. Slight improvements in $\text{FID}_{train}$ and motion diversity were observed but remain negligible overall. While the optimization techniques appeared to accelerate convergence during the early stages of training, this effect diminished over time. The smoothing behavior of EMA contributed to more stable training; however, since the baseline process was already relatively stable, its overall benefit appears limited. Importance sampling encouraged the model to emphasize higher diffusion steps, but this focus also led to reduced attention to lower diffusion steps, potentially compromising physical plausibility, particularly with respect to foot contact consistency.

Visual inspection of the generated samples further highlights the need for user studies to assess perceptual realism. Quantitative metrics alone fail to capture specific artifacts such as foot sliding or ground-plane penetration, which are easily perceptible to human observers. Since human perception is highly sensitive to even subtle errors, subjective evaluations could provide more reliable guidance for future refinements.

While the primary focus of this work lay in evaluating the FID metric as an indicator of realism, the results concerning motion diversity and multimodality suggest that the synthetic data generated by diffusion-based models hold promise for augmenting existing motion datasets. Nevertheless, for complex and fine-grained actions, further research should aim to improve the naturalness and physical plausibility of generated motion to ensure its effectiveness and comparability with real motion data.

Building upon the proposed method, several directions for future research can be identified. Importance sampling could be refined, for instance by combining it with decay weighting strategies [19] to better balance contributions from lower and higher diffusion steps. Similarly, alternative EMA schemes, such as the parameter exchange proposed by Li et al. [12], may further stabilize training.

To improve physical plausibility and naturalness, integrating a physics simulator [33] could allow training without geometric losses while still enforcing physical constraints. Additionally, recent work by Tevet et al. [28] explores combining diffusion models with Reinforcement Learning, outsourcing physical plausibility to it while leveraging diffusion for generating diverse motions. Their approach is applied to the human motion interaction generation domain [27], which comprises the generation of motion with respect to environmental interactions such as human-human interaction, human-object interaction, and human-scene interaction. Investigating this domain could be particularly promising for generating synthetic data to enhance existing datasets, as interaction plays a critical role in recognition and detection tasks.

# Bibliography


[1] Busbridge, Dan ; Ramapuram, Jason ; Ablin, Pierre ; Likhomanenko, Tatiana ; Dhekane, Eeshan G. ; Suau, Xavier ; Webb, Russ: How to scale your EMA. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA : Curran Associates Inc., 2023 (NIPS '23)

[2] Cervantes, Pablo ; Sekikawa, Yusuke ; Sato, Ikuro ; Shinoda, Koichi: Implicit Neural Representations for Variable Length Human Motion Generation. In: *Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*. Berlin, Heidelberg : Springer-Verlag, 2022, S. 356–372. – URL https://doi.org/10.1007/978-3-031-19790-1_22. – ISBN 978-3-031-19789-5

[3] Chen, Xin ; Jiang, Biao ; Liu, Wen ; Huang, Zilong ; Fu, Bin ; Chen, Tao ; Yu, Gang: Executing your Commands via Motion Diffusion in Latent Space. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, S. 18000–18010

[4] Dhariwal, Prafulla ; Nichol, Alex: *Diffusion Models Beat GANs on Image Synthesis*. 2021. – URL https://arxiv.org/abs/2105.05233

[5] Goodfellow, Ian J. ; Pouget-Abadie, Jean ; Mirza, Mehdi ; Xu, Bing ; Warde-Farley, David ; Ozair, Sherjil ; Courville, Aaron ; Bengio, Yoshua: Generative Adversarial Nets. In: Ghahramani, Z. (Hrsg.) ; Welling, M. (Hrsg.) ; Cortes, C. (Hrsg.) ; Lawrence, N. (Hrsg.) ; Weinberger, K.Q. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 27, Curran Associates, Inc., 2014. – URL https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf

[6] Guo, Chuan ; Zuo, Xinxin ; Wang, Sen ; Zou, Shihao ; Sun, Qingyao ; Deng, Annan ; Gong, Minglun ; Cheng, Li: Action2Motion: Conditioned Generation

of 3D Human Motions. In: *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY, USA : Association for Computing Machinery, 2020 (MM '20), S. 2021–2029. – URL https://doi.org/10.1145/3394171.3413635. – ISBN 9781450379885

[7] Ho, Jonathan ; Jain, Ajay ; Abbeel, Pieter: Denoising Diffusion Probabilistic Models. In: Larochelle, H. (Hrsg.) ; Ranzato, M. (Hrsg.) ; Hadsell, R. (Hrsg.) ; Balcan, M.F. (Hrsg.) ; Lin, H. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 33, Curran Associates, Inc., 2020, S. 6840–6851. – URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

[8] Ji, Yanli ; Xu, Feixiang ; Yang, Yang ; Shen, Fumin ; Shen, Heng T. ; Zheng, Wei-Shi: A Large-scale RGB-D Database for Arbitrary-view Human Action Recognition. In: *Proceedings of the 26th ACM International Conference on Multimedia*. New York, NY, USA : Association for Computing Machinery, 2018 (MM '18), S. 1510–1518. – URL https://doi.org/10.1145/3240508.3240675. – ISBN 9781450356657

[9] Karras, Tero ; Aittala, Miika ; Lehtinen, Jaakko ; Hellsten, Janne ; Aila, Timo ; Laine, Samuli: Analyzing and Improving the Training Dynamics of Diffusion Models . In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA : IEEE Computer Society, Juni 2024, S. 24174–24184. – URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.02282

[10] Kingma, Diederik P. ; Welling, Max: *Auto-Encoding Variational Bayes*. 2022. – URL https://arxiv.org/abs/1312.6114

[11] Kocabas, Muhammed ; Athanasiou, Nikos ; Black, Michael J.: VIBE: Video Inference for Human Body Pose and Shape Estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020

[12] Li, Siyuan ; Liu, Zicheng ; Tian, Juanxi ; Wang, Ge ; Wang, Zedong ; Jin, Weiyang ; Wu, Di ; Tan, Cheng ; Lin, Tao ; Liu, Yang ; Sun, Baigui ; Li, Stan Z.: *Switch EMA: A Free Lunch for Better Flatness and Sharpness*. 2024. – URL https://arxiv.org/abs/2402.09240

[13] Li, Xuheng ; Gu, Quanquan: *Understanding SGD with Exponential Moving Average: A Case Study in Linear Regression*. 2025. – URL https://arxiv.org/abs/2502.14123

[14] Loper, Matthew ; Mahmood, Naureen ; Romero, Javier ; Pons-Moll, Gerard ; Black, Michael J.: SMPL: a skinned multi-person linear model. In: *ACM Trans. Graph.* 34 (2015), Oktober, Nr. 6. – URL https://doi.org/10.1145/2816795.2818013. – ISSN 0730-0301

[15] Lu, Qiujing ; Zhang, Yipeng ; Lu, Mingjian ; Roychowdhury, Vwani: Action-conditioned On-demand Motion Generation. In: *Proceedings of the 30th ACM International Conference on Multimedia*. New York, NY, USA : Association for Computing Machinery, 2022 (MM '22), S. 2249–2257. – URL https://doi.org/10.1145/3503161.3548287. – ISBN 9781450392037

[16] Nichol, Alexander Q. ; Dhariwal, Prafulla: Improved Denoising Diffusion Probabilistic Models. In: Meila, Marina (Hrsg.) ; Zhang, Tong (Hrsg.): *Proceedings of the 38th International Conference on Machine Learning* Bd. 139, PMLR, 18–24 Jul 2021, S. 8162–8171. – URL https://proceedings.mlr.press/v139/nichol21a.html

[17] Petrovich, Mathis ; Black, Michael J. ; Varol, Gül: Action-Conditioned 3D Human Motion Synthesis with Transformer VAE. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, S. 10965–10975

[18] Polyak, B. T. ; Juditsky, A. B.: Acceleration of stochastic approximation by averaging. In: *SIAM J. Control Optim.* 30 (1992), Juli, Nr. 4, S. 838–855. – URL https://doi.org/10.1137/0330046. – ISSN 0363-0129

[19] Qi, Qiaosong ; Zhuo, Le ; Zhang, Aixi ; Liao, Yue ; Fang, Fei ; Liu, Si ; Yan, Shuicheng: DiffDance: Cascaded Human Motion Diffusion Model for Dance Generation. In: *Proceedings of the 31st ACM International Conference on Multimedia*. New York, NY, USA : Association for Computing Machinery, 2023 (MM '23), S. 1374–1382. – URL https://doi.org/10.1145/3581783.3612307. – ISBN 9798400701085

[20] Raab, Sigal ; Leibovitch, Inbal ; Li, Peizhuo ; Aberman, Kfir ; Sorkine-Hornung, Olga ; Cohen-Or, Daniel: MoDi: Unconditional Motion Synthesis from Diverse Data. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, S. 13873–13883

[21] Radford, Alec ; Kim, Jong W. ; Hallacy, Chris ; Ramesh, Aditya ; Goh, Gabriel ; Agarwal, Sandhini ; Sastry, Girish ; Askell, Amanda ; Mishkin, Pamela ; Clark, Jack ; Krueger, Gretchen ; Sutskever, Ilya: Learning Transferable Visual Models From Natural Language Supervision. In: Meila, Marina (Hrsg.) ; Zhang, Tong (Hrsg.): *Proceedings of the 38th International Conference on Machine Learning* Bd. 139, PMLR, 18–24 Jul 2021, S. 8748–8763. – URL https://proceedings.mlr.press/v139/radford21a.html

[22] Rezende, Danilo ; Mohamed, Shakir: Variational Inference with Normalizing Flows. In: Bach, Francis (Hrsg.) ; Blei, David (Hrsg.): *Proceedings of the 32nd International Conference on Machine Learning* Bd. 37. Lille, France : PMLR, 07–09 Jul 2015, S. 1530–1538. – URL https://proceedings.mlr.press/v37/rezende15.html

[23] Shahroudy, Amir ; Liu, Jun ; Ng, Tian-Tsong ; Wang, Gang: NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, S. 1010–1019

[24] Sitzmann, Vincent ; Martel, Julien ; Bergman, Alexander ; Lindell, David ; Wetzstein, Gordon: Implicit Neural Representations with Periodic Activation Functions. In: Larochelle, H. (Hrsg.) ; Ranzato, M. (Hrsg.) ; Hadsell, R. (Hrsg.) ; Balcan, M.F. (Hrsg.) ; Lin, H. (Hrsg.): *Advances in Neural Information Processing Systems* Bd. 33, Curran Associates, Inc., 2020, S. 7462–7473. – URL https://proceedings.neurips.cc/paper_files/paper/2020/file/53c04118df112c13a8c34b38343b9c10-Paper.pdf

[25] Sohl-Dickstein, Jascha ; Weiss, Eric ; Maheswaranathan, Niru ; Ganguli, Surya: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In: Bach, Francis (Hrsg.) ; Blei, David (Hrsg.): *Proceedings of the 32nd International Conference on Machine Learning* Bd. 37. Lille, France : PMLR, 07–09 Jul 2015, S. 2256–2265. – URL https://proceedings.mlr.press/v37/sohl-dickstein15.html

[26] Song, Jiaming ; Meng, Chenlin ; Ermon, Stefano: Denoising Diffusion Implicit Models. In: *CoRR* abs/2010.02502 (2020). – URL https://arxiv.org/abs/2010.02502

[27] Sui, Kewei ; Ghosh, Anindita ; Hwang, Inwoo ; Zhou, Bing ; Wang, Jian ; Guo, Chuan: *A Survey on Human Interaction Motion Generation*. 2025. – URL https://arxiv.org/abs/2503.12763

[28] Tevet, Guy ; Raab, Sigal ; Cohan, Setareh ; Reda, Daniele ; Luo, Zhengyi ; Peng, Xue B. ; Bermano, Amit H. ; Panne, Michiel van de: *CLoSD: Closing the Loop between Simulation and Diffusion for multi-task character control*. 2024. – URL https://arxiv.org/abs/2410.03441

[29] Tevet, Guy ; Raab, Sigal ; Gordon, Brian ; Shafir, Yoni ; Cohen-or, Daniel ; Bermano, Amit H.: Human Motion Diffusion Model. In: *The Eleventh International Conference on Learning Representations*, URL https://openreview.net/forum?id=SJ1kSyO2jwu, 2023

[30] Varol, Gül ; Laptev, Ivan ; Schmid, Cordelia ; Zisserman, Andrew: Synthetic Humans for Action Recognition from Unseen Viewpoints. In: *International Journal of Computer Vision* 129 (2021), April, S. 2264–2287. – URL https://inria.hal.science/hal-02435731

[31] Vaswani, Ashish ; Shazeer, Noam ; Parmar, Niki ; Uszkoreit, Jakob ; Jones, Llion ; Gomez, Aidan N. ; Kaiser, Łukasz ; Polosukhin, Illia: Attention is all you need. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA : Curran Associates Inc., 2017 (NIPS'17), S. 6000–6010. – ISBN 9781510860964

[32] Weng, Lilian: *What are diffusion models?* Jul 2021. – URL https://lilianweng.github.io/posts/2021-07-11-diffusion-models/. – Zugriffsdatum: 2025-07-27

[33] Yuan, Ye ; Song, Jiaming ; Iqbal, Umar ; Vahdat, Arash ; Kautz, Jan: PhysDiff: Physics-Guided Human Motion Diffusion Model. In: *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, S. 15964–15975

[34] Zhang, Mingyuan ; Cai, Zhongang ; Pan, Liang ; Hong, Fangzhou ; Guo, Xinying ; Yang, Lei ; Liu, Ziwei: MotionDiffuse: Text-Driven Human Motion Generation With Diffusion Model. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46 (2024), Nr. 6, S. 4115–4128

[35] Zhou, Wenyang ; Dou, Zhiyang ; Cao, Zeyu ; Liao, Zhouyingcheng ; Wang, Jingbo ; Wang, Wenjia ; Liu, Yuan ; Komura, Taku ; Wang, Wenping ; Liu,

Lingjie: EMDM: Efficient Motion Diffusion Model for Fast and High-Quality Motion Generation. In: *Computer Vision - ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part II*. Berlin, Heidelberg : Springer-Verlag, 2024, S. 18–38. – URL https://doi.org/10.1007/978-3-031-72627-9_2. – ISBN 978-3-031-72626-2

[36] Zhou, Yi ; Barnes, Connelly ; Lu, Jingwan ; Yang, Jimei ; Li, Hao: On the Continuity of Rotation Representations in Neural Networks. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, S. 5738–5746

[37] Zhu, Wentao ; Ma, Xiaoxuan ; Ro, Dongwoo ; Ci, Hai ; Zhang, Jinlu ; Shi, Jiaxin ; Gao, Feng ; Tian, Qi ; Wang, Yizhou: Human Motion Generation: A Survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46 (2024), Nr. 4, S. 2430–2449

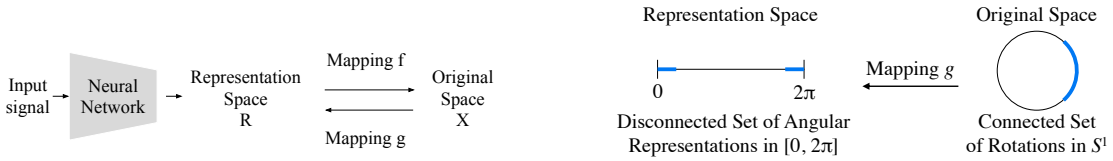
# A 6D Rotation Representation

Zhou et al. [36] have shown that pose representations with dimensionality lower than five are discontinuous, which leads to large errors when neural networks attempt to predict rotations. Figure A.1a illustrates the definition of continuous rotation representations. Let $X$ denote the original space (e.g., the set of all 3D rotations). Let $R$ denote a representation space, e.g. Euler angles. Then, a neural network generates intermediate representations in $R$. Passing a representation to a mapping function $f : R \rightarrow X$, recovers an element of the original space $X$. Based on the mapping function $g : X \rightarrow R$ it is possible to convert elements of the original space into their representation, and thus the pair $(f, g)$ is called a representation if for every $x \in X, f(g(x)) = x$. Such a representation is considered continuous if the mapping $g$ is continuous. However, if $g$ is discontinuous - as illustrated in Figure A.1b, the connected rotations in the original space might be mapped to disconnected elements in the representation space. This creates errors in neural networks, which rely on continuity. One solution to this problem is to rely on identity mappings, using $n \times n$ matrices. However, this approach is inefficient, and thus Zhou et al. propose an approach enforcing orthogonality directly within the representation. Regarding 3D rotations, they propose a mapping function $g$ to drop the last vector column of a $n \times n$ matrix, and convert rotations to the representation space. They further define a mapping function $f$, to reconstruct $X$, by generalizing the cross product to $n$ dimensions. This approach results in a continuous representation with $n^2 - n$ dimensions for $n$-dimensional rotations. In the 3-dimensional case, this yields a 6D representation obtained by flattening the first two columns of a rotation matrix

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},$$

into a vector

$$\mathbf{r} = (r_{11}, r_{21}, r_{31}, r_{12}, r_{22}, r_{32}) \in \mathbb{R}^6.$$




(a) Illustration of a continuous rotation representation (adapted from Figure 2 in [36]).

(b) Example of discontinuity in a 2D rotation representation (adapted from Figure 1 in [36]).

Figure A.1: Continuity of pose representations and their connection to neural networks.

The third column $\mathbf{r}_3$ is then reconstructed using the cross product $\mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2$, followed by an orthogonalization to ensure a valid rotation matrix. This continuous representation is especially useful, as the resulting $3 \times 3$ matrix generated through the mapping function $f$ is orthogonal. Hence, the orthogonal matrices generated by the neural network allow for further processing using methods like FK or IK. The continuous 6D representation has been widely adopted in recent works, such as [2, 17, 29, 34, 35].

# B Action-Conditioned Motion Methods

| Study | Year | Novelty/Contribution | Architecture | Datasets | Representation |
|---|---|---|---|---|---|
| **Latent- /Representation-Based Methods** | | | | | |
| Action2Motion [6] | 2020 | - first work on action-conditioned 3D human motion generation<br>- introduced NTU-RGB+D dataset for the task | Conditional Temporal VAE | HumanAct12,<br>NTU RGB+D,<br>MoCap | 3D joint<br>rotations |
| ACTOR [17] | 2021 | - action-aware latent representation for human motions<br>- variable sequence lengths via positional encoding | Transformer,<br>Conditional VAE | UESTC,<br>HumanAct12,<br>NTU RGB+D | 6D joint<br>rotations |
| ODMO [15] | 2022 | - generation of long-length motion<br>- on-demand customization (e.g., interpolation of modes)<br>- contrastive learning for hierarchical motion embedding | VAE-like latent model<br>one encoder +<br>two decoders (trajectory & motion) | UESTC,<br>HumanAct12,<br>MoCap | 3D joint<br>positions |
| INR [2] | 2022 | - variation INR Framework<br>- improved variable sequence-length generation<br>- novel metric, the Mean Maximum Similarity for GMMs | MLP-Decoder | UESTC,<br>HumanAct12,<br>NTU RGB+D | 6D joint<br>rotations |
| **Model-Agnostic Methods** | | | | | |
| PhysDiff [33] | 2023 | - a physics-guided motion diffusion model<br>- integrable into any motion diffusion model without retraining | Physics-based motion-<br>projection module,<br>RL-trained MLP policy,<br>Physics simulator | UESTC,<br>HumanAct12 | - |
| **Diffusion-Based Methods** | | | | | |
| MDM [29] | 2023 | - a lightweight and controllable diffusion-based model<br>- improved synthesis through geometric losses<br>- solves multiple tasks, such as action-to-motion and motion editing | Transformer (encoder-only),<br>Diffusion | UESTC,<br>HumanAct12 | 6D joint<br>rotations |
| MLD [3] | 2023 | - diffusion-based model working in latent space<br>- faster inference time due to latent approach | VAE,<br>Diffusion | UESTC,<br>HumanAct12 | 3D joint<br>rotations |
| EMDM [35] | 2024 | - improving inference time without decreasing sample quality<br>- use of a Denoising Diffusion GAN | Generative Adversarial Network,<br>Diffusion | HumanAct12 | 6D joint<br>rotations |
| MotionDiffuse [34] | 2024 | - manipulation of single body parts<br>- synthesis variable sequence-lengths | Cross-Modality Linear Transformer,<br>Diffusion | UESTC,<br>HumanAct12 | 6D joint<br>rotations |

Table B.1: Representative action-conditioned motion generation methods.

# C Overview of Hyperparameters

| Parameter | Value(s) | Notes |
|---|---|---|
| seed | 10 | Seed used across all trainings |
| batch_size | 8 | global batch size = 64 ($8 \times$ 8GPUs) |
| num_steps | 1e6, 2e6 | Distributed Data Parallel: local_steps = num_steps $\div$ world_size |
| save_interval | 25,000 | Save a checkpoint every this many steps |
| **Learning Rate** | | |
| lr | 1.2e−4 | Learning rate after warmup |
| warmup_steps | 5,000 | Linear warmup over this many steps |
| t_max | 1e6, 2e6 | Number of steps for cosine annealing aligned with num_steps |
| eta_min | 1e−5 | Minimum learning rate for cosine annealing |
| weight_decay | 5e−5 | Weight decay of the AdamW optimizer |
| **Optimization Techniques** | | |
| ema_decay | 0.9995, 0.9999 | Exponential moving average decay rate |
| schedule_sampler | uniform, loss_aware | Importance sampling strategy |
| **Geometric Losses** | | |
| lambda_rcxyz | 0.0 | Weight for the joint position loss |
| lambda_vel | 25.0 | Weight for the velocity loss |
| lambda_fc | 45.0 | Weight for the foot contact loss |

Table C.1: Training hyperparameters

| Parameter | Value(s) | Notes |
|---|---|---|
| n_heads | 8 | Number of attention heads |
| n_layers | 8 | Number of transformer layers |
| model_dim | 512 | Dimension of transformer model |
| ff_size | 1,024 | Dimension of feedforward network |
| dropout | 0.0 | Dropout rate in transformer |
| cond_mask_prob | 0.1 | Probability of masking condition (action) inputs |
| activation | gelu | Activation function in feedforward network |
| **Dataset-specific** | | |
| n_actions | 40 | Number of action classes |
| n_joints | 25 | Number of joints in the skeleton (SMPL) |
| n_feats | 6 | Number of features per joint (6D rotation) |
| data_repr | rot6d | Data representation: 6D joint rotations |

Table C.2: Model hyperparameters

| Parameter | Value(s) | Notes |
|---|---|---|
| num_diffusion_steps | 100, 1,000 | Number of diffusion steps during training |
| rescale_timesteps | False | Whether to rescale timesteps |
| loss_type | Mean Squared Error | Loss type for diffusion model (See Equation 2.11) |
| **Noise Schedule** | | |
| beta_schedule | cosine | Schedule for noise levels $\beta_t$ |
| scale_beta | 1.0 | Scaling factor for noise levels $\beta_t$ (1.0 means no scaling) |
| **Model Output** | | |
| predict_x_start | True | Predict $\mathbf{x}_0$ instead of noise $\epsilon$ |
| learn_sigma | False | Whether to learn variance |
| sigma_small | True | use smaller sigma values for improved sample quality |

Table C.3: Diffusion hyperparameters

## Erklärung zur selbständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

*Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI*

## Erklärung zur selbständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Bachelorarbeit — bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit — mit dem Thema:

**A Diffusion-Based Approach to 3D Human Motion Synthesis**

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe, einschließlich unterstützender Software- und Online-Tools zum Paraphrasieren und zur sprachlichen Verbesserung. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

| _____ | _____ | _____ |
|---|---|---|
| Ort | Datum | Unterschrift im Original |