Guided Motion Diffusion for Controllable Human Motion Synthesis

Korrawe Karunratanakul1  Konpat Preechakul2  Supasorn Suwajanakorn2  Siyu Tang1

1ETH Zürich, Switzerland  2VISTEC, Thailand

https://korrawe.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 [11, 62, 64] outperforming other alternatives such as GAN or VAE in terms of both quality and diversity [7, 51, 58]. Several studies have focused on generating motion based on expressive text prompts [7, 51], or music [52, 64]. The state-of-the-art motion generation methods, such as MDM [51], 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 [43], usually with more focus on local poses. For instance, the common motion representation [14] 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 [50, 54], 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 backpropagating 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 [20, 47, 48, 49] 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 [12], image super-resolution [30, 46], speech synthesis [26, 26, 39], video generation [19, 22], 3D shape generation [38, 55], and reinforcement learning [25].

The surge in interest in DPMs may be attributed to their impressive controllable generation capabilities, including text-conditioned generation [42, 44, 45] and image editing [4, 6, 9, 18, 34]. Latent diffusion models (LDM) are an-
other area of interest, which includes representation learning [27, 40] and more efficient modeling techniques [7, 44].

Moreover, DPMs exhibit a high degree of versatility in terms of conditioning. There are various methods for conditional generation, such as imputation and inpainting [9, 10, 34], classifier guidance [10, 12], and classifier-free guidance [21, 42, 44, 45]. 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 [13, 17, 51], trajectories [24, 53, 60], images [8, 43], music [28, 29, 31], text [1, 14, 15, 25, 37], objects [56], action labels [16, 36], or unconditioned [35, 57, 61, 63]. Recently, many diffusion-based motion generation models have been proposed [11, 25, 32, 62, 64] and demonstrate better quality compared to alternative models such as GAN or VAE. Employing the CLIP model [41], these models showed great improvements in the challenging text-to-motion generation task [7, 58, 59] as well as allowing conditioning on partial motions [51] or music [2, 52]. 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 $x_t$ with varying noise levels $t$. The noising process of DPM is defined cumulatively as $q(x_t | x_0) = \mathcal{N}(\sqrt{\alpha_t} x_0, (1 - \alpha_t) I)$, where $x_0$ is the clean input, $\alpha_t = \prod_{s=1}^t (1 - \beta_s)$, and $\beta_t$ is a noise scheduler. The denoising model $p_\theta(x_{t-1} | x_t)$ with parameters $\theta$ learns to reverse the noising process by modeling the Gaussian posterior distribution $q(x_{t-1} | x_t, x_0)$. DPMs can map a prior distribution $\mathcal{N}(0, I)$ to any distribution $p(x)$ after $T$ successive denoising steps.

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

$$\mu_t = \frac{\sqrt{\alpha_t - \beta_t}}{1 - \alpha_t} x_0 + \frac{\sqrt{1 - \beta_t (1 - \alpha_t^{-1})}}{1 - \alpha_t} x_t$$

and $\Sigma_t$ is a variance scheduler of choice, usually $\Sigma_t = \frac{1 - \alpha_t^{-1}}{1 - \alpha_t} \beta_t$ [20]. $x_0$ in Eq. 1 is the prediction from a denoising model. For an $\epsilon_\theta$ model, $x_0 = \frac{1}{\sqrt{\alpha_t}} x_t + \sqrt{1 - \alpha_t} \epsilon_\theta(x_t)$.

There are multiple choices for the denoising model to predict including the clean input $x_0$, the noise $\epsilon$, and the one-step denoised target $\mu_t$. An $x_{0,\theta}$ model is trained using the squared loss to the clean input $\|x_{0,\theta}(x_t) - x_0\|^2$, an $\epsilon_\theta$ model is trained using the squared loss $\|\epsilon_\theta(x_t) - \epsilon\|^2$, and $\mu_{t,\theta}$ model is trained using the squared loss $\|\mu_{t,\theta}(x_t) - \mu_t\|^2$.

### 3.2. Controllable generation with diffusion models.

#### Classifier-free guidance. The conditioning signals are treated as additional inputs to the denoiser $p_\theta(x_{t-1} | x_t, d)$ 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 [44, 45, 51]. To draw samples, the effective denoiser becomes $\tilde{\epsilon}_\theta(x_t, d) = w \epsilon_\theta(x_t, d) + (1 - w) \epsilon_\theta(x_t, \emptyset)$, where $w$ controls the conditional strength. The new $\tilde{\epsilon}_\theta$ 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 $p(x_{t-1} | x_t, d)$ from $p_\theta(x_{t-1} | x_t) p(d | x_t)$ [12], where $p(d | x_t)$ is any probability function that we can approximate its score function $\nabla_{x_t} \log p(d | x_t)$ effectively. The new sampling process is similar to the original (Eq. 1) but with the mean shifted by the scaled score function as

$$\mu_t = \mu'_t + s \Sigma_t \nabla_{x_t} \log p(d | x_t)$$

where $\mu'_t$ is the original mean, $s$ controls the conditioning strength, and $\Sigma_t$ is a variance scheduler which can be the same as in Eq. 1. Since $\Sigma_t$ is a decreasing sequence, the guidance signal diminishes as $t \to 0$ which corresponds to the characteristic of DPMs that tend to modify $x_t$ 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 $p(d | x_t)$ 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 $y$ be a partial target value in an input $x$ that we want to impute. The imputation region of $y$ on $x$ is denoted by $M'^y$, and a projection $P'^y$ that resizes $y$ to that of $x$ by filling in zeros. In DPMs, imputation can be done on the sample $x_{t-1}$ after every denoising step [10]. We have the new imputed sample $\hat{x}_{t-1}$ as

$$\hat{x}_{t-1} = (1 - M'^y) \odot x_{t-1} + M'^y \odot P'^y y_{t-1}$$
where $\odot$ is a Hadamard product and $y_{t-1}$ is a noised target value. $y_{t-1} \sim \mathcal{N}(\alpha_{t-1}y_{t}, (1 - \alpha_{t-1})I)$ following Ho et al. [20] is one of the simplest choices of $y_{t-1}$.

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 $z_{0,,t}$, a motion DPM $x_{0,,t}$, a goal function $G_{z}()$, and keyframe locations $y$ (if any).

1: # Stage 1: Trajectory generation
2: $z_{0} \sim \mathcal{N}(0, I)$
3: for all $t$ from $T$ to 1 do
4: $z_{0} \leftarrow z_{0,,t}$
5: $\mu, \Sigma \leftarrow \mu(z_{0}, z_{t}), \Sigma_{t}$
6: # Classifier guidance (Eq. 2)
7: # Dense signal propagation
8: $z_{t-1} \sim \mathcal{N}(\mu - s\Sigma\nabla_{z_{t}}G_{z}(z_{0}), \Sigma)$
9: # Impute $y$ on $z$ (Eq. 3) (if any)
10: $z_{t-1} \leftarrow (1 - M_{g}^{t}) \odot z_{t-1} + M_{g}^{t}y_{t-1}$
end for

12: # Stage 2: Trajectory-conditioned motion generation
13: $x_{t}^{proj} \leftarrow$ sample from $\mathcal{N}(0, I)$
14: for all $t$ from $T$ to 1 do
15: $M \leftarrow P_{za}^{t}M_{g}^{t}$
16: $x_{0}^{proj} \leftarrow x_{0,0}^{proj}(x_{t}^{proj})$ # Emphasis projection
17: # Impute $y$ on $x^{proj}$ (Eq. 6)
18: $\hat{x}_{t}^{proj} \leftarrow A((1 - M) \odot A^{-1}x_{0}^{proj} + MP_{za}^{t}x_{t})$
19: $\mu, \Sigma \leftarrow \mu(\hat{x}_{t}^{proj}, x_{t}^{proj}), \Sigma_{t}$
20: # Masked classifier guidance (Eq. 9)
21: # Dense signal propagation
22: $\Delta \leftarrow -s\Sigma A^{-1}\nabla_{x_{0}^{proj}}G_{z}(P_{za}^{t}A^{-1}x_{0}^{proj})$
23: $\mu \leftarrow \mu + (1 - \hat{M}) \odot \Delta$
24: $x_{t-1} \sim \mathcal{N}(\mu, \Sigma)$
end for
26: return $z_{0}$

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 [7, 51], 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 $G_{z}()$ that takes a motion representation $x$ and measures how far the motion $x$ is from the goal (lower is better). More specifically, $x \in \mathbb{R}^{N \times M}$ 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 [14] dataset. Let $X$ be the random variable associated with $x$. Our goal is to model the following conditional probability using a motion DPM

$$p(x|G_{x}(X) = 0)$$

This can be extended to $p(x|G_{x}(X) = 0, d)$, 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 $G_{x}$ that only depends on the trajectory $z$ of the human motion, not the whole motion $x$. Let us define $z \in \mathbb{R}^{L \times M}$ to be the trajectory part of $x$ with length $M$ and $L = 2$ describing the ground location of the pelvis of a human body. A particular location $z^{(t)}$ at motion step $t$ describes the pelvis location of the human body on the ground plane. We define a projection $P_{za}^{t}$ that resizes $z$ to match $x$ by taking only the $z$ part, and its reverse $P_{za}^{t}$ that resizes $z$ to match $x$ by filling in zeros. With this, our conditional probability becomes $p(x|G_{x}(P_{za}^{t}X) = 0)$.

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 $G_{x}()$ is by analyzing what trajectories that minimize $z^{*} = \arg\min_{z}G_{x}(z)$ look like. For a trajectory conditioning task, a whole trajectory $z^{*}$ is directly given. Our task is to generate the rest of the motion $x$. With such knowledge, we can employ imputation & inpainting technique by supplying the motion DPM with the x-shaped $P_{za}^{t}z^{*}$ to guide the generation process.

Problem 1: Motion incoherence

Since the imputing trajectory $z^{*}$ is only a small part of the whole motion $x$ ($L \ll N$), we often observe that the DPM ignores the change from imputation and fails to make appropriate changes on the rest of $x$. 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 $x$. More specifically, we propose an Emphasis projection method that increases the trajectory’s
relative importance within motion $x$. 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, $(\text{rot}, x, z)$, in $x$ by $c$ times. We now have a projected motion $x_{\text{proj}} = \frac{1}{1 + \alpha_t} A x$. Note that the fractional term is to maintain the unit variance on $x_{\text{proj}}$. The noising process of the projected motion becomes $q(x_{\text{proj}} | x_{\text{proj}}) = \mathcal{N}(\sqrt{\alpha_t} x_{\text{proj}}, (1 - \alpha_t) I)$. There is no change on how a DPM that works on the projected motion $p_{\theta}(\bar{x}_{t-1}^{\text{proj}} | x_{\text{proj}})$ operates and treats $x_{\text{proj}}^{\text{proj}}$.

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 $x_{\text{proj}}$**. We have discussed imputing on the sample $x_{t-1}$ in Eq. 3. Here, we introduce an imputation on $x_0$ which modifies the DPM’s belief on the final outcome $x_{0, \theta}$ by imputing it with $z$. We have found this technique useful in many tasks we are interested in.

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

$$\hat{x}_0 = (1 - M_z^x) \odot x_{0, \theta} + M_z^x \odot P_z^x z^*$$

(5)

Now operating on the projected motion $x_{\text{proj}}$, before we can do imputation, we need to unproject it back to the original motion using $x_0 = A^{-1} x_{\text{proj}}^{\text{proj}}$, and then project the imputed $x_0$ back using $\hat{x}_{0, \theta}^{\text{proj}} = A \hat{x}_0$. We obtain the imputed motion under emphasis projection $\hat{x}_0^{\text{proj}}$ from

$$\hat{x}_0^{\text{proj}} = A \left( (1 - M_z^x) \odot (A^{-1} x_{0, \theta}^{\text{proj}} + M_z^x \odot P_z^x z^*) \right)$$

(6)

Substituting $x_0^{\text{proj}}$ into Eq. 1, we obtain the new mean $\tilde{\mu}_t^{\text{proj}}$ for sampling $x_{t-1}^{\text{proj}} \sim \mathcal{N}(\tilde{\mu}_t^{\text{proj}}, \Sigma_t)$.

**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 $x_{t-1}$ toward a region with lower $G_z$. This trick is called classifier guidance [12]. The direction of change corresponds to a score function $\nabla_{x_t} \log p(z | x_t) = 0 | x_t)$ which can be approximated as a direction $\Delta_{x_0} = -\nabla_{x_0} G_z(P_z^x z_{0, \theta})$ that reduces the goal function. We can guide the generative process by nudging the DPM’s prediction as $x_0 = x_{0, \theta} + \Delta_{x_0}$. While imputation requires the minimizer $z^*$ 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 $x$. For example, in trajectory-conditioned generation, $z$ 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 \( f(x_t) = x_0 \), 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:
\[
\nabla_{x_t} \log p(G_z(X_t) = 0|x_t) \approx -\nabla_{x_t} G_z\left( P^z_x f(x_t) \right) \tag{7}
\]

While an external function can be used as \( f \), we observe that the existing DPM model \( x_{0,\theta}(x_t) \) 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 \( x_t \) through \( x_{0,\theta}(x_t) \) 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
\[
\mu_t = \tilde{\mu}_t - (1 - M^z_x) \odot s_{\sum_i} G_z(P^z_x f(x_t)) \tag{8}
\]
where \( \tilde{\mu} \) is an imputed sampling mean. By replacing \( \mu \) with \( \mu_{\text{proj}} \), we get classifier guidance together with imputation that works with emphasis projection as
\[
\Delta_{\mu} = -s_{\sum_i} A \nabla_{x_{t,\text{proj}}} G_z(P^z_x A^{-1} f(x_{t,\text{proj}})) \tag{9}
\]
\[
\mu_{\text{proj}} = \mu_{\text{proj}} + A(1 - M^z_x) \odot \Delta_{\mu} \tag{10}
\]

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 \( \Sigma_t \) (Eq. 2).

We investigate the coefficient \( \sqrt{\frac{M^z_{\Theta}}{1 - M^z_{\Theta}}} \) of \( x_0 \) in the sampling mean \( \mu_t \) (Eq. 1). This coefficient reaches its maximum value at \( t = 0 \), meaning that an \( x_{0,\theta} \) 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 \( \epsilon_\Theta \) 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 \( \epsilon_\Theta \) instead of \( x_{0,\theta} \) 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 \( x \) that matches a given trajectory \( z \). Our objective is to minimize the distance between the generated motion and the given trajectory, which we define as
\[
G_z(x) := \left\| z - P^z_x x \right\|_p \tag{11}
\]

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 \( y \in \mathbb{R}^{2 \times M} \) be a trajectory describing keyframe locations and a mask \( M^y_z \) describe the key motion steps. Our goal function of a motion \( x \) is
\[
G_z(x) := \sum_t \left\| M^y_z (P^z_x x - y) \right\|_p \tag{12}
\]

Consequently, \( G_z(z) = \sum_t \left\| M^y_z (z - y) \right\|_p \). Due to the partial trajectory \( y \), the imputation region of \( y \) on \( x \) becomes \( M^y_z = P^z_x M^y_z \).

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 \( z \) that satisfies the keyframe locations and then generate the motion \( x \) 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 \( y \): 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 \( \tau \) is large enough, the DPM will adjust the given trajectory to a higher quality one. However, if \( \tau \) is too large, the DPM may generate a motion that does not perform well on \( G_t \) point-to-point trajectory can serve as a useful guidance signal to generate the trajectory under the guidance signal from \( G_z \). We train 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 \( \tau \), 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 \( G_x^{\text{loc}} \) (defined as in Eq. 12), and another that pushes back when the human model contacts 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 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 [14] dataset, which is a collection of text-annotate motion sequences from AMASS [33] and HumanAct12 [16] 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 [12] depicted in details in the Supplementary. The motion DPM is an \( x \) model, while the trajectory DPM is an \( \epsilon \) model, as explained in Section 4.2, to enhance controllability. We utilized DDPM [20] 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 [21] manner, similar to MDM [51], and use the CLIP [41]
Table 2. Trajectory-conditioned motions evaluation. The ground truth trajectory is used for imputing after each diffusion step. Comparing the effect of an original $x$ with emphasis loss functions to the emphasis projection $x^{\text{proj}}$ after imputing whole trajectories after each diffusion step.

<table>
<thead>
<tr>
<th>Model</th>
<th>Space</th>
<th>Emphasis</th>
<th>FID ↓</th>
<th>Foot skating ↓ ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x)</td>
<td>loss (1 \times)</td>
<td>0.904</td>
<td>0.284</td>
<td></td>
</tr>
<tr>
<td>MDM [51]</td>
<td>(c = 1)</td>
<td>0.632</td>
<td>0.304</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c = 2)</td>
<td>0.464</td>
<td>0.309</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c = 5)</td>
<td>0.466</td>
<td>0.256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c = 10)</td>
<td>1.029</td>
<td>0.161</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>loss (1 \times)</td>
<td>0.278</td>
<td>0.262</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loss (2^2 \times)</td>
<td>0.256</td>
<td>0.250</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loss (5^2 \times)</td>
<td>0.240</td>
<td>0.249</td>
<td></td>
</tr>
<tr>
<td></td>
<td>loss (10^2 \times)</td>
<td>0.320</td>
<td>0.265</td>
<td></td>
</tr>
<tr>
<td>(x^{\text{proj}})</td>
<td>(c = 1)</td>
<td>0.307</td>
<td>0.268</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c = 2)</td>
<td>0.290</td>
<td>0.257</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c = 5)</td>
<td>0.274</td>
<td>0.199</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c = 10)</td>
<td>(0.198)</td>
<td>(0.128)</td>
<td></td>
</tr>
</tbody>
</table>

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 [59], MDM [51], MLD [7], and PhysDiff [58]. 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 [51] model.

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 [14] test set.

<table>
<thead>
<tr>
<th>Model</th>
<th>Conditioning</th>
<th>FID ↓</th>
<th>Foot ↓ skating ratio</th>
<th>Traj. ↑ diversity (m.)</th>
<th>Traj. err. ↓ (50 cm)</th>
<th>Loc. err. ↓ (50 cm)</th>
<th>Avg. err. ↓</th>
<th>R-precision ↑ (Top-3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDM [51]</td>
<td>Single stage $x + \tau=0$</td>
<td>1.256</td>
<td>0.202</td>
<td>0.134</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.631</td>
</tr>
<tr>
<td></td>
<td>$x^{proj} + \tau=0$</td>
<td>2.994</td>
<td>0.151</td>
<td>0.134</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.554</td>
</tr>
<tr>
<td></td>
<td>$x^{proj} + \tau=100$</td>
<td>2.213</td>
<td>0.095</td>
<td>0.214</td>
<td>0.326</td>
<td>0.127</td>
<td>0.236</td>
<td>0.555</td>
</tr>
<tr>
<td></td>
<td>$x^{proj} + \text{no p2p}$</td>
<td>1.679</td>
<td>0.092</td>
<td>0.394</td>
<td>0.519</td>
<td>0.326</td>
<td>0.543</td>
<td>0.548</td>
</tr>
<tr>
<td>Ours ($x^{proj}$)</td>
<td>Single stage $\tau=0$</td>
<td>0.902</td>
<td>0.127</td>
<td>0.117</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.594</td>
</tr>
<tr>
<td></td>
<td>$\tau=100$</td>
<td>0.523</td>
<td>0.086</td>
<td>0.157</td>
<td>0.176</td>
<td>0.049</td>
<td>0.139</td>
<td>0.599</td>
</tr>
<tr>
<td></td>
<td>Two stage $\tau=100$</td>
<td>0.937</td>
<td>0.098</td>
<td>0.120</td>
<td>0.076</td>
<td>0.020</td>
<td>0.109</td>
<td>0.574</td>
</tr>
<tr>
<td></td>
<td>$\tau=300$</td>
<td>0.938</td>
<td>0.098</td>
<td>0.127</td>
<td>0.118</td>
<td>0.031</td>
<td>0.128</td>
<td>0.573</td>
</tr>
<tr>
<td></td>
<td>$\tau=500$</td>
<td>0.908</td>
<td>0.098</td>
<td>0.140</td>
<td>0.157</td>
<td>0.043</td>
<td>0.140</td>
<td>0.577</td>
</tr>
<tr>
<td></td>
<td>$\tau=700$</td>
<td>0.898</td>
<td>0.098</td>
<td>0.162</td>
<td>0.196</td>
<td>0.058</td>
<td>0.153</td>
<td>0.580</td>
</tr>
<tr>
<td></td>
<td>$\tau=900$</td>
<td>0.874</td>
<td>0.098</td>
<td>0.192</td>
<td>0.238</td>
<td>0.080</td>
<td>0.180</td>
<td>0.581</td>
</tr>
<tr>
<td></td>
<td>no p2p</td>
<td>0.862</td>
<td>0.104</td>
<td>0.222</td>
<td>0.287</td>
<td>0.118</td>
<td>0.282</td>
<td>0.577</td>
</tr>
</tbody>
</table>

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 [51] 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 $\tau$ 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 $\tau$ 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, 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

Li. eDiff-I: Text-to-Image diffusion models with an ensemble of expert denoisers. Nov. 2022. 2


[27] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion models already have a semantic latent space. Feb. 2023. 3


[34] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Junyan Zhu, and Stefano Ermon. SDEdit: Image synthesis and


[38] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. Sept. 2022. 2


[49] 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


