

Guided Motion Diffusion for Controllable Human Motion Synthesis

Korraue Karunratanakul¹ Konpat Preechakul² Supasorn Suwajanakorn² Siyu Tang¹

¹ETH Zürich, Switzerland ²VISTEC, Thailand

<https://korraue.github.io/gmd-project/>

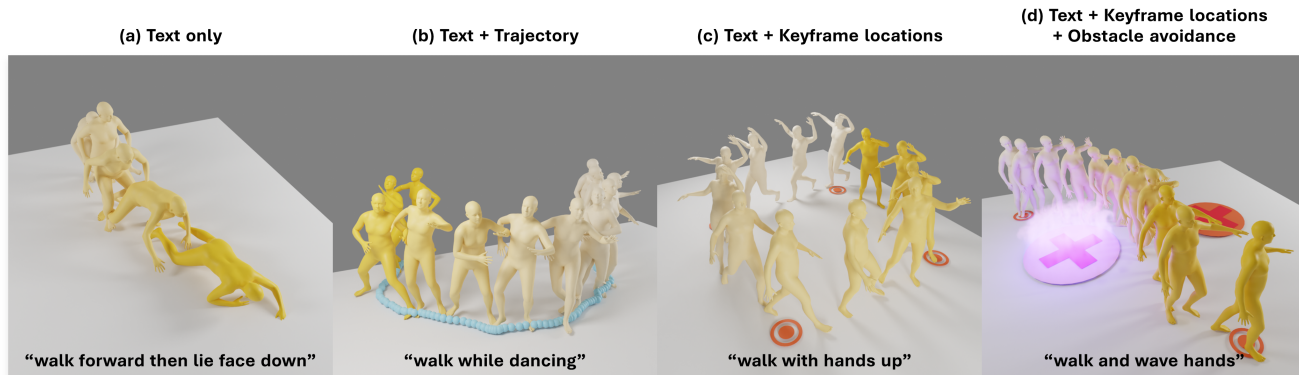


Figure 1. Our proposed Guided Motion Diffusion (GMD) can generate high-quality and diverse motions given a text prompt and a goal function. We demonstrate the controllability of GMD on four different tasks, guided by the following conditions: (a) text only, (b) text and trajectory, (c) text and keyframe locations (double circles), and (d) with obstacle avoidance (red-cross areas represent obstacles). The darker the colors, the later in time.

Abstract

Denoising diffusion models have shown great promise in human motion synthesis conditioned on natural language descriptions. However, integrating spatial constraints, such as pre-defined motion trajectories and obstacles, remains a challenge despite being essential for bridging the gap between isolated human motion and its surrounding environment. To address this issue, we propose Guided Motion Diffusion (GMD), a method that incorporates spatial constraints into the motion generation process. Specifically, we propose an effective feature projection scheme that manipulates motion representation to enhance the coherency between spatial information and local poses. Together with a new imputation formulation, the generated motion can reliably conform to spatial constraints such as global motion trajectories. Furthermore, given sparse spatial constraints (e.g. sparse keyframes), we introduce a new dense guidance approach to turn a sparse signal, which is susceptible to being ignored during the reverse steps, into denser signals to guide the generated motion to the given constraints. Our extensive experiments justify the development of GMD, which achieves a significant improvement over state-of-the-art methods in text-based motion generation while allowing control of the synthesized motions with spatial constraints.

1. Introduction

Recently, denoising diffusion models have emerged as a promising approach for human motion generation [12, 65, 67] outperforming other alternatives such as GAN or VAE in terms of both quality and diversity [8, 54, 61]. Several studies have focused on generating motion based on expressive text prompts [8, 54], or music [55, 67]. The state-of-the-art motion generation methods, such as MDM [54], utilize classifier-free guidance to generate motion conditioned on text prompts. However, incorporating spatial constraints into diffusion models remains underexplored. Human motions consist of both semantic and spatial information, where the semantic aspect can be described using natural languages or action labels and the spatial aspect governs physical interaction with surroundings. To generate realistic human motion in a 3D environment, both aspects must be incorporated. Our experiments show that simply adding spatial constraint guidance, such as global trajectories, into the state-of-the-art models or using imputation and in-painting approaches do not yield satisfactory results.

We identify two main issues that make the motion diffusion models likely to ignore the guidance when conditioned on spatial objectives: the sparseness of global orientation

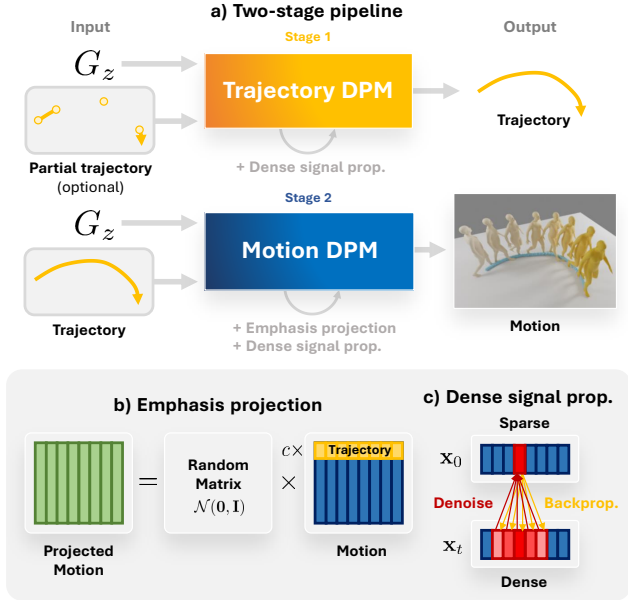


Figure 2. We tackle the problem of spatially conditioned motion generation with GMD, depicted in **a**). Our main contributions are **b**) **Emphasis projection**, for better trajectory-motion coherence, and **c**) **Dense signal propagation**, for a more controllable generation even under sparse guidance signal.

in the motion representation and sparse frame-wise guidance signals. By design, the diffusion models are a denoising model that consecutively denoises the target output over multiple steps. With sparse guidance, a small portion of the output that receives guidance will be inconsistent with all other parts that do not, therefore, are more likely to be treated as noise and discarded in subsequent steps.

First, the sparseness within a frame is a result of common motion representations that separate local pose information, like joint rotations, from global orientations, such as pelvis translations and rotations [46], usually with more focus on local poses. For instance, the common motion representation [15] uses 4 values to represent global orientation and 259 values for local pose in each frame. Such imbalance can cause the model to focus excessively on local pose information, and consequently, perceive guided global orientation as noise, resulting in a discrepancy such as foot skating.

Second, in many applications such as character animation, gaming, and virtual reality, the spatial control signals are defined on only a few keyframes such as target locations on the ground. We show that the current diffusion-based motion generation models struggle to follow such sparse guidance as doing so is equivalent to guiding an image diffusion model with only a few pixels. As a result, either the guidance at the provided keyframes will be ignored during the denoising process or the output motion will contain an artifact where the character warps to satisfy the guidance only in those specific keyframes.

To effectively incorporate sparse spatial constraints into the motion generation process, we propose GMD, a novel and principled Guided Motion Diffusion model. To alleviate the discrepancy between local pose and global orientation in the guided denoising steps, we introduce *emphasis projection*, a general representation manipulation method that we use to increase the importance of spatial information during training. Additionally, we derive a new imputation and inpainting formulation that enables the existing inpainting techniques to operate in the projected space, which we leverage to generate significantly more coherent motion under guidance by spatial conditions. Then, to address the highly sparse guidance, we draw inspiration from the credit assignment problem in Reinforcement Learning [53, 57], where sparse rewards can be distributed along a trajectory to allow for efficient learning [3]. Our key insight is that motion denoisers, including the diffusion model itself, can be used to expand the spatial guidance signal at a specific location to its neighboring locations without any additional model. By turning a sparse signal into a dense one by back-propagating through a denoiser, it enables us to achieve high-quality controllable motion synthesis, even with extremely sparse guidance signals.

In summary, our contributions are: (1) Emphasis projection, a method to adjust relative importance between different parts of the representation vector, which we use to encourage coherency between spatial information and local poses to allow spatial guidance. (2) Dense signal propagation, a conditioning method to tackle the sparse guidance problem. (3) GMD, an effective spatially controllable motion generation method that enables the unexplored synthesizing of motions based on free-text and spatial conditioning by integrating the above contributions into our proposed Unet-based architecture. We provide extensive analysis to support our design decisions and show the versatility of GMD on three tasks: trajectory conditioning, keyframe conditioning, and obstacle avoidance. Additionally, GMD’s model also significantly outperforms the state-of-the-art in traditional text-to-motion tasks.

2. Related Work

Diffusion-based probabilistic generative models (DPM). DPMs [21, 50, 51, 52] have gained significant attention in recent years due to their impressive performance across multiple fields of research. They have been used for tasks such as image generation [13], image super-resolution [31, 49], speech synthesis [27, 27, 42], video generation [20, 23], 3D shape generation [41, 58], and reinforcement learning [24].

The surge in interest in DPMs may be attributed to their impressive controllable generation capabilities, including text-conditioned generation [45, 47, 48] and image editing [4, 6, 10, 19, 35]. Latent diffusion models (LDM) are an-

other area of interest, which includes representation learning [28, 43] and more efficient modeling techniques [8, 47].

Moreover, DPMs exhibit a high degree of versatility in terms of conditioning. There are various methods for conditional generation, such as imputation and inpainting [10, 11, 35], classifier guidance [11, 13], and classifier-free guidance [22, 45, 47, 48]. Inpainting and classifier guidance can be applied to any pretrained DPM, which extends the model’s capabilities further without the need for retraining.

Human motion generation. The goal of the human motion generation task is to generate motions based on the conditioning signals. Various conditions have been explored such as partial poses [14, 18, 54], trajectories [25, 56, 63], images [9, 46], music [29, 30, 32], text [1, 15, 16, 26, 40], objects [59], action labels [17, 39], or unconditioned [37, 60, 64, 66]. Recently, many diffusion-based motion generation models have been proposed [12, 26, 33, 65, 67] and demonstrate better quality compared to alternative models such as GAN or VAE. Employing the CLIP model [44], these models showed great improvements in the challenging text-to-motion generation task [8, 61, 62] as well as allowing conditioning on partial motions [54] or music [2, 55]. However, they do not support conditioning signals that are not specifically trained, for example, following keyframe locations or avoiding obstacles. Maintaining the capabilities of the diffusion models, we propose methods to enable spatial guidance without retraining the model for each new objective.

3. Background

3.1. Diffusion-based generative models

Diffusion-based probabilistic generative models (DPMs) are a family of generative models that learn a sequential denoising process of an input \mathbf{x}_t with varying noise levels t . The noising process of DPM is defined cumulatively as $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_0, (1 - \alpha_t)\mathbf{I})$, where \mathbf{x}_0 is the clean input, $\alpha_t = \prod_{s=1}^t(1 - \beta_s)$, and β_t is a noise scheduler. The denoising model $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ with parameters θ learns to reverse the noising process by modeling the Gaussian posterior distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. DPMs can map a prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to any distribution $p(\mathbf{x})$ after T successive denoising steps.

To draw samples from a DPM, we start from a sample \mathbf{x}_T from the prior distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, for each t , we sample $\mathbf{x}_{t-1} \sim \mathcal{N}(\mu_t, \Sigma_t)$ until $t = 0$, where

$$\mu_t = \frac{\sqrt{\alpha_{t-1}}\beta_t}{1 - \alpha_t}\mathbf{x}_0 + \frac{\sqrt{1 - \beta_t}(1 - \alpha_{t-1})}{1 - \alpha_t}\mathbf{x}_t \quad (1)$$

and Σ_t is a variance scheduler of choice, usually $\Sigma_t = \frac{1 - \alpha_{t-1}}{1 - \alpha_t}\beta_t$ [21]. \mathbf{x}_0 in Eq. 1 is the prediction from a denoising model. For an ϵ_θ model, $\mathbf{x}_0 = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t + \frac{\sqrt{1 - \alpha_t}}{\sqrt{\alpha_t}}\epsilon_\theta(\mathbf{x}_t)$.

There are multiple choices for the denoising model to predict including the clean input \mathbf{x}_0 , the noise ϵ , and the one-step denoised target μ_t . An $\mathbf{x}_{0,\theta}$ model is trained using the squared loss to the clean input $\|\mathbf{x}_{0,\theta}(\mathbf{x}_t) - \mathbf{x}_0\|^2$, an ϵ_θ model is trained using the squared loss $\|\epsilon_\theta(\mathbf{x}_t) - \epsilon\|^2$, and $\mu_{t,\theta}$ model is trained using the squared loss $\|\mu_{t,\theta}(\mathbf{x}_t) - \mu_t\|^2$.

3.2. Controllable generation with diffusion models.

Classifier-free guidance. The conditioning signals are treated as additional inputs to the denoiser $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, d)$ where d is the conditioning signals which can be omitted $d = \emptyset$ to generate unconditionally. Classifier-free guidance has been shown to generate very high-quality results [47, 48, 54]. To draw samples, the effective denoiser becomes $\hat{\epsilon}_\theta(\mathbf{x}_t, d) = w\epsilon_\theta(\mathbf{x}_t, d) + (1 - w)\epsilon_\theta(\mathbf{x}_t, \emptyset)$, where w controls the conditional strength. The new $\hat{\epsilon}_\theta$ model can be used in Eq. 1. Two downsides of this method are that the nature of conditioning signals need to be known before hand and the denoiser needs to be adjusted and retrained for each specific case restricting its flexibility.

Classifier guidance. We can also obtain $p(\mathbf{x}_{t-1}|\mathbf{x}_t, d)$ from $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)p(d|\mathbf{x}_t)$ [13], where $p(d|\mathbf{x}_t)$ is any probability function that we can approximate its score function $\nabla_{\mathbf{x}_t} \log p(d|\mathbf{x}_t)$ effectively. The new sampling process is similar to the original (Eq. 1) but with the mean shifted by the scaled score function as

$$\mu_t = \mu'_t + s\Sigma_t\nabla_{\mathbf{x}_t} \log p(d|\mathbf{x}_t) \quad (2)$$

where μ'_t is the original mean, s controls the conditioning strength, and Σ_t is a variance scheduler which can be the same as in Eq. 1. Since Σ_t is a decreasing sequence, the guidance signal diminishes as $t \rightarrow 0$ which corresponds to the characteristic of DPMs that tend to modify \mathbf{x}_t less and less as time goes. Classifier guidance is a post-hoc method, i.e., there is no change to the DPM model, one only needs to come up with $p(d|\mathbf{x}_t)$ which is extremely flexible.

Imputation and inpainting. To generate human motion sequences from partial observations, such as global motion trajectories or keyframe locations, inpainting is used. These partial observations, called imputing signals, are used to adjust the generative process towards the observations. Imputation and inpainting are two sides of the same coin.

Let \mathbf{y} be a partial target value in an input \mathbf{x} that we want to impute. The imputation region of \mathbf{y} on \mathbf{x} is denoted by M_y^x , and a projection P_y^x that resizes \mathbf{y} to that of \mathbf{x} by filling in zeros. In DPMs, imputation can be done on the sample \mathbf{x}_{t-1} after every denoising step [11]. We have the new imputed sample $\tilde{\mathbf{x}}_{t-1}$ as

$$\tilde{\mathbf{x}}_{t-1} = (1 - M_y^x) \odot \mathbf{x}_{t-1} + M_y^x \odot P_y^x \mathbf{y}_{t-1} \quad (3)$$

where \odot is a Hadamard product and \mathbf{y}_{t-1} is a noised target value. $\mathbf{y}_{t-1} \sim \mathcal{N}(\sqrt{\alpha_{t-1}}\mathbf{y}, (1 - \alpha_{t-1})\mathbf{I})$ following Ho *et al.* [21] is one of the simplest choices of \mathbf{y}_{t-1} .

Note that all three modes of conditioning presented here are not mutually exclusive. One could apply one or more tricks in a single pipeline.

4. Guided Motion Diffusion

Algorithm 1 GMD’s two-stage guided motion diffusion

Require: A trajectory DPM $\mathbf{z}_{0,\phi}$, a motion DPM $\mathbf{x}_{0,\theta}$, a goal function $G_z(\cdot)$, and keyframe locations \mathbf{y} (if any).

```

1: # Stage 1: Trajectory generation
2:  $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
3: for all  $t$  from  $T$  to 1 do
4:    $\mathbf{z}_0 \leftarrow \mathbf{z}_{0,\phi}(\mathbf{z}_t)$ 
5:    $\mu, \Sigma \leftarrow \mu(\mathbf{z}_0, \mathbf{z}_t), \Sigma_t$ 
6:   # Classifier guidance (Eq. 2)
7:   # Dense signal propagation
8:    $\mathbf{z}_{t-1} \sim \mathcal{N}(\mu - s\Sigma\nabla_{\mathbf{z}_t}G_z(\mathbf{z}_0), \Sigma)$ 
9:   # Impute  $\mathbf{y}$  on  $\mathbf{z}$  (Eq. 3) (if any)
10:   $\mathbf{z}_{t-1} \leftarrow (1 - M_y^z) \odot \mathbf{z}_{t-1} + M_y^z \mathbf{y}_{t-1}$ 
11: end for
12: # Stage 2: Trajectory-conditioned motion generation
13:  $\mathbf{x}_T^{\text{proj}} \leftarrow \text{sample from } \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
14: for all  $t$  from  $T$  to 1 do
15:    $M \leftarrow P_z^x M_y^z$  # Imputation region of  $\mathbf{y}$  on  $\mathbf{x}$ 
16:    $\mathbf{x}_0^{\text{proj}} \leftarrow \mathbf{x}_{0,\theta}^{\text{proj}}(\mathbf{x}_t^{\text{proj}})$  # Emphasis projection
17:   # Impute  $\mathbf{y}$  on  $\mathbf{x}^{\text{proj}}$  (Eq. 6)
18:    $\tilde{\mathbf{x}}_0^{\text{proj}} \leftarrow A \left( (1 - M) \odot A^{-1} \mathbf{x}_0^{\text{proj}} + M P_z^x \mathbf{y} \right)$ 
19:    $\mu, \Sigma \leftarrow \mu(\tilde{\mathbf{x}}_0^{\text{proj}}, \mathbf{x}_t^{\text{proj}}), \Sigma_t$ 
20:   # Masked classifier guidance (Eq. 9)
21:   # Dense signal propagation
22:    $\Delta \leftarrow -s\Sigma A^{-1} \nabla_{\mathbf{x}_t^{\text{proj}}} G_z(P_x^z A^{-1} \mathbf{x}_0^{\text{proj}})$ 
23:    $\mu \leftarrow \mu + A(1 - M) \odot \Delta$ 
24:    $\mathbf{x}_{t-1} \sim \mathcal{N}(\mu, \Sigma)$ 
25: end for
26: return  $\mathbf{z}_0$ 

```

We aim to generate realistic human motions that can be guided by spatial constraints, enabling the generated human motion to achieve specific goals, such as following a global trajectory, reaching certain locations, or avoiding obstacles. Although diffusion-based models have significantly improved text-to-motion modeling [8, 54], generating motions that achieve specific goals is still beyond the reach of the current models. Our work addresses this limitation and advances the state-of-the-art in human motion modeling.

We are interested in modeling a full-body human motion that satisfies a certain scalar goal function $G_x(\cdot)$ that takes a motion representation \mathbf{x} and measures how far the

motion \mathbf{x} is from the goal (lower is better). More specifically, $\mathbf{x} \in \mathbb{R}^{N \times M}$ represents a sequence of human poses for M motion steps, where N is the dimension of human pose representations, e.g., $N = 263$ in the HumanML3D [15] dataset. Let X be the random variable associated with \mathbf{x} . Our goal is to model the following conditional probability using a motion DPM

$$p(\mathbf{x}|G_x(X) = 0) \quad (4)$$

This can be extended to $p(\mathbf{x}|G_x(X) = 0, d)$, where d is any additional signal, such as text prompts. From now on, we omit d to reduce clutter.

Many challenging tasks in motion modeling can be encapsulated within a goal function G_z that only depends on the trajectory \mathbf{z} of the human motion, not the whole motion \mathbf{x} . Let us define $\mathbf{z} \in \mathbb{R}^{L \times M}$ to be the trajectory part of \mathbf{x} with length M and $L = 2$ describing the ground location of the pelvis of a human body. A particular location $\mathbf{z}^{(i)}$ at motion step i describes the pelvis location of the human body on the ground plane. We define a projection P_x^z that resizes \mathbf{x} to match \mathbf{z} by taking only the \mathbf{z} part, and its reverse P_z^x that resizes \mathbf{z} to match \mathbf{x} by filling in zeros. With this, our conditional probability becomes $p(\mathbf{x}|G_z(P_x^z X) = 0)$.

In this work, we will show how text-to-motion DPMs can be extended to solve several challenging tasks, including trajectory-conditioned motion generation, location-conditioned trajectory planning, and obstacle avoidance trajectory planning. Using our proposed Emphasis projection and dense signal propagation, we alleviate the sparse guidance problem and enable motion generation based on spatial conditions. The overview of our methods is shown in Fig. 3.

4.1. Emphasis projection

One of the most straightforward approaches for minimizing the goal function $G_z(\cdot)$ is by analyzing what trajectories that minimize $\mathbf{z}^* = \arg \min_{\mathbf{z}} G_z(\mathbf{z})$ look like. For a trajectory conditioning task, a whole trajectory \mathbf{z}^* is directly given. Our task is to generate the rest of the motion \mathbf{x} . With such knowledge, we can employ **imputation & inpainting** technique by supplying the motion DPM with the \mathbf{x} -shaped $P_z^x \mathbf{z}^*$ to guide the generation process.

Problem 1: Motion incoherence

Since the imputing trajectory \mathbf{z}^* is only a small part of the whole motion \mathbf{x} ($L \ll N$), we often observe that the DPM ignores the change from imputation and fails to make appropriate changes on the rest of \mathbf{x} . This results in an incoherent local motion that is not aligned or well coordinated with the imputing trajectory.

Solution 1: Emphasis projection

We tackle this problem by giving more emphasis on the trajectory part of motion \mathbf{x} . More specifically, we propose an **Emphasis projection** method that increases the trajectory’s

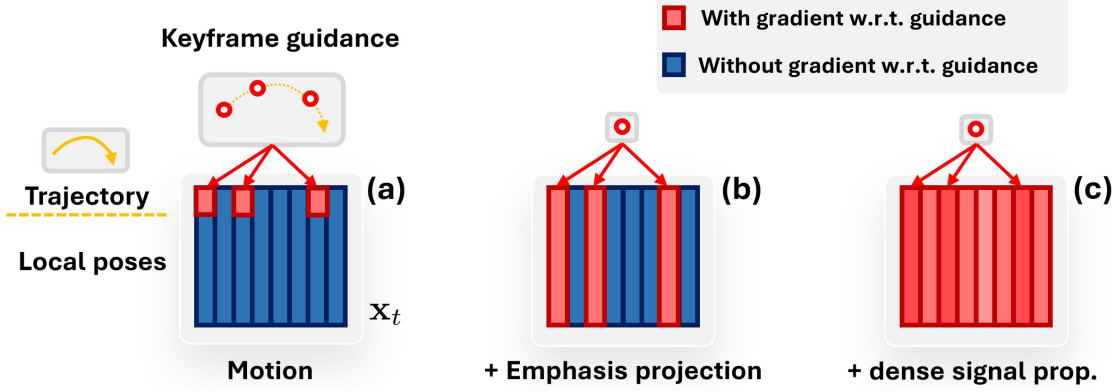


Figure 3. (a) Under standard motion representation and guiding method, only a few values in the motion representation are updated according to the guidance. (b) With Emphasis projection, all values in each frame describing the motion receives gradients w.r.t. the guidance, leading to better coherence between global orientation and local pose in each frame. (c) With dense gradient propagation, all frames are updated according to the guidance at the keyframes, making the guidance less likely to be ignored.

relative importance within motion \mathbf{x} . We achieve this by utilizing a random matrix $A = A'B$, where $A' \in \mathbb{R}^{N \times N}$ is a matrix with elements randomly sampled from $\mathcal{N}(0, 1)$ and $B \in \mathbb{R}^{N \times N}$ is a diagonal matrix whose trajectory-related diagonal indexes are c and the rest are 1 for emphasizing those trajectory elements. In our case, we emphasize the rotation and ground location of the pelvis, (rot, x, z) , in \mathbf{x} by c times. We now have a projected motion $\mathbf{x}^{\text{proj}} = \frac{1}{N-3+3c^2} A\mathbf{x}$. Note that the fractional term is to maintain the unit variance on \mathbf{x}^{proj} . The noising process of the projected motion becomes $q(\mathbf{x}_t^{\text{proj}} | \mathbf{x}_0^{\text{proj}}) = \mathcal{N}(\sqrt{\alpha_t} \mathbf{x}_0^{\text{proj}}, (1 - \alpha_t) \mathbf{I})$. There is no change on how a DPM that works on the projected motion $p_\theta(\mathbf{x}_{t-1}^{\text{proj}} | \mathbf{x}_t^{\text{proj}})$ operates and treats $\mathbf{x}_t^{\text{proj}}$.

In Section 6.3, we show that emphasis projection is an effective way of solving the motion incoherence problem, and is shown to be substantially better than a straightforward approach of retraining a DPM with an increased loss weight on the trajectory.

Imputation on the projected motion \mathbf{x}^{proj} . We have discussed imputing on the sample \mathbf{x}_{t-1} in Eq. 3. Here, we introduce an imputation on \mathbf{x}_0 which modifies the DPM’s belief on the final outcome $\mathbf{x}_{0,\theta}$ by imputing it with \mathbf{z} . We have found this technique useful in many tasks we are interested in.

Let us define the imputation region of \mathbf{z} on \mathbf{x} as M_z^x . We obtain the imputed $\tilde{\mathbf{x}}_0$ from

$$\tilde{\mathbf{x}}_0 = (1 - M_z^x) \odot \mathbf{x}_{0,\theta} + M_z^x \odot \underbrace{P_z^x \mathbf{z}^*}_{\mathbf{x} \text{ shaped}} \quad (5)$$

Now operating on the projected motion \mathbf{x}^{proj} , before we can do imputation, we need to unproject it back to the original motion using $\mathbf{x}_0 = A^{-1} \mathbf{x}_0^{\text{proj}}$, and then project the imputed $\tilde{\mathbf{x}}_0$ back using $\tilde{\mathbf{x}}_0^{\text{proj}} = A \tilde{\mathbf{x}}_0$. We obtain the imputed

motion under emphasis projection $\tilde{\mathbf{x}}_0^{\text{proj}}$ from

$$\tilde{\mathbf{x}}_0^{\text{proj}} = A \left((1 - M_z^x) \odot (A^{-1} \mathbf{x}_{0,\theta}^{\text{proj}}) + M_z^x \odot P_z^x \mathbf{z}^* \right) \quad (6)$$

Substituting $\tilde{\mathbf{x}}_0^{\text{proj}}$ into Eq. 1, we obtain the new mean $\tilde{\mu}_t^{\text{proj}}$ for sampling $\mathbf{x}_{t-1}^{\text{proj}} \sim \mathcal{N}(\tilde{\mu}_t^{\text{proj}}, \Sigma_t)$.

4.2. Dense guidance signal with a learned denoiser

Another way to minimize the goal function $G_z(\cdot)$ is by adjusting the sample of each diffusion step \mathbf{x}_{t-1} toward a region with lower G_z . This trick is called classifier guidance [13]. The direction of change corresponds to a score function $\nabla_{\mathbf{x}_t} \log p(G_x(X_t) = 0 | \mathbf{x}_t)$ which can be approximated as a direction $\Delta_{\mathbf{x}_0} = -\nabla_{\mathbf{x}_0} G_z(P_z^x \mathbf{x}_{0,\theta})$ that reduces the goal function. We can guide the generative process by nudging the DPM’s prediction as $\mathbf{x}_0 = \mathbf{x}_{0,\theta} + \Delta_{\mathbf{x}_0}$. While imputation requires the minimizer \mathbf{z}^* of G_z , which might not be easy to obtain or may not be unique, this trick only requires the easier-to-obtain direction of change.

Problem 2: Sparse guidance signal

In the motion domain, conditioning signals can often be sparse. There are two types of sparsity that can occur: sparsity in feature and sparsity in time. **Sparsity in feature** is when the conditioning signal is a small part of the feature dimension of \mathbf{x} . For example, in trajectory-conditioned generation, \mathbf{z} may only consist of a sequence of ground locations over time. This type of sparsity can be addressed by emphasis projection, as explained in Section 4.1. **Sparsity in time** refers to cases where the conditioning signal consists of small segments of a trajectory spread out over time. For instance, in keyframe location conditioning task, only a sparse set of keyframe locations are given. When the conditioning signal-to-noise ratio becomes too small, the conditioning signal may be mistaken as noise and ignored during the denoising process.

Solution 2: Dense signal propagation

To turn a sparse signal into a dense signal, we need domain knowledge. One way to achieve this is by using a denoising function $f(\mathbf{x}_t) = \mathbf{x}_0$, which is trained on a motion dataset to denoise by gathering information from the nearby motion frames. With the ability to relate a single frame to many other frames, the denoising function is capable of expanding a sparse signal into a denser one.

We can use backward propagation through the denoising function f to take advantage of this. Therefore, a dense classifier guidance can be obtained as follows:

$$\nabla_{\mathbf{x}_t} \log p(G_x(X_t) = 0 | \mathbf{x}_t) \approx -\nabla_{\mathbf{x}_t} G_z \left(\underbrace{P_x^z f(\mathbf{x}_t)}_{\mathbf{z} \text{ shaped}} \right) \quad (7)$$

While an external function can be used as f , we observe that the existing DPM model $\mathbf{x}_{0,\theta}(\mathbf{x}_t)$ itself is a motion denoiser, and thus can be used to turn a sparse signal into a dense signal without the need for an additional model. In practice, this process amounts to computing the gradient of G with respect to \mathbf{x}_t through $\mathbf{x}_{0,\theta}(\mathbf{x}_t)$ using autodiff.

Applying classifier guidance together with imputation.

Whenever available, we want to utilize signals from both imputation and classifier guidance techniques to help guide the generative process. Imputation is explicit but may encounter sparsity in time, while classifier guidance is indirect but dense. We want to use the direct signal from imputation wherever available (with mask M_z^x), and the rest from classifier guidance (with mask $1 - M_z^x$). Based on Eq. 2, imputation-aware classifier guidance can be written as

$$\mu_t = \tilde{\mu}_t - (1 - M_z^x) \odot s \Sigma_t \nabla_{\mathbf{x}_t} G_z(P_x^z f(\mathbf{x}_t)) \quad (8)$$

where $\tilde{\mu}$ is an imputed sampling mean. By replacing $\tilde{\mu}$ with $\tilde{\mu}^{\text{proj}}$, we get classifier guidance together with imputation that works with emphasis projection as

$$\Delta_\mu = -s \Sigma_t A^{-1} \nabla_{\mathbf{x}_t^{\text{proj}}} G_z(P_x^z A^{-1} f(\mathbf{x}_t^{\text{proj}})) \quad (9)$$

$$\mu_t^{\text{proj}} = \tilde{\mu}_t^{\text{proj}} + A(1 - M_z^x) \odot \Delta_\mu \quad (10)$$

Problem 3: DPM’s bias hinders the guidance signal

A DPM removes noise from an input based on the distribution of the training data it has seen. This could be problematic when it comes to conditional generation because the conditioning signal may be outside of the training distribution. As a result, any changes made to the classifier guidance may be reversed by the DPM in the next time step, due to its inherent bias towards the data, shown in Figure 4.

Solution 3: Epsilon modeling

While it is unlikely to train an unbiased DPM model, there are ways to minimize the influence of model’s bias under the guidance signal. Conceptually, the DPM model usually makes less and less change near the final outcome. This is

in tandem with the guidance signal that gradually decreases over time due to Σ_t (Eq. 2).

We investigate the coefficient $\frac{\sqrt{\alpha_t - 1} \beta_t}{1 - \alpha_t}$ of \mathbf{x}_0 in the sampling mean μ_t (Eq. 1). This coefficient reaches its maximum value at $t = 0$, meaning that an $\mathbf{x}_{0,\theta}$ model could have a significant impact on the sampling mean even at $t = 0$, which contradicts the weak guidance signal at that time.

On the other hand, an ϵ_θ model will have the most influence on the sampling mean at $t = T$, which aligns with our intuition. In Section 6.4 and Figure 4, we demonstrate that modeling ϵ_θ instead of $\mathbf{x}_{0,\theta}$ is a successful approach for managing the bias effect of the DPM model in classifier guidance. We further discuss this point in Supplementary.

5. Applications

5.1. Trajectory-conditioned generation

This task aims at generating a realistic motion \mathbf{x} that matches a given trajectory \mathbf{z} . Our objective is to minimize the distance between the generated motion and the given trajectory, which we define as

$$G_x(\mathbf{x}) := \left\| \mathbf{z} - \underbrace{P_x^z \mathbf{x}}_{\mathbf{z} \text{ part of } \mathbf{x}} \right\|_p \quad (11)$$

Despite the apparent simplicity of this task, a traditional DPM faces the challenge of ensuring coherence in the generated motion. However, our emphasis projection method can effectively address this problem.

5.2. Keyframe-conditioned generation

The locations of ground positions at specific times can be used to define locations that we wish the generated motion to reach. This task is a generalized version of the trajectory-conditioned generation where only a partial and potentially sparse trajectory is given. Let $\mathbf{y} \in \mathbb{R}^{2 \times M}$ be a trajectory describing keyframe locations and a mask M_y^z describe the key motion steps. Our goal function of a motion \mathbf{x} is

$$G_x(\mathbf{x}) := \sum_i \left\| M_y^z (P_x^z \mathbf{x} - \mathbf{y}) \right\|_p \quad (12)$$

Consequently, $G_z(\mathbf{z}) = \sum_i \left\| M_y^z (\mathbf{z} - \mathbf{y}) \right\|_p$. Due to the partial trajectory \mathbf{y} , the imputation region of \mathbf{y} on \mathbf{x} becomes $M_y^x = P_z^x M_y^z$.

Two-stage guided motion generation. Generating both the trajectory and motion simultaneously under a conditioning signal can be challenging and may result in lower quality motion. To address this issue, we propose a two-step approach. First, we generate a trajectory \mathbf{z} that satisfies the keyframe locations and then generate the motion \mathbf{x} given the trajectory (following Section 5.1). Our overall pipeline

is depicted in Figure 2 (a). We offer two options for generating the trajectory from keyframe locations \mathbf{y} : a point-to-point trajectory and a trajectory DPM.

The **point-to-point trajectory** connects consecutive keyframe locations with a straight line. These unrealistic trajectories can be used as imputation signals for the motion DPM during the early phase ($t \geq \tau$). If τ is large enough, the DPM will adjust the given trajectory to a higher quality one. However, if τ is too large, the DPM may generate a motion that does not perform well on G_z .

The **trajectory DPM** $p_\phi(\mathbf{z}_{t-1}|\mathbf{z}_t)$, which is trained using the same dataset but with a smaller network, can be used to generate the trajectory under the guidance signal from G_z . We summarize our two-stage approach in Algorithm 1.

It is also possible to combine the two methods, as the point-to-point trajectory can serve as a useful guidance signal for the trajectory DPM during $t \geq \tau$. After that, the trajectory DPM is subject to the usual imputation and classifier guidance from G_z . By tuning τ , we can balance between trajectory diversity and lower scores on G_z .

5.3. Obstacle avoidance motion generation

Humans have the ability to navigate around obstacles while traveling from point A to B. Under our framework, this problem can be defined using two goal functions: one that navigates from A to B, called G_x^{loc} (defined as in Eq. 12), and another that pushes back when the human model crosses the obstacle’s boundary, called G_x^{obs} , as follows

$$G_x^{\text{obs}}(\mathbf{x}) := \sum_i -\text{clipmax}(\text{SDF}((P_x^z \mathbf{x})^{(i)}), c) \quad (13)$$

where c is the safe distance from the obstacle. These two goal functions are combined additively to obtain the final goal function, $G_x(\mathbf{x}) = G_x^{\text{loc}}(\mathbf{x}) + G_x^{\text{obs}}(\mathbf{x})$, for this task.

We utilize the same pipeline as in Section 5.2, with the exception that imputation is not possible for obstacle avoidance. Therefore, minimizing the obstacle avoidance goal relies solely on classifier guidance.

6. Experiments

To evaluate our methods, we perform experiments on the standard human motion generation task conditioned on text descriptors and spatial objectives. In particular, we evaluate (1) the performance of our model in standard text-condition motion generation tasks, (2) the effect of emphasis projection to alleviate incoherence between spatial locations and local poses, (3) the ability to conditionally generate motion based on spatial information by conditioning with given trajectories, keyframe locations, and obstacles.

6.1. Settings

Evaluation metrics. We evaluate generative text-to-motion models using standard metrics introduced by Guo et al.

Table 1. Text-to-motion evaluation on the HumanML3D [15] dataset. The right arrow \rightarrow means closer to real data is better.

	FID ↓	R-precision ↑ (Top-3)	Diversity →
Real	0.002	0.797	9.503
JL2P [1]	11.02	0.486	7.676
Text2Gesture [5]	7.664	0.345	6.409
T2M [15]	1.067	0.740	9.188
MotionDiffuse [62]	0.630	0.782	9.410
MDM [54]	0.556	0.608	9.446
MLD [8]	0.473	0.772	9.724
PhysDiff [61]	0.433	0.631	-
Ours	0.212	0.670	9.440
Ours \mathbf{x}^{proj}	0.235	0.652	9.726

[15]. These include Fréchet Inception Distance (FID), R-Precision, and Diversity. **FID** measures the distance between the distributions of ground truth and generated motion using a pretrained motion encoder. **R-Precision** evaluates the relevance of the generated motion and its text prompt, while **Diversity** measures the variability within the generated motion. We also report **Foot skating ratio**, which measures the proportion of frames in which either foot skids more than a certain distance (2.5 cm) while maintaining contact with the ground (foot height < 5 cm), as a proxy for the incoherence between trajectory and human motion.

In addition, for conditional generation with keyframe locations, we use Trajectory diversity, Trajectory error, Location error, and Average error of keyframe locations. **Trajectory diversity** measures the root mean square distance of each location of each motion step from the average location of that motion step across multiple samples with the same settings. **Trajectory error** is the ratio of unsuccessful trajectories, defined as those with *any* keyframe location error exceeding a threshold. **Location error** is the ratio of keyframe locations that are not reached within a threshold distance. **Average error** measures the mean distance between the generated motion locations and the keyframe locations measured at the keyframe motion steps.

Datasets. We evaluate the text-to-motion generation using the HumanML3D [15] dataset, which is a collection of text-annotate motion sequences from AMASS [34] and Human-Act12 [17] datasets. It contains 14,646 motions and 44,970 motion annotations.

Implementation details. Both our motion DPM and trajectory DPM are based on UNET with AdaGN [13] depicted in details in the Supplementary. The motion DPM is an \mathbf{x}_0 model, while the trajectory DPM is an ϵ model, as explained in Section 4.2, to enhance controllability. We utilized DDPM [21] with $T=1,000$ denoising steps for training and inference of both models. Additionally, we condition the generation process on text prompts in a classifier-free [22] manner, similar to MDM [54], and use the CLIP [44]

Table 2. Trajectory-conditioned motions evaluation. The ground truth trajectory is used for imputing after each diffusion step. Comparing the effect of an original \mathbf{x} with emphasis loss functions to the emphasis projection \mathbf{x}^{proj} after imputing whole trajectories after each diffusion step.

Model	Space	Emphasis	FID ↓	Foot skating ↓ ratio
MDM [54]	\mathbf{x}	loss $1\times$	0.904	0.284
	\mathbf{x}^{proj}	$c = 1$	0.632	0.304
		$c = 2$	0.464	0.309
		$c = 5$	0.466	0.256
		$c = 10$	1.029	0.161
Ours	\mathbf{x}	loss $1\times$	0.278	0.262
		loss $2^2\times$	0.256	0.250
		loss $5^2\times$	0.240	0.249
		loss $10^2\times$	0.320	0.265
	\mathbf{x}^{proj}	$c = 1$	0.307	0.268
		$c = 2$	0.290	0.257
		$c = 5$	0.274	0.199
		$c = 10$	0.198	0.128

model as the text encoder across all tasks.

Computational resources Our GMD architecture is capable of running both the motion and trajectory models on a single commercial GPU, such as the Nvidia RTX 2080 Ti, 3080, or 3090. The trajectory model achieved a throughput of 2,048 samples per second when run on an RTX 3090, with a training time of approximately 4.34 GPU hours. Meanwhile, the motion model achieved a throughput of 256 samples per second on an RTX 3090, with a training time of around 34.7 GPU hours. The total inference time for one sample is approximately 110 seconds.

6.2. Text-to-motion generation

This section evaluates our model’s performance in the standard text-to-motion generation task and compares it with other motion DPM baselines: MotionDiffuse [62], MDM [54], MLD [8], and PhysDiff [61]. Tab. 1 shows the results where our model architecture outperforms the baselines significantly in terms of motion quality measured by FID, while maintaining similar R-Precision and Diversity.

6.3. Trajectory-conditioned generation

This section demonstrates how our emphasis projection method can address the issue of incoherent motion caused by spatial conditioning, specifically in the trajectory conditioning task, where the model is provided with ground-truth trajectories for imputation at each denoising step and is required to generate corresponding local poses. Both quantitative and qualitative results support that our emphasis projection leads to a reduction in Foot skating ratio, as evidenced in Tab. 2 and a more coherent motion in Fig. 5 compared to the MDM [54] model.

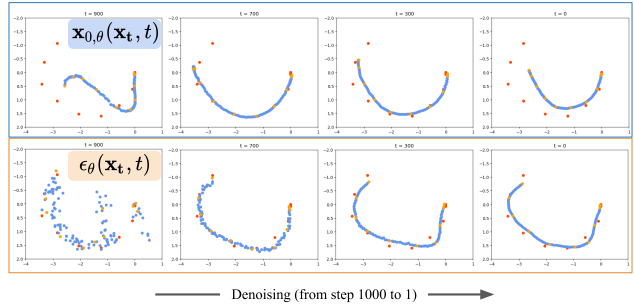


Figure 4. Comparing the evolution of the clean trajectory subject to classifier guidance from \mathbf{x}_0 and ϵ DPMs. The \mathbf{x}_0 DPM shows significant resistance on the guidance signal as exhibited by the trajectory “contraction” behavior at $t \rightarrow 0$.

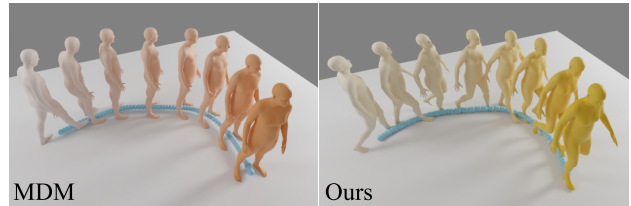


Figure 5. Generated motion, conditioned a given trajectory and text “walking forward”. MDM [54] exhibits motion incoherence where the model disregards the trajectory and generates an inconsistent motion. Our method, improved by emphasis projection, deals effectively with the conditioning.

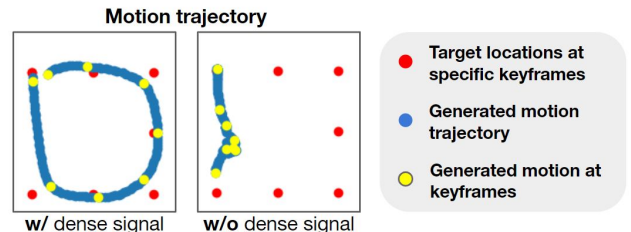


Figure 6. Generated motion trajectories, conditioned on target locations at given keyframes. Without dense signal propagation, the model ignores the target conditions.

We also compare our emphasis projection method with an alternative approach of increasing the trajectory loss strength during training. We include loss $k^2\times$ baselines, where $k \in \{1, 2, 5, 10\}$, for comparison. The results in Tab. 2 indicate that, while increasing the loss strength marginally improves both FID and Foot skating ratio, increasing it beyond a certain point leads to a decline in both FID and Foot skating ratio. By contrast, our emphasis projection method consistently leads to improvements in both metrics. We discuss this topic further in the Supplementary.

6.4. Keyframe-conditioned generation

This section evaluates the quality and adherence of the generated motion to the desired goal. A viable solution

Table 3. The effect of different conditioning strategies tested on keyframe-conditioning task. The keyframes ($N = 5$) are sampled from the ground truth motion trajectories with the same text prompts in the HumanML3D [15] test set.

Model	Conditioning	FID ↓	Foot ↓ skating ratio	Traj. ↑ diversity (m.)	Traj. err. ↓ (50 cm)	Loc. err. ↓ (50 cm)	Avg. err. ↓	R-precision ↑ (Top-3)	
MDM [54]	Single stage	$\mathbf{x} + \tau=0$	1.256	0.202	0.134	0.000	0.000	0.000	0.631
		$\mathbf{x}^{\text{proj}} + \tau=0$	2.994	0.151	0.134	0.000	0.000	0.000	0.554
		$\mathbf{x}^{\text{proj}} + \tau=100$	2.213	0.095	0.214	0.326	0.127	0.236	0.555
		$\mathbf{x}^{\text{proj}} + \text{no p2p}$	1.679	0.092	0.394	0.519	0.326	0.543	0.548
Ours (\mathbf{x}^{proj})	Single stage	$\tau=0$	0.902	0.127	0.117	0.000	0.000	0.000	0.594
		$\tau=100$	0.523	0.086	0.157	0.176	0.049	0.139	0.599
	Two stage	$\tau=100$	0.937	0.098	0.120	0.076	0.020	0.109	0.574
		$\tau=300$	0.938	0.098	0.127	0.118	0.031	0.128	0.573
		$\tau=500$	0.908	0.098	0.140	0.157	0.043	0.140	0.577
		$\tau=700$	0.898	0.098	0.162	0.196	0.058	0.153	0.580
		$\tau=900$	0.874	0.098	0.192	0.238	0.080	0.180	0.581
		no p2p	0.862	0.104	0.222	0.287	0.118	0.282	0.577

must meet both criteria to an acceptable degree.

To achieve high-quality motion, both FID and Foot skating ratio are essential since FID alone cannot adequately measure the trajectory-motion coherence. Our Emphasis projection technique significantly improves motion coherence, reducing foot skating as shown in Tab. 3 while MDM [54] is unsuitable for this task due to the high motion incoherence. Furthermore, our improved architecture significantly improves motion quality in all cases. Note that without dense signal propagation, the model ignores the keyframe conditioning as shown in Fig 6.

While a single-stage model performs reasonably well due to emphasis projection, it is too restrictive at $\tau = 0$ (forced trajectory), resulting in relatively high Foot skating. This issue can be addressed by allowing more modification (increasing to τ to 100) but at the cost of higher Loc. error.

Lastly, the trajectory model’s better controllability reduces Location error by more than half compared to the single-stage model at $\tau = 100$. As expected, increasing τ leads to more freedom in the model, resulting in increased Trajectory diversity, lower FID, and higher Location error.

6.5. Obstacle avoidance motion generation

Finally, we demonstrate our model’s ability to generate motion on additional guidance on the obstacle avoidance task. In this task, we randomly sample the target point that the human needs to reach at a specific motion step along with a set of obstacles it needs to avoid, represented as a 2D SDF (Sec. 5.3). We show the qualitative results in Fig 7.

7. Discussion and Limitations

In this work, we propose GMD, a controllable human motion generation method based on goal functions. GMD produces high-quality and diverse motions and supports diverse possibilities for goal functions. Since obtaining necessary data and designing a classifier-free learning method for non-explicit goals, such as obstacle avoidance,

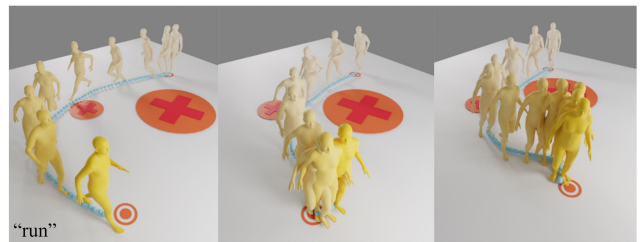


Figure 7. Qualitative results from the obstacle avoidance task given keyframe locations and obstacles. The red crossed areas represent obstacles to avoid. More results are in the supplementary.

can be challenging, our GMD utilizes a classifier-based method which allows for more conditioning flexibility without retraining the model. Thus, our studies on effective classifier guidance will be useful for further including more guiding signals.

Acknowledgement. This work was supported by the SNSF project grant 200021 204840.

References

- [1] Chaitanya Ahuja and Louis-Philippe Morency. Language2pose: Natural language grounded pose forecasting. In *2019 International Conference on 3D Vision (3DV)*, pages 719–728. IEEE, 2019. 3, 7
- [2] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *arXiv preprint arXiv:2211.09707*, 2022. 3
- [3] Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, 2019. 2
- [4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu

- Liu. eDiff-I: Text-to-Image diffusion models with an ensemble of expert denoisers. Nov. 2022. 2
- [5] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *2021 IEEE virtual reality and 3D user interfaces (VR)*, pages 1–10. IEEE, 2021. 7
- [6] T Brooks, A Holynski, and A A Efros. InstructPix2Pix: Learning to follow image editing instructions. *arXiv preprint arXiv:2211.09800*, 2022. 2
- [7] He Cao, Jianan Wang, Tianhe Ren, Xianbiao Qi, Yihao Chen, Yuan Yao, and Lei Zhang. Exploring vision transformers as diffusion learners. Dec. 2022. 3
- [8] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, Jingyi Yu, and Gang Yu. Executing your commands via motion diffusion in latent space. *arXiv preprint arXiv:2212.04048*, 2022. 1, 3, 4, 7, 8
- [9] Xin Chen, Zhuo Su, Lingbo Yang, Pei Cheng, Lan Xu, Bin Fu, and Gang Yu. Learning variational motion prior for video-based motion capture. *arXiv preprint arXiv:2210.15134*, 2022. 3
- [10] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. ILVR: Conditioning method for denoising diffusion probabilistic models. Aug. 2021. 2, 3
- [11] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. June 2022. 3
- [12] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. *arXiv preprint arXiv:2212.04495*, 2022. 1, 3
- [13] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. May 2021. 2, 3, 5, 7, 4
- [14] Yinglin Duan, Tianyang Shi, Zhengxia Zou, Yenan Lin, Zhehui Qian, Bohan Zhang, and Yi Yuan. Single-shot motion completion with transformer. *arXiv preprint arXiv:2103.00776*, 2021. 3
- [15] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022. 2, 3, 4, 7, 9
- [16] Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 580–597. Springer, 2022. 3
- [17] Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2021–2029, 2020. 3, 7
- [18] Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020. 3
- [19] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt image editing with cross attention control. Aug. 2022. 2
- [20] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. Oct. 2022. 2
- [21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. June 2020. 2, 3, 4, 7
- [22] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 3, 7
- [23] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. Apr. 2022. 2
- [24] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. May 2022. 2, 3, 4
- [25] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. Convolutional autoencoders for human motion infilling. In *2020 International Conference on 3D Vision (3DV)*, pages 918–927. IEEE, 2020. 3
- [26] Jihoon Kim, Jiseob Kim, and Sungjoon Choi. Flame: Free-form language-based motion synthesis & editing. *arXiv preprint arXiv:2209.00349*, 2022. 3
- [27] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. Sept. 2020. 2
- [28] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. Feb. 2023. 3
- [29] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. *Advances in neural information processing systems*, 32, 2019. 3
- [30] Buyu Li, Yongchi Zhao, Shi Zhelun, and Lu Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1272–1279, 2022. 3
- [31] Haoying Li, Yifan Yang, Meng Chang, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. SRDiff: Single image Super-Resolution with diffusion probabilistic models. Apr. 2021. 2
- [32] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021. 3
- [33] Jianxin Ma, Shuai Bai, and Chang Zhou. Pretrained diffusion models for unified human motion synthesis. *arXiv preprint arXiv:2212.02837*, 2022. 3
- [34] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 7

- [35] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Image synthesis and editing with stochastic differential equations. Aug. 2021. [2](#), [3](#)
- [36] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. Feb. 2021. [3](#)
- [37] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. [3](#)
- [38] William Peebles and Saining Xie. Scalable diffusion models with transformers. Dec. 2022. [3](#)
- [39] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. [3](#)
- [40] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 480–497. Springer, 2022. [3](#)
- [41] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. Sept. 2022. [2](#)
- [42] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-TTS: A diffusion probabilistic model for Text-to-Speech. May 2021. [2](#)
- [43] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizardwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022. [3](#)
- [44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. [3](#), [7](#)
- [45] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *ArXiv*, 2022. [2](#), [3](#)
- [46] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11488–11499, 2021. [2](#), [3](#)
- [47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution image synthesis with latent diffusion models. Dec. 2021. [2](#), [3](#)
- [48] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic Text-to-Image diffusion models with deep language understanding. May 2022. [2](#), [3](#)
- [49] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image Super-Resolution via iterative refinement. Apr. 2021. [2](#)
- [50] Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. Mar. 2015. [2](#)
- [51] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. July 2019. [2](#)
- [52] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based generative modeling through stochastic differential equations. Nov. 2020. [2](#)
- [53] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, Aug. 1988. [2](#)
- [54] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. Sept. 2022. [1](#), [3](#), [4](#), [7](#), [8](#), [9](#), [2](#)
- [55] Jonathan Tseng, Rodrigo Castellon, and C Karen Liu. Edge: Editable dance generation from music. *arXiv preprint arXiv:2211.10658*, 2022. [1](#), [3](#)
- [56] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9401–9411, 2021. [3](#)
- [57] Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards*. PhD thesis, King’s College, Cambridge, 1989. [2](#)
- [58] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. Oct. 2022. [2](#)
- [59] Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. Saga: Stochastic whole-body grasping with contact. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 257–274. Springer, 2022. [3](#)
- [60] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4394–4402, 2019. [3](#)
- [61] Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. Physdiff: Physics-guided human motion diffusion model. *arXiv preprint arXiv:2212.02500*, 2022. [1](#), [3](#), [7](#), [8](#)
- [62] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. [3](#), [7](#), [8](#)
- [63] Siwei Zhang, Yan Zhang, Federica Bogo, Marc Pollefeys, and Siyu Tang. Learning motion priors for 4d human body

- capture in 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11343–11353, 2021. [3](#)
- [64] Yan Zhang, Michael J Black, and Siyu Tang. Perpetual motion: Generating unbounded human motion. *arXiv preprint arXiv:2007.13886*, 2020. [3](#)
- [65] Mengyi Zhao, Mengyuan Liu, Bin Ren, Shuling Dai, and Nicu Sebe. Modiff: Action-conditioned 3d motion generation with denoising diffusion probabilistic models. *arXiv preprint arXiv:2301.03949*, 2023. [1](#), [3](#)
- [66] Rui Zhao, Hui Su, and Qiang Ji. Bayesian adversarial human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6225–6234, 2020. [3](#)
- [67] Zixiang Zhou and Baoyuan Wang. Ude: A unified driving engine for human motion generation. *arXiv preprint arXiv:2211.16016*, 2022. [1](#), [3](#)

Guided Motion Diffusion for Controllable Human Motion Synthesis

Appendix

A. Analysis on $\mathbf{x}_{0,\theta}$ vs. ϵ_θ DPMs

In this section, we discuss the differences in behavior between the $\mathbf{x}_{0,\theta}$ and ϵ_θ models used to train DPMs. While both models are capable of generating high-quality samples, their denoising processes differ significantly. In Section 4.2, we previously claimed that the \mathbf{x}_0 predicting model maximizes its influence on the outcome \mathbf{x}_{t-1} when $t \rightarrow 0$, whereas the ϵ predicting model maximizes its influence when $t \rightarrow T$. Based on this observation, we argue that the ϵ predicting model is more favorable than the \mathbf{x}_0 predicting model in circumstances where the outcome of the diffusion process will be altered by an external factor from the classifier.

To further understand the behavior of the two models, we examine Equation 1, which indicates that \mathbf{x}_{t-1} is sampled from a Normal distribution with mean

$$\mu_t = \underbrace{\frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t}}_a \mathbf{x}_0 + \underbrace{\frac{\sqrt{1-\beta_t}(1-\alpha_{t-1})}{1-\alpha_t}}_b \mathbf{x}_t \quad (14)$$

The coefficients a and b in μ_t modulate the contribution of the \mathbf{x}_0 model and the previous output \mathbf{x}_t . The larger the coefficient a is relative to b , the larger the contribution of the \mathbf{x}_0 model on the outcome of the denoising process.

In the case of an ϵ model, we substitute \mathbf{x}_0 based on the relationship $\mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon}{\sqrt{\alpha_t}}$ and get a different expression for μ_t as

$$\mu_t = \underbrace{\left(\frac{a}{\alpha_t} + b\right)}_c \mathbf{x}_t - \underbrace{\frac{a\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}}}_d \epsilon \quad (15)$$

We can see that the contribution of the \mathbf{x}_0 model and the ϵ model are starkly different, with the ϵ model having a stronger contribution on μ_t , and hence \mathbf{x}_{t-1} , where t is large, while the opposite is true for the \mathbf{x}_0 model. In other words, an ϵ model is restricted to make a smaller change over time while an \mathbf{x}_0 model can still make a large change even at the very end of the diffusion process.

From the analysis above, we conclude that the choice of modeling ϵ or \mathbf{x}_0 is no longer arbitrary. Given the fact that the classifier guidance strength is modulated by Σ_t , which is smaller as $t \rightarrow 0$, and the fact that all DPM models are biased toward their training datasets, an \mathbf{x}_0 model capable of ever larger change as the guidance signal diminishes is not an ideal choice because it could easily overpower the guidance signal, especially at the end of the diffusion process, undoing all the guidance signal. Therefore, our GMD's tra-

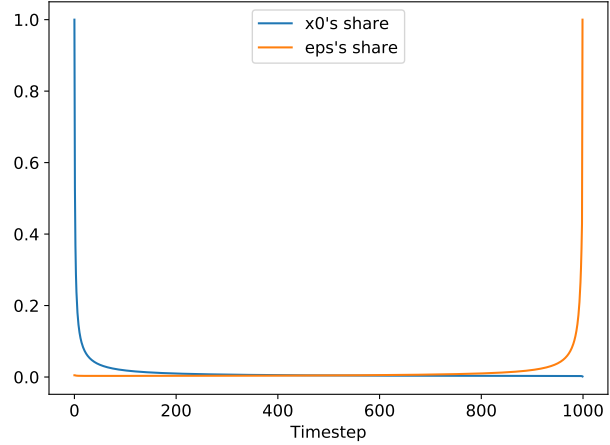


Figure A.1. Comparing \mathbf{x}_0 and ϵ contributions in the prediction of \mathbf{x}_{t-1} based on the Cosine β_i scheduler.

jectory model, which is subject to classifier guidance signals, is carefully chosen to be an ϵ model. We visualize the relative share over time of each model on μ_t in Figure A.1 and show the impact of the choice of model in Figure 3 in the main paper.

A.1. Challenges of modeling ϵ in practice

In Section A, we discussed the benefits of modeling ϵ over \mathbf{x}_0 from the perspective of classifier guidance. However, there are fundamental differences and requirements for architectures that excel in predicting \mathbf{x}_0 versus ϵ . Specifically, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is independent and full-rank, meaning there is no smaller latent manifold that it resides in. On the other hand, \mathbf{x}_0 usually has a smaller latent manifold, which is the case for many real-world data including motions as most of the possible values in $\mathbf{x} \in \mathbb{R}^{263 \times M}$ are not valid human motions, only a small subset of that is. Due to these differences, it requires special considerations for architectural design in models that successfully predict ϵ .

Although there is no sufficient reason to believe that modeling ϵ is fundamentally harder than modeling \mathbf{x}_0 , in practice, modeling ϵ is restricted to cases where its shape is relatively small compared to the latent dimension of the denoising model. For example, when modeling $\epsilon \in \mathbb{R}^{263 \times M}$ for a motion DPM, the original MDM architecture for ϵ prediction generated low-quality jagged motions compared to the same architecture for \mathbf{x}_0 prediction, which produced high-quality motions flawlessly. Increasing the latent dimension of MDM from 512 to 1,536 did not solve the problem entirely, indicating that predicting \mathbf{x}_0 and ϵ requires

different architectural designs that may not be satisfied by a single architecture. We argue that there require further studies on how to effectively design an ϵ predicting model.

However, when the space of ϵ is relatively small, such as in a trajectory DPM, the choice of architecture seems to matter less. The MDM transformer architecture was applied to trajectory modeling with relatively no problem. Ultimately, we used a convolution-based UNET with AdaGN [13] as the final architecture for our proposed method, as it demonstrated superior performance for both modeling trajectories and motions.

B. Relative vs. Absolute root representation

In this section, we discuss different ways of representing the root locations \mathbf{z} of the motion. Generally, the root locations can be represented as absolute rotations and translations (**abs**) or relative rotations and translations compared to the previous frame (**rel**). In MDM [54], the root locations are represented with relative representation following the HumanML3D [15] dataset. In this case, the global locations at an exact frame i can be obtained by a cumulative summation of rotations and translations before i .

However, we observe that representing the root with absolute coordinates (**abs**) is more favorable than the relative one (**rel**) in two aspects: being more straightforward for imputation and easier to optimize. Therefore, we adopt the absolute root representation for our models.

In **rel**, a trajectory is described as velocity $\Delta\mathbf{z}^{(i)}/\Delta i$ in the local coordinate frame of the current pelvis rotation. This representation makes each $\mathbf{z}^{(i)}$ dependent on all previous motion steps in a non-linear relationship. Optimization becomes less stable as a small change in early motion steps may compound and become a larger change later on. Also, imputing specific values becomes ill-posed since there are many possible sets of values that are satisfiable.

On the other hand, for **abs**, the imputation and optimization of \mathbf{z} become straightforward as they only involve replacing or updating $\mathbf{z}^{(i)}$ without dependency on other motion steps. We ablated the root representation by retraining MDM [54] and our model with both relative and absolute root representation, then show the results in Tab B.1. MDM shows a significant drop in performance when converted to the absolute representation, likely because the architecture is highly optimized for the relative representation, while for our models, the representation change results in a trade-off between the *FID* and *R-precision*.

Lastly, we note that the use of absolute root representation is necessary for our final model as the spatial guidance is done via a combination of imputation and optimization.

Table B.1. Text-to-motion evaluation on the HumanML3D [15] dataset. Comparison between relative and absolute root representation. The right arrow \rightarrow means closer to real data is better.

	FID \downarrow	R-precision \uparrow (Top-3)	Diversity \rightarrow
Real	0.002	0.797	9.503
MDM [54] (rel)	0.556	0.608	9.446
MDM [54] (abs)	0.894	0.638	8.819
Ours (rel)	0.305	0.666	9.861
Ours (abs)	0.212	0.670	9.440
Ours \mathbf{x}^{proj}	0.235	0.652	9.726

C. Analysis on Emphasis projection

In this section, we discuss in greater detail our proposed Emphasis projection. Conceptually, we wish to increase the relative importance of the trajectory representation \mathbf{z} within the motion representation \mathbf{x} . This could be done most simply by increasing the magnitude of those values of \mathbf{z} by multiplying it with a constant $c > 1$. More precisely, let us assume the shape of x is $263 \times M$. A single motion frame $x = \mathbf{x}^{(i)}$ is a column vector of 263 scalars in which 3 elements (rot, x , z) are a column vector of a trajectory frame $z = \mathbf{z}^{(i)}$ that comprises root rotation and a ground location. The new trajectory elements become $z \times c$.

How to calculate a suitable scalar c ?

By introducing a scalar $c > 1$, the trajectory elements z are given a higher relative importance than the remaining 260 elements in x . This relative importance is determined by the cumulative variance of the z elements compared to that of the remaining 260 elements. Assuming that all elements in x are independently and identically distributed according to a standard Normal distribution $\mathcal{N}(0, 1)$, we can represent the cumulative variance of the trajectory elements as

$$\text{Var}[x^{(\text{rot})} + x^{(x)} + x^{(z)}] = \sum_{j \in \text{Traj.}} \text{Var}[x^{(j)}] = 3 \quad (16)$$

where $j \in \text{Traj.}$ refers to the indexes in x that are related to trajectory.

Similarly, we can represent the cumulative variance of the remaining 260 elements as $\text{Var}[\sum_{j \notin \text{Traj.}} x^{(j)}] = 260$, where $j \notin \text{Traj.}$ refers to the indexes in x that are not related to trajectory.

When we multiply trajectory by c , the new cumulative variance becomes $\text{Var}[c \times (x^{(\text{rot})} + x^{(x)} + x^{(z)})] = c^2 \sum_{j \in \text{Traj.}} \text{Var}[x^{(j)}] = 3c^2$. Therefore, the relative importance of the scaled trajectory elements compared to the remaining 260 elements in x is given by the expression

$$\frac{3c^2}{260 + 3c^2} \quad (17)$$

Setting $c = \sqrt{\frac{260}{3}} \approx 9.3$ results in a relative importance of 50%, which strikes a reasonable balance between the trajectory and the rest of human motion. We have selected $c = 10$ as a rounded number of this fact, and it has been found to work well in practice.

Maintaining the uniform unit variance after scaling

After scaling up the trajectory elements by a factor of c , the variance of the new motion representation is no longer uniform. This presents a problem when trying to model it using the original DPM’s β_t scheduler. In order to maintain uniform variance, we can redistribute the increased values from the trajectory part $c \times z$ to the rest in x via a random matrix projection.

There are two reasons why a random matrix projection is a good choice. First, it maintains the distance measure of the original space with high probability, meaning that the properties of the motion representation remain relatively unchanged. Second, a random matrix projection is easy to obtain and linear. It has an exact inverse projection, which ensures that there is no loss of information after the projection.

Finally, to maintain unit variance, we scale down the entire vector uniformly by a factor of $\frac{1}{263-3+3c^2}$.

C.1. Trajectory loss scaling

One approach to increase the emphasis on the trajectory part $\mathbf{z}^{(i)}$ of the motion $\mathbf{x}^{(i)}$ is to scale the reconstruction loss of only the trajectory part during the training of the motion DPM. This method does not change the representation but can potentially increase the model’s emphasis on the trajectory part of the motion compared to the rest of the motion.

To compare the loss scaling method with the proposed Emphasis projection, we formulate a new loss function for a specific motion frame i , which increases the trajectory importance by a factor of k . This is given by the equation:

$$\mathcal{L}_k^{(i)} = \sum_{j \in \text{Traj.}} \left\| k\hat{x}^{(j)} - kx^{(j)} \right\|^2 + \sum_{j \notin \text{Traj.}} \left\| \hat{x}^{(j)} - x^{(j)} \right\|^2 \quad (18)$$

Here, $\hat{x} = \mathbf{x}_{0,\theta}(\mathbf{x}_t)^{(i)}$ represents the i -th motion frame of the DPM’s prediction and $x = \mathbf{x}_0^{(i)}$ represents the i -th motion frame of the ground truth motion. The value of k multiplies inside the squared loss, resulting in k^2 times more importance on the trajectory part of the motion. For example, setting $k = 10$ would increase the importance of the trajectory part by 100-fold, which has the same scaling effect as setting $c = 10$ in Emphasis projection. Hence, the reasonable range of k is the same as that of c .

In the main text, we experimented with $k \in 1, 2, 5, 10$ and found that Emphasis projection consistently outperformed loss scaling regarding motion coherence.

D. GMD’s Model Architecture

The trajectory and motion architectures of GMD are both based on UNET with Adaptive Group Normalization (AdaGN), which was originally proposed by [13] for class-conditional image generation tasks. However, we have adapted this model for sequential prediction tasks by using 1D convolutions. It should be noted that our architectures share some similarities with [24] with the addition of AdaGN. The architecture overview is depicted in Figure D.1 while the Adaptive Group Normalization is depicted in Figure D.2. The hyperparameter settings of the two DPMs are shown in Table D.1. We currently are in the process of open-sourcing the code base of GMD.

Convolution-based architectures are commonly used in state-of-the-art image-domain DPMs, such as those proposed by [47] and [48]. On the other hand, transformer-based architectures, which were used in the original MDM proposed by [54], are not well-studied architectures for DPMs [7, 38].

Our proposed architecture alone has led to a significant improvement in motion generation tasks, reducing the Fréchet Inception Distance (FID) by more than half compared to the original MDM (0.556 vs 0.212), as shown in Table 1 in the main paper.

Table D.1. Network architecture of our GMD’s models based on the proposed 1D UNET with AdaGN.

Parameter	Trajectory DPM	Motion DPM
Batch size	512	64
Base channels	512	512
Channel multipliers	[0.125, 0.25, 0.5]	[2, 2, 2, 2]
Attention resolution	No attention	No attention
Samples trained	32M	
β scheduler	Cosine [36]	
Learning rate	1e-4	
Optimizer	AdamW (wd = 1e-2)	
Training T	1000	
Diffusion loss	ϵ prediction	\mathbf{x}_0 prediction
Diffusion var.	Fixed small $\tilde{\beta}_t = \frac{1-\alpha_t-1}{1-\alpha_t} \beta_t$	
Model avg. beta	0.9999	

E. Training details

GMD’s models. We used a batch size of 64 for motion models and a batch size of 512 for trajectory models. No dropout was used in all of the GMD’s models: both trajectory and motion. We used AdamW with a learning rate of 0.0001 and weight decay of 0.01. We clipped the gradient norm to 1 which was found to increase training stability.

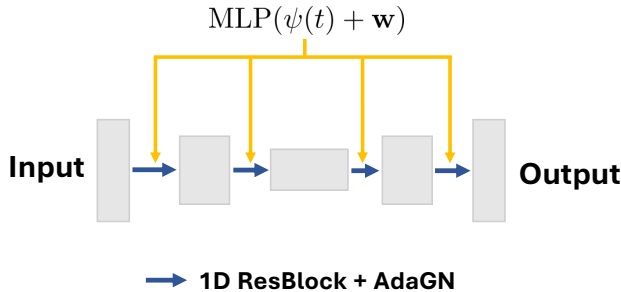


Figure D.1. A simplified overview of our GMD’s 1D UNET + AdaGN architecture that is designed to process two input signals: the time step $\psi(t)$ and a text-prompt embedding \mathbf{w} . The time step is encoded using sinusoidal functions, while the text-prompt embedding is generated by the CLIP text encoder model, as described in [54]. The ResBlock + AdaGN component of the model is explained in Figure D.2.

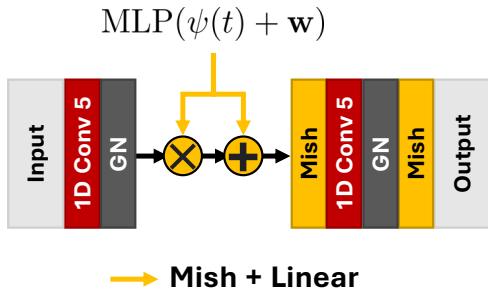


Figure D.2. A single 1D ResBlock with Adaptive Group normalization (AdaGN) [13]. The conditioning signal from the MLP, shared across all ResBlocks, is projected by first applying a Mish activation and then a resizing linear projection specific to each ResBlock. All kernel sizes are 5. We use Mish activation function following [24].

We used mixed precision during training and inference. We trained all motion models for 32,000,000 samples (equivalent to 500,000 iterations at batch size 64, and 62,500 iterations at batch size 512). We also employed the moving average of models during training ($\beta = 0.9999$) [21] and used the averaged model for better generation quality. Do note that our model architecture still improves over the baseline MDM without the moving average.

GMD’s trajectory model. While the crucial trajectory elements are only the ground x-z locations, we have found it useful to train the trajectory model with all four components (rot, x, y, z). The additional (rot, y) seem to provide useful information that helps the model learn and reduce overfitting in the trajectory model. Note that the trajectory DPM is sensitive to the overfitting problem. Overtraining the model will result in a strong trajectory bias in the model making the model more resistant to classifier guidance and imputation. Our choice of training the trajectory model for 32,000,000 samples was carefully chosen based on this ob-

servation.

Retraining of MDM models. We retrained the original MDM using our absolute root representation and proposed Emphasis projection as the two main baselines. In order to maintain consistency, we kept the original optimization settings for the MDM models. Specifically, we used AdamW optimizer with a learning rate of 0.0001 and without weight decay. We found that gradient clipping of 1 provided more stability, so we also applied it here. We did not utilize mixed precision training for these models. To match the settings of the original MDM, we trained these models for 400,000 iterations at a batch size of 64.

F. Inferencing details

We have chosen the value of s as 100 for the classifier guidance strength. Our experiments have shown that this value of s performs well within the range of 100 to 200. For all our goal functions G_x , we always used the $p = 1$ norm. Whenever feasible, we implemented both imputation and classifier guidance concurrently. However, we ceased the guidance signals, i.e., classifier guidance and imputation, at $t = 20$ as this led to a slight improvement in the motion coherence.

Obstacle avoidance task. In this particular case, it was not feasible to create a point-to-point trajectory because doing so could potentially lead to a collision with an obstacle. As a result, we decided against utilizing any point-to-point trajectory imputation for this task.