DOI: 10.1111/cgf.70093

COMPUTER GRAPHICS *forum*
Volume 44 (2025), number 6, e70093

LEAD: Latent Realignment for Human Motion Diffusion

Nefeli Andreou,[1,2,3,*] Xi Wang,[2] Victoria Fernández Abrevaya,[3] Marie-Paule Cani,[2] Yiorgos Chrysanthou[4] and Vicky Kalogeiton[2]

[1]University of Cyprus, Nicosia, Cyprus
nefeliandreou@outlook.com
[2]LIX, École Polytechnique, CNRS, Institut Polytechnique de Paris, Palaiseau, France
xi.wang.robotics@gmail.com, marie-paule.cani@polytechnique.edu, vicky.kalogeiton@gmail.com
[3]Max Planck Institute for Intelligent Systems, Tübingen, Germany
vabrevaya@gmail.com
[4]CYENS - Centre of Excellence, Nicosia, Cyprus
y.chrysanthou@cyens.org.cy

Abstract

Our goal is to generate realistic human motion from natural language. Modern methods often face a trade-off between model expressiveness and text-to-motion (T2M) alignment. Some align text and motion latent spaces but sacrifice expressiveness; others rely on diffusion models producing impressive motions but lacking semantic meaning in their latent space. This may compromise realism, diversity and applicability. Here, we address this by combining latent diffusion with a realignment mechanism, producing a novel, semantically structured space that encodes the semantics of language. Leveraging this capability, we introduce the task of textual motion inversion to capture novel motion concepts from a few examples. For motion synthesis, we evaluate LEAD on HumanML3D and KIT-ML and show comparable performance to the state-of-the-art in terms of realism, diversity and text-motion consistency. Our qualitative analysis and user study reveal that our synthesised motions are sharper, more human-like and comply better with the text compared to modern methods. For motion textual inversion (MTI), our method demonstrates improvements in capturing out-of-distribution characteristics in comparison to traditional VAEs.

Keywords: animation, animation; motion capture, animation; motion control

CCS Concepts: • Computing methodologies → Motion capture; Activity recognition and understanding; Learning paradigms

1. Introduction

Text-to-motion (T2M) generation is the process of creating human-like motion that reflects a given language instruction. Generating motions that comply with textual descriptions is a task that received significant attention [CJL*23, TRG*23, PBV22, APBV22] due to its potential to democratise 3D content creation and its numerous applications in fields such as robotics [PMA18], entertainment [HSK16, HKS17] and virtual reality [BRB*21].

A T2M model should be able to accurately reproduce arbitrary descriptions in natural language while accounting for the many-to-many nature of the problem. This is a challenging task, since there is a large discrepancy between the space of natural language and the space of human motions (i.e., skeletal poses) [TGH*22, AM19, CJL*23]. Solutions can be classified into two main categories. Initial works such as TEMOS [PBV22], L2JP [AM19] and MotionCLIP [TGH*22] build a common latent space that simultaneously encodes natural language and motion. These approaches typically rely on (variational) autoencoders [KW13] and are thus restricted in the distribution they can model, where the mapping from text to motion is either assumed to be one-to-one [GSAH17, AM19] or to follow a normal distribution in the latent space [PBV22].

A second category leverages diffusion models to learn the distribution of human motions conditioned on text, offering enhanced diversity and realism surpassing the previous constraints [CJL*23, KKC23, ZCP*22, TRG*23]. Diffusion models provide a probabilistic framework that captures the multimodal relationship between text and motion. The iterative refinement process in diffusion models enhances the fidelity and diversity of the generated motion. This structured and gradual approach to motion generation allows users to intervene at different stages of the generation to make

[*]Work done prior to joining Amazon.

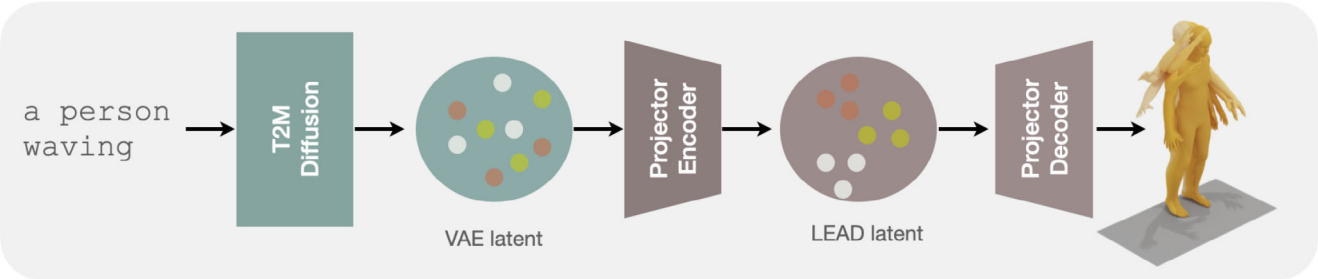


Figure 1: *LEAD enhances human motion diffusion models by introducing a latent realignment scheme that enforces semantic consistency in the latent space. We show that this leads to improved realism and expressiveness in the tasks of motion generation and motion inversion.*

adjustments without compromising the overall coherence of the generated motion. Despite their impressive generations and improvement in generations, we observe that the synthesised animations might not consistently follow the input text and may still generate unrealistic movement. We hypothesise that a semantically structured motion latent space, that is, one that inherits some of the rich properties of the language space, can facilitate and improve the task of T2M generation.

In the image domain, diffusion models have gone beyond general T2M generation towards *personalised* generation, where the goal is to synthesise concepts that are hard to describe (e.g., a specific instance of an object) based on a few examples. The work of Gal *et al.* [GAA*22] was among the first to address this through an optimisation approach, where the unknown embedding in the language space is learnt given a set of examples. This *inversion* approach that recovers the embedding of a concept could certainly be beneficial for the motion domain, too. If one could invert a motion diffusion model, the generated motions could then be customised to reflect specific attributes of the exemplar motions such that the generated sequence is not only contextually relevant to the given text but also aligned with the unique characteristics of the references. This process is conceptually related to motion style transfer [AWL*20, HHKK17, PSK21] but surpasses some of its limitations. While style transfer methods generally focus on applying surface-level attributes to pre-existing motions, they are often limited in generating deterministic outputs [AWL*20], heavily depend on dataset characteristics [HHKK17, MZD*24] or require finetuning for each style, resulting in high memory requirements [HZY*24]. In contrast, textual inversion leverages large pre-trained motion models, optimises the text space for each new style and goes beyond mere 'style transfer' synthesising diverse motions that fully incorporate the essence of the references while adapting them to new contexts.

Nevertheless, to the best of our knowledge, no motion diffusion work demonstrates the ability to perform inversion or personalisation. We argue (and our experiments suggest) that stems from the misalignment between the latent space in which they operate and the latent space of a language model.

In this work, we propose LEAD, a new T2M model based on latent diffusion [RBL*22, CJL*23] that addresses the lack of semantic structure in the latent space. Key to our approach is the incorporation of a *projector* module that realigns the original latent space of a motion VAE towards one that is in more congruence with a language model, namely CLIP [RKH*21]. The proposed module is realised via an autoencoder that efficiently projects the diffused latents into the new space and re-projects them back into the VAE space at inference time (Figure 1). Furthermore, we explore the performance of the proposed model towards personalised motion generation. We introduce the concept of motion textual inversion (MTI), and observe that the proposed projector improves on this task in comparison to baseline methods. Similar to [GAA*22], we formulate motion inversion as the task of finding an embedding in the latent space of a pre-trained language model that best captures the characteristics of the exemplar motion. We evaluate our method on the task of T2M generation on the standard HumanML3D [GZZ*22] and KIT-ML [MTD*15] datasets. Our results show that LEAD achieves on-par performance with the state of the art in terms of motion quality while retaining good performance in terms of diversity and multimodality—a trade-off that none of the competing methods can handle well. LEAD shows quantitative and qualitative improvements for personalised motion generation over a vanilla implementation of textual inversion on existing models.

Our contributions can be summarised as follows: (a) We propose LEAD, a motion diffusion method relying on a text-motion realignment mechanism. (b) We show quantitatively and qualitatively that LEAD results in improved performance in two datasets for T2M generation, without additional computational cost. (c) We introduce the concept of MTI and explore how the simple yet elegant realignment in LEAD can help generate motions with specific characteristics from text.

2. Related Work

Human motion synthesis can be split into unconditional and conditional. Unconditional synthesis [YLX*19, ZBT20, ZSJ20] models the entire manifold of possible motions, while conditional synthesis introduces constraints such as audio or text that guide the generation. For a complete overview of motion synthesis, we refer the reader to [MHLC*21, KAK*22]. Here, we focus on conditional synthesis using multimodal constraints.

Multimodal Motion Synthesis. To condition motion generation, research works have explored the use of text [PBV21, PBV22, GZW*20, GZZ*22], images [GWE*20, RBH*21], audio [GBK*19], music [TCL23, AYA*21, LYL*19] and scenes [HCV*21, SZZK21]. Generating motion from text is

an intuitive way to produce 3D content and has received significant attention. Initial works targeted the action-to-motion task [PBV21, GZW*20, LBWR22] that produces motions depicting a single action. [GZW*20] propose a temporal-VAE based on GRUs to produce diverse motions, with a disentangled representation to better capture the kinematic properties. [PBV21] designs a transformer-based VAE with learnable tokens for each action, while [LBWR22] compresses motion into a discrete latent space and realise future states as next-index predictions.

Instead of relying on a fixed set of action categories, several works incorporate a text-encoder that transforms natural language into a latent space that acts as the conditioning signal [ZZC*23, GZWC22, GZZ*22, BRB*21, JCL*24]. Guo *et al.* [GZZ*22] first learn motion codes using a motion autoencoder and then use a recurrent VAE to map the text condition to a motion snippet code sequence. Similarly, Zhang *et al.* [ZZC*23] learn a mapping from motion to discrete codes using a VQ-VAE [vdOVK17], and use a transformer module to generate motion indices using text. MotionGPT [JCL*24] extends this idea to a uniform, versatile framework designed to address a variety of motion-language related tasks. Some T2M works align the motion-text spaces directly [AM19, PBV21, PBV22, GCO*21, APBV22]. JL2P [AM19] learns a joint pose-language space with a cross-modal loss, while [GCO*21] follows a similar approach but constructs two separate manifolds for the upper and lower body. TEMOS [PBV22] bypasses the need for two manifolds and ensures diverse sampling by encoding distribution parameters using a VAE and a pre-trained language model. To enable sequential generation of motions, TEACH [APBV22] augments the text-encoder branch of TEMOS to account for temporal compositions.

Recent works leverage the representational power of large language models. SINC [APBV23] enhances TEMOS [PBV21] by exploiting GPT-3 [BMR*20] to distil knowledge on the correspondence between actions and body parts, allowing for spatial compositions from text. More related to our work, Tevet *et al.* [TGH*22] propose MotionCLIP, a model that leverages the power of CLIP by aligning the latent motion representation to the image and text representations of CLIP [RKH*21]. Importantly, MotionCLIP [TGH*22] is designed to produce a one-to-one mapping between language and motion, failing to properly reflect the inherent diversity and ambiguity of human action, where multiple distinct movements can be appropriate responses to the same verbal instruction. Additionally, although MotionCLIP excels in action-to-motion and motion generation using high-level text, it does not account for global displacement and shows inconsistencies for out-of-distribution (OOD) generation. In a similar fashion, Ao *et al.* [AZL23] demonstrate the power of LLMs and contrastive learning in creating a shared latent space between full-body gestures and speech semantics, which is used to improve the task of speech-to-gesture generation.

Motion Diffusion Models. Diffusion models have demonstrated unprecedented capabilities for text-to-image generation [HJA20, RBL*22]. Recently their potential has been explored for the T2M task [TRG*23, ZCP*22, KKC23, DHMGT22, ZGP*23, KPST23, WLWBL*23, XJZ*23]. Tevet *et al.* [TRG*23] introduced the transformer-based Motion Diffusion Model (MDM) and proposed geometric losses specifically designed for the motion domain. Geometric losses are applied in numerous subsequent studies in motion generation [ZDC*24, LCM*24]. MotionDiffuse [ZZC*23] uses a cross-modality linear transformer as the backbone to enable a soft control, thus increasing the diversity of generated motions. ReMoDiffuse [ZGP*23] augments MotionDiffuse [ZZC*23] with a retrieval mechanism that refines the denoising process. OmniControl [XJZ*23] introduces an additional spatial control signal for individual joints. Dabral *et al.* [DHMGT22] design a system based on 1D U-Net with a cross-modal transformer and multihead attention that can be conditioned on both textual and audio signals. More related to our work, inspired by image latent diffusion [RBL*22]. Chen *et al.* [CJL*23] propose MLD, a method that performs diffusion over the VAE motion latent to handle noisy data and decrease computational complexity. The approach is structured such that a VAE encoder–decoder is first trained to compress high-dimensional motion data into a lower-dimensional latent space, helping mitigate noise artefacts from the capturing process. Diffusion is then performed over this learnt representation rather than the raw motion space. This structure allows the diffusion process to operate on a more compact representation, leading to reduced computational cost and memory requirements. The separation of components enables independent optimisation: the encoder–decoder focuses on learning high-quality latent motion representations, while the diffusion model is dedicated to generating diverse motions based on text control. MotionLCM [DCW*24] takes this a step further, enabling faster inference through a motion latent consistency model. They first pre-train MLD and observe improved performance by replacing the CLIP [RKH*21] text encoder with T5 [RSR*20]. MotionLCM then distils information from the pre-trained MLD and uses a motion ControlNet in the latent motion space, offering enhanced spatial control.

In our work, we explicitly align the motion latent space with language and demonstrate how a simple yet elegant realignment step during inference can boost the realism in the generations without sacrificing multimodality and diversity. In comparison to other controls which achieve a similar degree of realism [XJZ*23], our method generates realistic and expressive motions purely from language, without the need for explicit spatial control.

Textual Inversion in Diffusion Models. Textual inversion is the process of inverting data from other modalities to the language latent space in order to introduce new concepts in a pre-trained model [GAA*22, GAA*23, DD22, WK24]. Gal *et al.* [GAA*22] formulate an optimisation task for image inversion with diffusion models, where visual reconstruction is used as guidance to find the token embedding that corresponds to the new concept. Daras *et al.* [DD22] extend this idea to enable multilevel resolution, with various levels of agreement to the given example concepts. This is achieved using the observation that the conditioning signal is dependent on the diffusion timestep. Instead of forming an optimisation problem, Gal *et al.* [GAA*23] reformulate the problem as a regression based on the observation that one can underfit a large set of concepts from a given domain and use weight offsets to inject novel concepts of the same domain. In this work, we take inspiration from Gal *et al.* [GAA*22] and propose *motion textual inversion*. We keep the initial formulation of using the reconstruction loss to learn the new embedding but apply it to a different realigned space. Our results suggest that this leads to improved realism and consistency with the example concepts.

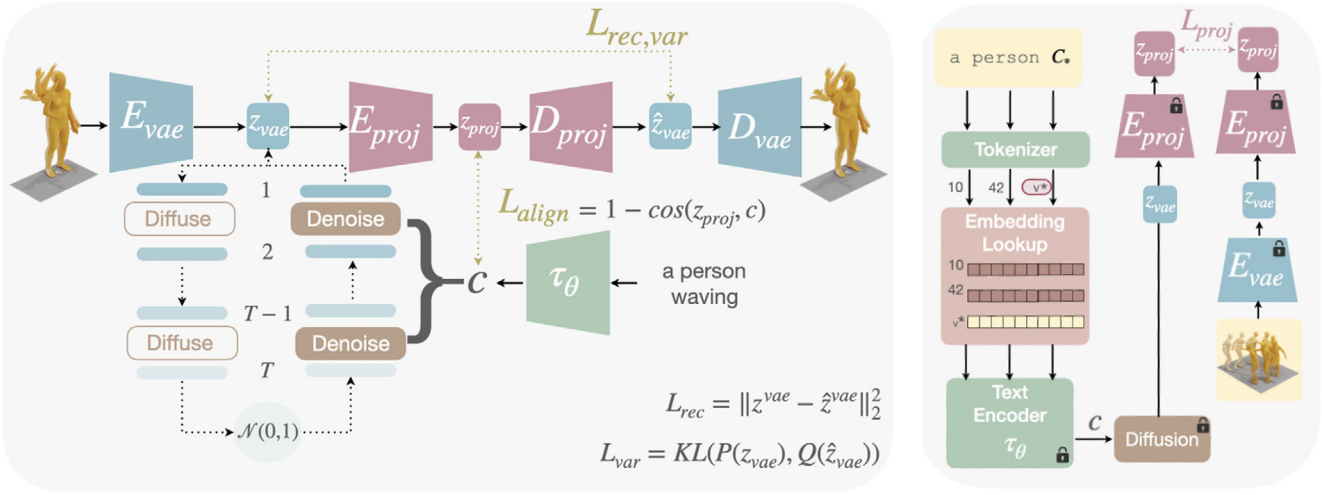


Figure 2: *LEAD. Left: Text-to-motion generation with LEAD. LEAD consists of four modules: (1) a motion VAE (blue), a text encoder (green), a diffusion model (brown) and our new projector module (pink). Similar to latent diffusion model (LDM) [RBL*22], we first train the VAE and then the diffusion model. We then train the projector module (pink) using an alignment loss towards the CLIP embedding, and a reconstruction loss towards the VAE embedding. Once all modules are trained, we generate a motion latent z^{vae} by sampling noise from the Gaussian distribution conditioned on the input text. The resulting latent is then auto-encoded by the projector and decoded through the VAE (blue) to obtain the final motion. Right: Motion textual inversion. A pseudo-word (C_*) is added as an additional token, and we seek the optimal embedding v_* to best reproduce the input. Text conditioning guides the generation of motion through the diffusion module (brown). The embedding of the new token is learnt using the reconstruction objective on the realigned space.*

3. Preliminaries

Diffusion models are a class of generative models based on progressively corrupting data $\mathbf{x}_0$, where the noising process $\{\mathbf{x}_t\}_{t=0}^T$ follows the Markovian principle:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)I), \quad (1)$$

with $\alpha_t \in (0, 1)$ hyperparameters that control the rate of diffusion at each timestep t. Note that when α_t is small, we can approximate $\mathbf{x}_t \sim \mathcal{N}(0, I)$. New samples can be generated by reversing the diffusion process, starting from a random vector $\mathbf{x}_T \sim \mathcal{N}(0, I)$ and predicting the next diffusion step iteratively. Here we follow the DDPM variant [HJA20] and train a neural network $\epsilon_\theta(\mathbf{x}_t, t)$ to predict the noise from step t to $(t-1)$ by minimizing

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{z}_0 \sim q(\mathbf{z}_0|c), t\sim[1,T]}||\epsilon - \epsilon_\theta(\mathbf{x}_t, t)||_2^2. \quad (2)$$

The model can additionally receive conditioning information c such as an action class or language embedding, that is, $\epsilon_{t-1}{=}\epsilon_\theta(\mathbf{x}_t, t, c)$.

An effective variant is the latent diffusion model [RBL*22, CJL*23], where diffusion is performed on the latent space of a pre-trained VAE instead of the feature space.

4. Method

Our goal is to generate a motion sequence given a sentence in natural language. For this, we introduce 'Latent rEAlignment for human motion Diffusion' (LEAD). We build on MLD [CJL*23] and perform diffusion over the latent space of a previously trained motion VAE. Different from other approaches, LEAD includes a specialised *projector* module trained to produce embeddings aligned with CLIP [RKH*21], transforming the diffused latent into a better-structured semantic space (Section 4.1). Additionally, in Section 4.2 we introduce the task of *motion textual inversion*, where we optimise for the textual embedding given a set of example motions. As will be shown in Section 5, using the realigned space allows the optimisation to better capture the input motion, showing the potential of our approach for personalised downstream tasks.

4.1. LEAD

Given a sentence y, the goal of LEAD is to generate a motion sequence $\mathbf{x} \in \mathbb{R}^{N\times D}$, where N is the motion length and D the dimension of motion features, including joint rotations, positions, velocities and foot contacts as in [GZZ*22].

4.1.1. *Architecture*

LEAD consists of four modules (Figure 2).

(1) A **motion VAE** [KW13] (blue in Figure 2), with an encoder $z_0^{vae} = E_{vae}(\mathbf{x})$, where $z_0^{vae} \in \mathbb{R}^M$ is a compressed representation of the motion segment, and a decoder $\tilde{\mathbf{x}} = D_{vae}(z_0^{vae})$ that transforms the latent back into a motion sequence. Following [CJL*23], the VAE is a transformer-based architecture [VSP*17] with long skip connections [RFB15]. The encoder E_{vae} takes as input two learnable distribution tokens corresponding to μ and σ of a Gaussian distribution, along with the motion features. The VAE decoder D_{vae} takes as input the latent z^{vae} and zero motion tokens and generates the motion sequence via cross-attention.

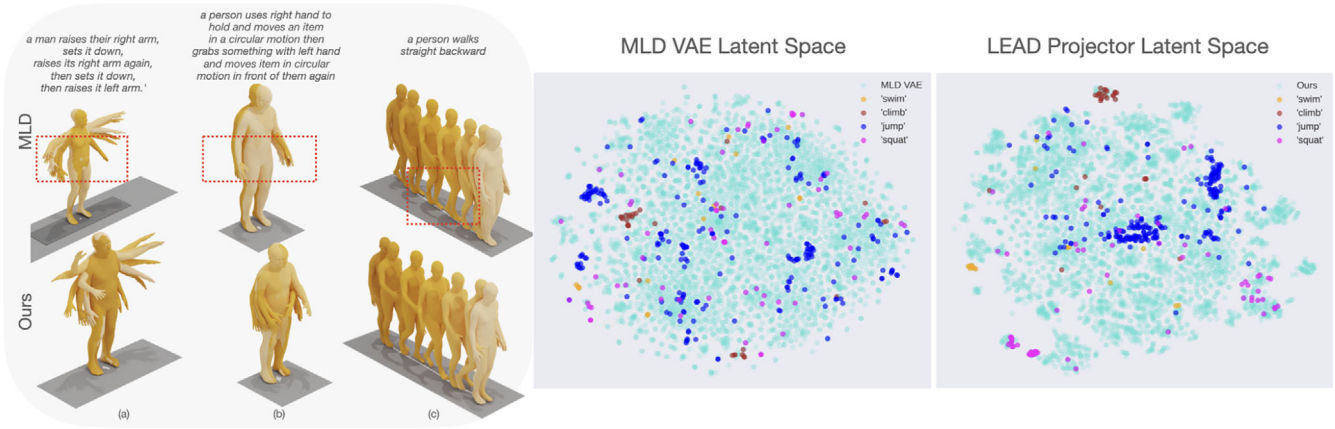


Figure 3: *Left: Qualitative results for T2M compared to the baseline model MLD [CJL*23]. Motions generated with our approach are more expressive and less static (a,b), and contain fewer artefacts like foot-sliding (c). Right: Latent space visualisation using tSNE [vdMH08].*

(2) A **conditional diffusion model** (brown in Figure 2) over the latent space, $\epsilon_\theta(z_t^{vae}, t, c)$, inspired by motion-latent-diffusion (MLD) [CJL*23], that predicts noise given the current noised version of the latent z_t^{vae}, a timestep t and a conditioning vector c (text embedding). We employ here a transformer with long skip connections [RFB15].

(3) A **text encoder** for the condition, τ_θ (green in Figure 2), which converts the input sentence into a latent embedding. In our work, τ_θ follows the CLIP text-encoder [RKH*21].

(4) The **projector module** $\mathcal{P}$=(E_{proj}, D_{proj}) (pink in Figure 2), consisting of the transfomer-based encoder E_{proj} and decoder D_{proj} with sinusoidal positional encodings.

The encoder transforms z^{vae} into a new embedding $z^{proj} = E_{proj}(z^{vae})$, while the decoder transforms back into the VAE space, $z^{\hat{vae}} = D_{proj}(z^{proj})$ from which the motion can be recovered, $\hat{\mathbf{x}} = D_{vae}(z^{\hat{vae}})$. Crucially, E_{proj} is trained to produce embeddings aligned with CLIP [RKH*21], such that they display a better semantic structure as shown in Figure 3.

While text-to-image models are capable of producing high-quality samples, their performance is more limited in T2M, due to the smaller scale of datasets as well as the large gap between motion and language. The explicit realignment module aims to address these shortcomings by using multimodal cues, without requiring extra motion data.

4.1.2. *Training and losses*

We train LEAD in three stages, where the first two steps follow previous works to train the latent diffusion model [CJL*23]. First, we train the motion VAE (E_{vae}, D_{vae}) over a large, unlabelled dataset consisting of motion data only. The VAE is trained using the mean squared error (MSE) and Kullback–Leibler divergence (KL) losses. The MSE loss on the motion features acts as a geometric loss, ensuring that the motion latents retain geometric properties, which are propagated to the diffusion process.

Second, we train the diffusion model ϵ_θ. For this step, we freeze the VAE and the text-encoder τ_θ. We use classifier-free guidance [HS22], to encourage a trade-off between quality and diversity. This is done by applying 10% dropout on the condition during training to learn both the conditioned and unconditioned distribution. The diffusion model is trained using Equation (2).

Finally, we freeze the weights of ϵ_θ, τ_θ and the VAE (E_{vae}, D_{vae}) and train only the realignment module $\mathcal{P} = (E_{proj}, D_{proj})$, using multimodal information from motion and language. For this, we first project the motion sequence $\mathbf{x}$ into the VAE latent space, $z^{vae} = E_{vae}(\mathbf{x})$. We next auto-encode z^{vae} using $\mathcal{P}$:

$$z^{proj} = E_{proj}(z^{vae}), \quad z^{\hat{vae}} = D_{proj}(z^{proj}). \tag{3}$$

In this third step, E_{proj} and D_{proj}, are jointly trained using the following losses:

(a) An alignment loss $\mathcal{L}_{align}$ that measures the similarity between the projected latent z_{proj} and the corresponding CLIP text latent c using cosine similarity:

$$\mathcal{L}_{align} = 1 - \cos(z_{proj}, c). \tag{4}$$

(b) A reconstruction loss $\mathcal{L}_{rec}$ that ensures we can recover the original VAE latent given the projected latent:

$$\mathcal{L}_{rec} = ||z^{vae} - z^{\hat{vae}}||_2^2. \tag{5}$$

(c) A variance loss $\mathcal{L}_{var}$ that ensures that the distribution of latents within a generated batch follows the distribution of latents within the training batch:

$$\mathcal{L}_{var} = KL(P(z_{vae}), Q(\hat{z}_{vae})), \tag{6}$$

where P, Q represent the probability distributions of ground-truth and predicted VAE latents. As shown in Table 1, this ensures that we retain the ability to generate diverse motions. The overall loss is:

$$\mathcal{L} = \lambda_{align}\mathcal{L}_{align} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{var}\mathcal{L}_{var}. \tag{7}$$

where λ_{align}, λ_{rec} and λ_{var} weigh the corresponding terms.

Table 1: *Ablation on projector module architecture and training losses.*

	R-prec. (top3) (↑)	FID (↓)	MMdist (↓)	Div (→)	MMod (↑)
Real	0.797	0.002	2.974	9.503	–
LEAD (mlp)	0.757	0.180	3.233	9.648	2.632
LEAD (skip.)	0.758	0.249	3.242	9.800	2.574
LEAD- w/o KL	**0.766**	0.320	3.199	9.799	2.489
LEAD- w/o Align	0.759	0.156	3.207	**9.509**	2.582
LEAD- w/o Rec	0.727	0.630	3.406	8.888	**2.749**
LEAD	0.762	**0.137**	**3.198**	9.656	2.572

Note: Top: transformer with long skip connections (skip) and a feedforwards network (mlp). Bottom: we remove each of proposed losses to highlight their contribution.

4.1.3. *Inference*

As shown in Figure 2, following a standard reverse diffusion process, we sample a latent noise vector $z_T^{vae} \sim \mathcal{N}(0, I)$ and gradually denoise it using $\epsilon_\theta(z_t^{vae}, t, c)$ with the CLIP condition c by relying on classifier-free guidance as in [CJL*23]:

$$\epsilon_\theta^s(z^{vae}, t, c) = s(\epsilon_\theta(z^{vae}, t, c)) + (1 - s)\epsilon_\theta(z^{vae}, t, \emptyset) \quad (8)$$

where s denotes the guidance scale. Once obtained the clean latent z_0^{vae}, we pass it through the projector $\mathcal{P}$ and recover the output motion through $\mathbf{x} = D_{vae}(D_{proj}(z^{proj}))$.

4.2. Motion textual inversion

Latent motion diffusion models allow us to introduce task of motion textual inversion (MTI), where given a few examples of a motion, the goal is to find the corresponding embedding in the language space such that it can later be used to generate action sequences that retain the exemplar's characteristics.

Following Gal *et al.* [GAA*22], we assume that the concept to be learned can be captured using a single word (C_*). Therefore, given a motion $m \in \mathbb{R}^{N \times D}$, we interpret the problem as seeking the optimal word embedding v^* that best represents the concept in the latent space of a pre-trained text encoder, that is, CLIP. Similar to [GAA*22], the weights of the text encoder τ_θ, the VAE (E_{vae}, D_{vae}) and the diffusion model ϵ_θ are frozen and only v_* is modified. The motion concept depicted by the examples is represented through a new place-holder word C_*, along with its corresponding optimal v_*. The approach for MTI is illustrated in Figure 2.

While image textual inversion [GAA*22] optimises the MSE between reference and generated images, our initial experiments showed that a simple reconstruction loss (either on the motion features $\mathbf{x}$, or on the VAE latent z^{vae}) does not yield satisfactory results (see Section 5.6). We hypothesise that the proposed text-motion realignment mechanism can provide more informative gradients during the token optimisation, leading to improved qualitative and quantitative performance. Based on our assumption, we propose minimising the following loss function:

$$\left\| E_{proj}(F(\epsilon)) - E_{proj}(F(\epsilon_\theta(z_t, t, \tau_\theta(y')))) \right\|_2^2 \quad (9)$$

where F denotes the conversion from a noised motion sample to a clean one with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, following [HJA20]:

$$x_0 \approx \hat{x}_0 = F(\epsilon) = (x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(x_t))/\sqrt{\bar{\alpha}_t} \quad (10)$$

Template texts When learning the pseudo-word of the new concept, we use predefined template texts to guide the diffusion model. We define the template texts using examples from the HumanML3D dataset. Below we provide some examples of template texts:

- the sim <*>.
- the man/woman <*>.
- figure <*>.
- someone <*>.
- he/she <*>.
- a robot <*>.

5. Experiments

In this section, we demonstrate the results of our method. First, we introduce the datasets and evaluation metrics (Section 5.1) as well as implementation details (Section 5.2). Next, we evaluate LEAD on the task of T2M generation (Section 5.3), and the newly proposed task of MTI (Section 5.5). Finally, in Section 5.6 we ablate our design choices for both tasks. More qualitative results can be found in the Supplementary Material.

5.1. Datasets and metrics

5.1.1. *Datasets*

For T2M we experiment on two standard datasets: HumanML3D [GZZ*22] and KIT-ML [MTD*15]. **HumanML3D** contains 14 616 human pose sequences with 44 970 descriptions, while **KIT-ML** contains 6353 textual descriptions for 3911 motions. Poses in both datasets are represented using the parameterisation from [GZZ*22] consisting of the root angular velocity along the *Y*-axis, the root linear velocities on the *XZ*-plane, the root height, the root local joint positions, velocities and rotations in root space and binary features for foot-ground contact.

For MTI, we perform all our quantitative comparisons on a randomly chosen subset of 100 motions from HumanML3D. This is due to the fact that we first need to optimise one placeholder token for each motion, which requires significant time. In addition, we qualitatively evaluate using the **100styles** dataset [MSZ*18], which contains a wide range of locomotion in signature styles. In particular, this dataset contains out-of-distribution movements which are not used when training LEAD. We illustrate the out-of-distribution scenario in Figure 7. To accommodate the differences in the skeletal structure between 100styles dataset to our framework (HumanML3D), we preprocess the dataset and obtain the representation as proposed in [GZZ*22].

Motion Representation We adopt the HumanML3D representation [GZZ*22], where each motion $\mathbf{x} \in \mathbb{R}^{N \times D}$ is expressed redundantly as $\mathbf{x} = \{\dot{r}_a, \dot{r}_x, \dot{r}_z, r_y, \mathbf{j}_p, \mathbf{j}_v, \mathbf{j}_r, \mathbf{c}_f\}$ that consists of:

- root angular velocity $\dot{r}_a \in \mathbb{R}$ along the *Y*-axis,

- root linear velocity $\dot{r}_x, \dot{r}_z \in \mathbb{R}$ along the *XZ*-axis,
- root height $r_y \in \mathbb{R}$,
- local joint positions $\mathbf{j}_p \in \mathbb{R}^{3N_j}$,
- local joint velocities $\mathbf{j}_v \in \mathbb{R}^{3N_j}$,
- local joint rotations $\mathbf{j}_r \in \mathbb{R}^{6N_j}$ in the root space,
- binary foot contact labels $\mathbf{c}_f \in \mathbb{R}^4$ that are obtained by thresholding on the velocities of the heel and toe joints.

Here, N_j denotes the number of joints ($N_J = 22$ for the HumanML3D dataset and $N_j = 21$ for KIT-ML).

5.1.2. *Evaluation metrics*

For T2M we follow the standard evaluation protocol with metrics from [GZW*20, GZZ*22] that measure four components: motion realism, text-motion consistency, generation diversity and multimodal matching between text and motion. To assess realism we compute the Fréchet Inception Distance (**FID**) [HRU*17] between the ground-truth and predicted motion features, that is,

$$\text{FID} = || \mu_r - \mu_g ||^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}), \quad (11)$$

where μ_r, Σ_r and μ_g, Σ_g correspond to the mean and covariance matrix of the real and generated motions respectively.

To measure text-motion consistency we report the multimodal distance (MMdist) and R-precision, where MMdist measures the average Euclidean distance between the generated motions and the corresponding conditioning texts, and R-precision measures the average retrieval accuracy for the top 1/2/3 matches. We compute **MMdist** using

$$\text{MMdist} = \frac{1}{N} \sqrt{(\sum_{i=1}^{N} \|v_i - t_i\|)} \quad (12)$$

where v_i are the features of the motions generated using texts y_i, and t_i are the features of the corresponding texts, y_i. For **R-precision**, following Guo *et al.* [GZZ*22], for each generated motion we select its ground-truth textual description along with 31 randomly selected mismatched descriptions. Then we calculate the Euclidean distances between the motion feature and the 32 text features and rank them based on the distance. We select the top-1, top-2 and top-3 and consider as successful retrieval when the corresponding real text falls into the top-k candidates. R-precision is computed as the average across all generated motions from the test set.

To evaluate generation diversity (**Div**) we compute the variance of synthesised motions across all categories. For a subset of size p the diversity can be computed as:

$$\text{DIV} = \frac{1}{p} \sum_{i=1}^{p} || v_i - \hat{v}_i || \quad (13)$$

where v_i and $\hat{v}_i$ are the features corresponding to the first and second subset of predictions respectively. Following previous work [CJL*23], we set $p = 300$.

Finally, multimodality (MModality) measures the variance across 100 motions generated using the same textual description. To calculate **MModality** (multimodality) we sample m textual descriptions. For each description we generate two subsets of motions of size d. Then MModality is computed as:

$$\text{MModality} = \frac{1}{m \times d} \sum_{j=1}^{m} \sum_{i=1}^{d} || v_{j,i} - \hat{v}_{j,i} || \quad (14)$$

where v and $\hat{v}$ are the motion features of each subset. Following [CJL*23] we set $m = 100$ and $d = 10$.

The metrics are computed on the features extracted by a pretrained motion feature extractor [GZZ*22].

5.2. Implementation details

For the text encoder we employ CLIP-ViT-14 [RKH*21]. The projector module is trained for 200 epochs with AdamW optimiser, which lasts approximately 2 h on an NVIDIA GeForce 3090. While training the realignment module we keep the text encoder and the VAE frozen. We use a learning rate of $1e^{-4}$, batch size 1024 with optimal hyperparameters $\lambda_{rec} = 4.7036$, $\lambda_{align} = 2.1591$ and $\lambda_{var} = 0.05960$ (Equation 7).

For T2M, we retain the hyperparameters from MLD [CJL*23], with 1K diffusion steps and 50 for training and inference respectively. For MTI, the pseudo-word embedding is initialised using a coarse one-word descriptor. We optimise for 20 steps using ADAM optimiser and batch of size 4, consisting of motion segments derived from one motion sequence.

The projector is realised as a transformer-based encoder/decoder. The encoder and decoder both consist of 9 layers with 4 heads and hidden size 1024. We use 0.1 dropout and the GeLu activation function [HG16]. We use sinusoidal positional encodings.

5.3. Text-to-motion generation

For T2M, we compare against the SOTA on the HumanML3D and KIT-ML datasets. We report the average statistics and 95% confidence interval across 20 runs.

Table 2 reports the results for HumanML3D and Table 3 for KIT-ML. Notably, on both datasets, LEAD achieves a significant improvement in motion realism compared to the other methods, as reflected by the FID. This includes the reference latent diffusion methods (MLD [TRG*23], MotionLCM [DCW*24]) which differ only in the use of the projector, showing that a simple realignment step during inference can substantially boost realism. Overall, we observe that the realignment mechanism leads to significant improvements in motion realism over the baselines (MLD, MotionLCM). Even though it demonstrates slightly worse performance compared to the state-of-the-art ReMoDiffuse, which relies on a retrieval mechanism at inference, our method achieves a better trade-off between realism and alignment as reflected in SOTA performance (R-precision, MMdist).

Additionally, we visualise using tSNE [vdMH08] the VAE latent space in comparison to the realigned space. As seen in Figure 3 (right), similar motions are better clustered in the projected space, suggesting that a semantic structure of the latent space might have an effect on the generation quality.

Table 2: *T2M on HumanML3D.*

			R-precision (↑)					
	LDM	FID (↓)	top1	top2	top3	MMdist (↓)	Diversity (→)	MModality (↑)
Real		$0.002^{\pm.008}$	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	–
JL2P [AM19]		$11.02^{\pm.046}$	$0.246^{\pm.001}$	$0.387^{\pm.002}$	$0.486^{\pm.002}$	$5.296^{\pm.008}$	$7.676^{\pm.058}$	–
T2G [BRB*21]		$7.664^{\pm.030}$	$0.165^{\pm.001}$	$0.267^{\pm.002}$	$0.345^{\pm.002}$	$6.030^{\pm.008}$	$6.409^{\pm.071}$	–
TEMOS [PBV22]		$3.734^{\pm.028}$	$0.424^{\pm.002}$	$0.612^{\pm.002}$	$0.722^{\pm.002}$	$3.703^{\pm.008}$	$8.973^{\pm.071}$	$0.368^{\pm.018}$
T2M [GZZ*22]		$1.067^{\pm.002}$	$0.457^{\pm.002}$	$0.639^{\pm.003}$	$0.740^{\pm.003}$	$3.340^{\pm.008}$	$9.188^{\pm.002}$	$2.090^{\pm.083}$
MDM [TRG*23]		$0.544^{\pm.044}$	$0.320^{\pm.005}$	$0.498^{\pm.004}$	$0.611^{\pm.007}$	$5.566^{\pm.027}$	$9.559^{\pm.086}$	$2.799^{\pm0.072}$
MotionDiffuse [ZCP*22]		$0.630^{\pm.001}$	$0.491^{\pm.001}$	$0.681^{\pm.001}$	$0.782^{\pm.001}$	$3.113^{\pm.001}$	$9.410^{\pm.049}$	$1.553^{\pm.042}$
TM2T [GZWC22]		$1.501^{\pm.017}$	$0.424^{\pm.003}$	$0.618^{\pm.001}$	$0.729^{\pm.002}$	$3.467^{\pm.011}$	$8.589^{\pm.076}$	$2.424^{\pm.093}$
T2M-GPT [ZZC*23]		$0.116^{\pm.004}$	$0.491^{\pm.003}$	$0.680^{\pm.003}$	$0.775^{\pm.002}$	$3.118^{\pm.011}$	$9.761^{\pm.081}$	$1.856^{\pm.011}$
GMD [KPST23]		$0.212^{\pm.000}$	–	–	$0.670^{\pm.000}$	–	$9.440^{\pm.000}$	–
Fg-T2M [WLWBL*23]		$0.243^{\pm.019}$	$0.492^{\pm.002}$	$0.683^{\pm.003}$	$0.783^{\pm.002}$	$3.109^{\pm.007}$	$9.278^{\pm.072}$	$1.614^{\pm.049}$
ReMoDiffuse [ZGP*23]		$\mathbf{0.103}^{\pm.004}$	$0.510^{\pm.005}$	$0.698^{\pm.006}$	$0.795^{\pm.004}$	$\mathbf{2.794}^{\pm.016}$	$9.018^{\pm.075}$	$1.796^{\pm.043}$
OmniControl [XJZ*23]		0.310	–	–	0.693	–	**9.502**	–
MotionGPT [JCL*24]		$0.232^{\pm.008}$	$0.492^{\pm.003}$	$0.681^{\pm.003}$	$0.778^{\pm.002}$	$3.096^{\pm.008}$	$9.528^{\pm.071}$	$2.008^{\pm.084}$
MotionLCM [DCW*24]	✓	$0.467^{\pm.012}$	$0.502^{\pm.003}$	$0.701^{\pm.002}$	$0.803^{\pm.002}$	$3.022^{\pm.009}$	$9.631^{\pm.066}$	$2.172^{\pm.082}$
LEAD (MotionLCM) (Ours)	✓	$0.296^{\pm.007}$	$\mathbf{0.522}^{\pm.003}$	$\mathbf{0.721}^{\pm.003}$	$\mathbf{0.818}^{\pm.003}$	$2.870^{\pm.010}$	$9.820^{\pm.074}$	$1.974^{\pm.054}$
MLD [CJL*23]	✓	$0.473^{\pm.013}$	$0.481^{\pm.003}$	$0.673^{\pm.003}$	$0.772^{\pm.002}$	$3.196^{\pm.010}$	$9.724^{\pm.082}$	$2.413^{\pm.079}$
LEAD (MLD) (Ours)	✓	$0.109^{\pm.005}$	$0.464^{\pm.003}$	$0.649^{\pm.004}$	$0.743^{\pm.003}$	$3.324^{\pm.015}$	$9.627^{\pm.146}$	$\mathbf{2.902}^{\pm.015}$

Note: Numbers are taken from previous works. Bold and underline denote first and second best, respectively. The second column indicates which models follow the latent diffusion (LDM) architecture. LEAD is designed for latent diffusion models (LDM), as they enable the downstream application of MTI. By incorporating LEAD, performance is further improved in LDM models.

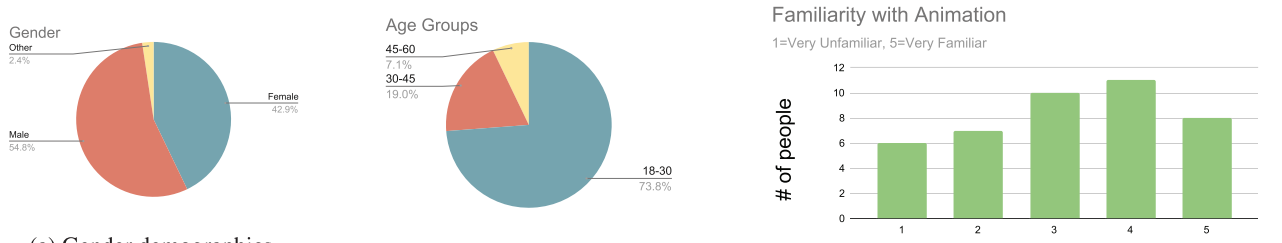
Table 3: *T2M on the KIT-ML dataset.*

			R-precision (↑)					
	LDM	FID (↓)	top1	top2	top3	MMdist (↓)	Diversity (→)	MModality (↑)
Real		$0.031^{\pm.004}$	$0.424^{\pm.005}$	$0.649^{\pm.006}$	$0.779^{\pm.006}$	$2.788^{\pm.012}$	$11.08^{\pm.097}$	–
LJ2P [AM19]		$6.545^{\pm.072}$	$0.221^{\pm.005}$	$0.373^{\pm.004}$	$0.483^{\pm.005}$	$5.147^{\pm.030}$	$9.073^{\pm.100}$	–
T2G [BRB*21]		$12.12^{\pm.183}$	$0.156^{\pm.004}$	$0.255^{\pm.004}$	$0.338^{\pm.005}$	$6.964^{\pm.029}$	$9.334^{\pm.079}$	–
TEMOS [PBV22]		$3.717^{\pm.051}$	$0.353^{\pm.006}$	$0.561^{\pm.007}$	$0.687^{\pm.005}$	$3.417^{\pm.019}$	$10.84^{\pm.100}$	$0.532^{\pm.034}$
T2M [GZZ*22]		$2.770^{\pm.109}$	$0.370^{\pm.005}$	$0.569^{\pm.007}$	$0.693^{\pm.007}$	$3.401^{\pm.008}$	$10.91^{\pm.119}$	$1.482^{\pm.065}$
MDM [TRG*23]		$0.497^{\pm.021}$	$0.164^{\pm.004}$	$0.291^{\pm.004}$	$0.396^{\pm.004}$	$9.191^{\pm.022}$	$10.85^{\pm.109}$	$1.907^{\pm.214}$
MotionDiffuse [ZCP*22]		$1.954^{\pm.062}$	$0.417^{\pm.004}$	$0.621^{\pm.004}$	$0.739^{\pm.004}$	$\mathbf{2.958}^{\pm.005}$	$11.10^{\pm.143}$	$0.730^{\pm.013}$
TM2T [GZWC22]		$3.599^{\pm.153}$	$0.280^{\pm.005}$	$0.463^{\pm.006}$	$0.587^{\pm.005}$	$4.591^{\pm.026}$	$9.473^{\pm.117}$	$\mathbf{3.292}^{\pm.081}$
T2M-GPT [ZZC*23]		$0.514^{\pm.029}$	$0.416^{\pm.006}$	$0.627^{\pm.006}$	$0.745^{\pm.006}$	$3.007^{\pm.023}$	$10.921^{\pm.108}$	$1.570^{\pm.039}$
Fg-T2M [WLWBL*23]		$0.571^{\pm.047}$	$0.418^{\pm.005}$	$0.626^{\pm.004}$	$0.745^{\pm.004}$	$3.114^{\pm.015}$	$10.93^{\pm.083}$	$1.019^{\pm.029}$
ReMoDiffuse [ZGP*23]		$\mathbf{0.155}^{\pm.006}$	$\mathbf{0.427}^{\pm.014}$	$\mathbf{0.641}^{\pm.004}$	$\mathbf{0.765}^{\pm.055}$	$2.814^{\pm.012}$	$10.80^{\pm.105}$	$1.239^{\pm.028}$
OmniControl [XJZ*23]		0.788	–	–	0.379	–	10.841	–
MotionGPT [JCL*24]		$0.510^{\pm.016}$	$0.366^{\pm.005}$	$0.558^{\pm.004}$	$0.680^{\pm.005}$	$3.527^{\pm.021}$	$10.350^{\pm.084}$	$2.328^{\pm.117}$
MLD [CJL*23]	✓	$0.404^{\pm.027}$	$0.390^{\pm.008}$	$0.609^{\pm.008}$	$0.734^{\pm.007}$	$3.204^{\pm.027}$	$10.800^{\pm.117}$	$2.192^{\pm.071}$
LEAD (MLD) (Ours)	✓	$0.246^{\pm.014}$	$0.388^{\pm.005}$	$0.608^{\pm.005}$	$0.732^{\pm.005}$	$3.199^{\pm.031}$	$\mathbf{11.099}^{\pm.080}$	$1.919^{\pm.060}$

Note: Numbers are taken from previous works. Bold and underline denote first and second best, respectively. The second column indicates which models follow the latent diffusion (LDM) architecture. LEAD is designed for latent diffusion models (LDM), as they enable the downstream application of MTI. By incorporating LEAD, performance is further improved in LDM models.

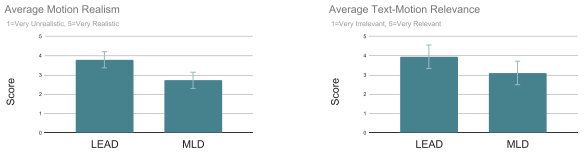
Finally, we show qualitative results in Figure 3. We observe that the synthesised motions from LEAD are more lifelike and expressive compared to the ones from MLD. In particular, Figure 3a,b show that motions generated with LEAD display richer movement that still complies with the text, while motions generated with MLD are smoother and inert, an observation that holds for more rare actions such as swimming. Furthermore, LEAD minimises motion artefacts such as foot-sliding, as can be seen in Figure 3c. We provide animated qualitative results and comparisons to baselines in the Supplementary Video. We compare T2M results of LEAD with those generated using T2M [GZZ*22], MDM [TRG*23] and our reference method, MLD [CJL*23].

User Study To assess the quality of our generated motions we conducted two user studies, where we compared LEAD against our reference methods MLD and MotionCLIP on two axes: motion



(a) Gender demographics. (b) Age demographics. (c) Familiarity of participants with computer animation.

Figure 4: *User study information of participants.*



(a) Realism (b) Text-Motion Relevance

	Realism (↑)	Text-Motion Relevance (↑)
MotionCLIP [TGH*22]	1.9	1.6
LEAD	**4.0**	**4.1**

(c) LEAD vs MotionCLIP user study results.

Figure 5: *User study results. We evaluate motion realism and text-motion relevance on a 1–5 scale where a higher score corresponds to better performance (1=very unrealistic/unrelated, 5=very realistic/related). Motions generated with LEAD are consistently perceived as more realistic and relevant than those generated using MLD [CJL*23] ([a] and [b]) and MotionCLIP [TGH*22] (c).*

realism and text-motion relevance. For this, we visualised 10 pairs of motions side-by-side for each method and asked participants to rate on a scale of 1–5 (1 = very unrealistic/unrelated, 5 = very realistic/related). We report our findings in Figure 5.

LEAD versus MLD: We first assessed motion realism (ignoring the text), where LEAD achieved 3.81 on average while MLD achieved 2.75, showing a significant outperformance when applying the projector module. Secondly, we showed the text description used to generate the motions and asked the users to rate how well the text and motion were matched. Here, LEAD scores 3.97 while MLD scores 3.12, showing that our generated motions are not only realistic but can also better match the input text. For the user study, we select motions generated using LEAD and MLD, conditioned on textual descriptions from HumanML3D [GZZ*22]. Our selection is based on the criterion of how much the motion differs before and after the realignment. We enlisted 40 participants with various age, gender and knowledge backgrounds (see Figure 4). Our results in terms of average motion realism and text-motion relevance can be seen in Figure 5, and clearly demonstrate a preference for our method.

LEAD versus MotionCLIP: Figure 5 shows the average results in a user study of 30 participants. On motion realism (ignoring the

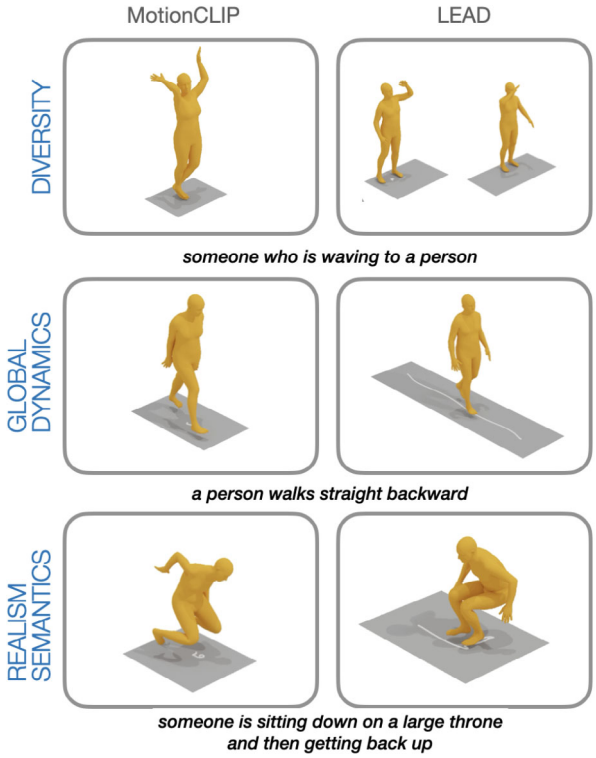


Figure 6: *Qualitative comparison between LEAD and MotionCLIP on 4 axes: diversity, global dynamics, realism and semantic alignment.*

text), LEAD significantly outperforms MotionCLIP achieving average scores of 4.0 and 1.9 respectively. In terms of text-motion alignment, LEAD scores 4.1 while MotionCLIP scores 1.6, showing that our generated motions are not only realistic but can also better match the input text. Qualitative results as seen in Figure 6 show that LEAD outperforms MotionCLIP in realism and semantic alignment, while additionally capturing motion diversity and generating global motion dynamics (translation, orientation).

Inference Time We provide a comparison in terms of inference times between LEAD and MLD [CJL*23]. We find that the performance benefits of LEAD come with marginally increased computational overhead compared to MLD [CJL*23]. Inference takes 0.236s per prompt for MLD and 0.245s for LEAD.

Table 4: *MotionLCM versus LEAD (MotionLCM) on the HumanML3D dataset.*

Steps	Model	FID (↓)	R-precision (↑) top1	top2	top3	MMdist (↓)	Diversity (→)	MModality (↑)
	Real	$0.002^{\pm.008}$	$0.511^{\pm.003}$	$0.703^{\pm.003}$	$0.797^{\pm.002}$	$2.974^{\pm.008}$	$9.503^{\pm.065}$	–
1	MotionLCM	$0.467^{\pm.012}$	$0.502^{\pm.003}$	$0.701^{\pm.002}$	$0.803^{\pm.002}$	$3.022^{\pm.009}$	$\mathbf{9.631}^{\pm.066}$	$\mathbf{2.172}^{\pm.082}$
	LEAD (MotionLCM)	$\mathbf{0.296}^{\pm.007}$	$\mathbf{0.522}^{\pm.003}$	$\mathbf{0.721}^{\pm.003}$	$\mathbf{0.818}^{\pm.003}$	$\mathbf{2.870}^{\pm.010}$	$9.820^{\pm.074}$	$1.974^{\pm.054}$
2	MotionLCM	$0.368^{\pm.011}$	$0.505^{\pm.003}$	$0.705^{\pm.002}$	$0.805^{\pm.002}$	$2.986^{\pm.008}$	$\mathbf{9.640}^{\pm.052}$	$\mathbf{2.187}^{\pm.094}$
	LEAD (MotionLCM)	$\mathbf{0.292}^{\pm.008}$	$\mathbf{0.518}^{\pm.003}$	$\mathbf{0.717}^{\pm.003}$	$\mathbf{0.814}^{\pm.002}$	$\mathbf{2.881}^{\pm.009}$	$9.753^{\pm.090}$	$1.985^{\pm.049}$
4	MotionLCM	$0.304^{\pm.012}$	$0.502^{\pm.003}$	$0.698^{\pm.002}$	$0.798^{\pm.002}$	$3.012^{\pm.007}$	$\mathbf{9.607}^{\pm.066}$	$\mathbf{2.259}^{\pm.092}$
	LEAD (MotionLCM)	$\mathbf{0.268}^{\pm.008}$	$\mathbf{0.509}^{\pm.003}$	$\mathbf{0.708}^{\pm.003}$	$\mathbf{0.806}^{\pm.003}$	$\mathbf{2.932}^{\pm.011}$	$9.682^{\pm.078}$	$2.110^{\pm.053}$

Note: Bold indicates the best performance.

5.4. Re-usability of LEAD latent space

We investigate the generalisation capacity of our proposed LEAD latent projector on different motion latent diffusion models.

In particular, we consider a different diffusion model for T2M generation such as MotionLCM [DCW*24]. MotionLCM offers speed enhancements by employing distillation, resulting in one-step (or few-step) inference. In Table 4, we report standard metrics to evaluate the performance of MotionLCM enhanced with the LEAD latent projector on the HumanML3D dataset. We observe that MotionLCM enhanced with the LEAD projector space improves in text-motion alignment (R-precision, MMdist) and motion quality (FID) consistently across different inference steps. We note here that we directly use the LEAD latent space **without retraining** for the MotionLCM model. Interestingly, while the LEAD latent space was trained using semantic language from CLIP, the learnt structure can be transferable to MotionLCM where the text is encoded using T5 [RSR*20] suggesting the robustness of the projector space.

5.5. Motion textual inversion

We explore the performance of the proposed MTI using LEAD (denoted MTI_{proj}) in comparison to the following baselines:

- MTI_{mld}: Same as in image textual inversion [GAA*22], we first consider MTI on MLD, that is, calculating the reconstruction in the VAE space, that is, $||\epsilon - \epsilon_\theta(z_t^{vae}, t, \tau_\theta(y')||_2^2$.
- MTI_{feat}: Here we optimise using the reconstruction loss directly on the decoded motion, with F defined in Equation (10): $\left\|D_{vae}(F(\epsilon)) - D_{vae}(F(\epsilon_\theta(z_t, t, \tau_\theta(y'))))\right\|_2^2$

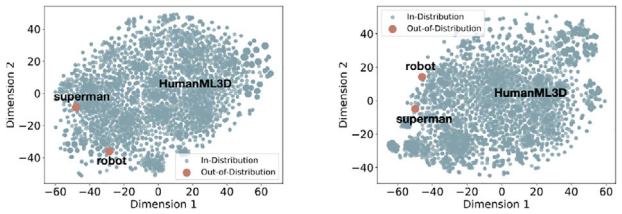
For all variants, we evaluate FID, R-precision, MMdist and diversity on a mini-subset of 100 motions from HumanML3D. The results, as shown in Table 5, demonstrate that MTI on the realigned space leads to improved results in terms of realism and multimodal alignment, compared to MTI on the MLD or motion feature space. Furthermore, we observe that incorporating the realignment mechanism during the T2M generation with the newly learnt token proves beneficial in all MTI variants.

Additionally, we qualitatively evaluate on the 100styles dataset [MSZ*18] as it contains out-of-distribution sequences, and compare ours (MTI_{proj}w/project) to the baseline MTI_{MLD}.

Table 5: *MTI evaluation on a mini-subset of HumanML3D.*

	Project	R-prec. (top3) (↑)	FID (↓)	MMdist (↓)	Div.(→)
Real	–	0.795	–	2.967	9.681
MTI_{feat}	–	0.447	5.929	5.360	8.296
MTI_{feat}	✓	0.473	4.754	5.244	8.396
MTI_{mld}	–	0.528	3.066	4.741	**8.874**
MTI_{mld}	✓	**0.537**	2.736	4.679	8.768
MTI_{lead}	–	0.521	3.042	4.760	8.741
MTI_{lead}	✓	0.533	**2.529**	**4.663**	8.741

Note: We show the results using the reconstruction objective on the VAE space of MLD (MTI_{mld}), the motion feature space (MTI_{feat}) and our realigned space (MTI_{lead}), with (w/project) and without autoencoding through the projector during generation.



(a) Text embedding space (b) Motion embedding space

Figure 7: *Visualisation of text (a) and motion (b) embedding spaces to highlight the out-of-distribution (OOD) scenario. We compute CLIP text embeddings and VAE motion embeddings and visualise them using t-SNE. The blue points represent samples from HumanML3D [GZZ*22] (training dataset, in-distribution), while the orange points correspond to examples from the 100Styles [MSZ*18] dataset (unseen during training, out-of-distribution).*

We select distinctive styles from 100styles such as *superman* and *airplane* which are considered OOD samples since they were not seen during training. In Figure 7 we visualise the text (CLIP embeddings) and motion spaces (VAE embeddings) and observe that styles such as 'superman' or 'robot' do not belong to dense regions of the training distribution (HumanML3D). We use MTI to learn a unique pseudo-word for each exemplar clip. The results can be found in Figure 8, where we see that our generated motions are more expressive and closer in content to the target examples.

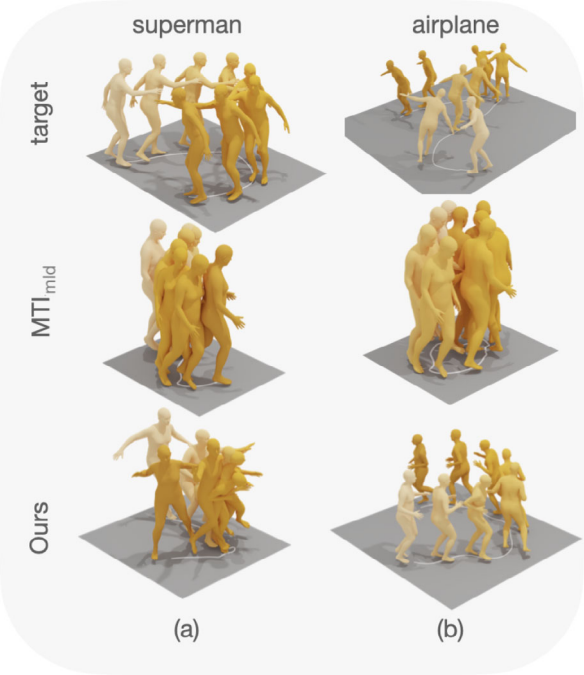


Figure 8: *Results for MTI on out-of-distribution data show that, compared to the baseline MTI on MLD, motions generated using textual inversion on LEAD are more expressive and better preserve the characteristics of the target examples.*

Table 6: *MTI ablation on mini-subset of HumanML3D.*

	Project	R-prec. (top3) (↑)	FID (↓)	MMdist (↓)	Div.(→)
Real	–	0.795	–	2.967	9.681
noALIGN	✓	**0.539**	2.762	**4.642**	8.729
noKL	✓	0.535	2.963	4.710	**8.945**
noREC	✓	0.526	3.537	4.725	8.373
PROJ	✓	0.533	**2.529**	4.663	8.741

Note: We report results with the realignment scheme during generation.

On the other hand, MTI$_{MLD}$ (that is, optimising directly towards the VAE latent) generates more inert motions, which illustrates the need for a realignment step during the inversion process. Animated results can be found in the supplementary video.

5.6. Ablations

Finally, we conduct ablation studies on the architecture design for T2M on HumanML3D (Table 1). In addition, we ablate the losses of LEAD for both T2M on HumanML3D (Table 1), as well as MTI on HumanML3D-mini (Table 6).

For T2M, we examine three different architectures (Table 1): a simple MLP (first row), a transformer with long skip connections as in the VAE architecture of [CJL*23] (second row), and a transformer-based architecture (the proposed LEAD). We observe that the best performance of LEAD consistently, across all metrics, is achieved when using a transformer-based architecture and all proposed losses (last row). Additionally, removing the reconstruction and KL loss leads to worse performance on FID and diversity while removing the alignment loss drops the R-precision.

For MTI, we examine the impact of the losses when using the projector without the KL loss (noKL), reconstruction loss (noREC) and alignment loss (noALIGN) and compare with the projector trained with all losses. Results in Table 6 show that the motion realism (FID) is significantly worse when the alignment loss is removed. We conclude that LEAD with all proposed losses achieves the best trade-off between R-precision, FID, MMdist and diversity.

6. Conclusion

We proposed LEAD, a human motion diffusion model equipped with a latent realignment mechanism. Leveraging the power of CLIP, we introduced a new latent space that is semantically better structured and can be trained with little effort given a pre-trained motion diffusion model.

Moreover, we proposed the new task of MTI, which optimises for the textual embedding that best explains a set of example motions. This enables the generation of actions that are hard to explain with natural language, laying the groundwork for a more personalised content creation process.

Our quantitative and qualitative results show improved performance on the T2M task on the HumanML3D and KIT-ML datasets, achieving greater realism without compromising diversity. Our evaluation of MTI on the 100styles dataset suggests that the realigned latent space may be more suitable for generating out-of-distribution sequences from a few examples. While the results are promising, further exploration is needed to fully assess its potential for personalised motion generation with limited exemplar motions.

Limitations As demonstrated quantitatively and reflected by the user study, compared to the baselines, motions generated with LEAD are more realistic. However, they still sometimes exhibit motion artefacts such as foot-sliding—especially for the MTI task. Furthermore, for long and sequential textual descriptions, LEAD may struggle to produce motions that faithfully comply with the motion description. Finally, even though LEAD can generate arbitrary-length motions due to its design, it remains bounded by the maximum motion length present in the dataset.

Social Impact Personalised motion generation from text introduces the risk of misuse in developing deceptive artificial content and raises ethical concerns regarding privacy and consent, as the technology may create motions that portray a specific person without explicit authorisation from them.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no 860768 (CLIPE project). Xi Wang was supported by ANR-22-CE39-0016 and a Hi!Paris fellowship, and was granted access to the High-Performance Com-

puting (HPC) resources of IDRIS under the allocation 2024-AD011014300R1 made by GENCI.

References

[AM19] Ahuja C., Morency L.-P.: Language2Pose: Natural language grounded pose forecasting. In *2019 International Conference on 3D Vision*. IEEE (2019), 719-728.

[APBV22] Athanasiou N., Petrovich M., Black M. J., Varol G.: TEACH: Temporal action compositions for 3D humans. In *2022 International Conference on 3D Vision (3DV)*. IEEE (2022), 414–423.

[APBV23] Athanasiou N., Petrovich M., Black M. J., Varol G.: SINC: Spatial composition of 3D human motions for simultaneous action generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE (2023), pp. 9950–9961.

[AWL*20] Aberman K., Weng Y., Lischinski D., Cohen-Or D., Chen B.: Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics 39*, 4 (2020).

[AYA*21] Aristidou A., Yiannakidis A., Aberman K., Cohen-Or D., Shamir A., Chrysanthou Y.: Rhythm is a dancer: Music driven motion synthesis with global structure. *IEEE Trans. on Visualization and Computer Graphics (TVCG) 29*, 08 (2021), 3519–3534.

[AZL23] Ao T., Zhang Z., Liu L.: GestureDiffuCLIP: Gesture diffusion model with CLIP latents. *ACM Transactions on Graphics 42*, 4 (2023), 1–18.

[BMR*20] Brown T., Mann B., Ryder N., Subbiah M., Kaplan J. D., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D., Wu J., Winter C., Hesse C., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner C., McCandlish S., Radford A., Sutskever I., Amodei D.: Language models are few-shot learners. In *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates, Inc. (2020), vol. *33*, pp. 1877–1901.

[BRB*21] Bhattacharya U., Rewkowski N., Banerjee A., Guhan P., Bera A., Manocha D.: Text2Gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE (2021), 1–10.

[CJL*23] Chen X., Jiang B., Liu W., Huang Z., Fu B., Chen T., Yu G.: Executing your commands via motion diffusion in latent space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE (2023), pp. 18000–18010.

[DCW*24] Dai W., Chen L.-H., Wang J., Liu J., Dai B., Tang Y.: MotionLCM: Real-time controllable motion generation via latent consistency model. In *Computer Vision–ECCV*. Springer-Verlag (2024).

[DD22] Daras G., Dimakis A. G.: Multiresolution textual inversion. *ArXiv PrePrint* (2022).

[DHMGT22] Dabral R., Hamza Mughal M., Golyanik V., Theobalt C.: MoFusion: A framework for denoising-diffusion-based motion synthesis. *Computer Vision and Pattern Recognition (CVPR)* (2023).

[GAA*22] Gal R., Alaluf Y., Atzmon Y., Patashnik O., Bermano A. H., Chechik G., Cohen-Or D.: An image is worth one word: Personalizing text-to-image generation using textual inversion. In *International Conference on Learning Representations*. ICLR (2022).

[GAA*23] Gal R., Arar M., Atzmon Y., Bermano A. H., Chechik G., Cohen-Or D.: Designing an encoder for fast personalization of text-to-image models. *ArXiv PrePrint* (2023).

[GBK*19] Ginosar S., Bar A., Kohavi G., Chan C., Owens A., Malik J.: Learning individual styles of conversational gesture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE (2019), pp. 3492–3501.

[GCO*21] Ghosh A., Cheema N., Oguz C., Theobalt C., Slusallek P.: Synthesis of compositional animations from textual descriptions. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE (2021), 1376–1386.

[GSAH17] Ghosh P., Song J., Aksan E., Hilliges O.: Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*. IEEE (2017), 458–466.

[GWE*20] Ghorbani S., Wloka C., Etemad A., Brubaker M. A., Troje N. F.: Probabilistic character motion synthesis using a hierarchical deep latent variable model. *Computer Graphics Forum (CGF) 39*, 8 (2020), 225–239.

[GZW*20] Guo C., Zuo X., Wang S., Zou S., Sun Q., Deng A., Gong M., Cheng L.: Action2Motion: Conditioned generation of 3D human motions. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*. ACM (2020), 2021-2029.

[GZWC22] Guo C., Zuo X., Wang S., Cheng L.: TM2T: Stochastic and tokenized modeling for the reciprocal generation of 3D human motions and texts. In *Computer Vision–ECCV*. Springer-Verlag (2022), pp. 580–597.

[GZZ*22] Guo C., Zou S., Zuo X., Wang S., Ji W., Li X., Cheng L.: Generating diverse and natural 3D human motions from text. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE (2022), pp. 5142–5151.

[HCV*21] Hassan M., Ceylan D., Villegas R., Saito J., Yang J., Zhou Y., Black M.: Stochastic scene-aware motion prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE (2021), pp. 11354–11364.

[HG16] Hendrycks D., Gimpel K.: Gaussian error linear units (GELUs). *ArXiv PrePrint* (2016).

[HHKK17] Holden D., Habibie I., Kusajima I., Komura T.: Fast neural style transfer for motion data. *IEEE Computer Graphics and Applications 37*, 4 (2017), 42–49.

[HJA20] Ho J., Jain A., Abbeel P.: Denoising diffusion probabilistic models. In *NIPS '20: Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc, Red Hook, NY, USA (2020).

[HKS17] Holden D., Komura T., Saito J.: Phase-functioned neural networks for character control. *ACM Transactions on Graphics 36* 4 (2017), 1–13 .

[HRU*17] Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates Inc. (2017), pp. 6629–6640.

[HS22] Ho J., Salimans T.: Classifier-free diffusion guidance. *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications* (2022).

[HSK16] Holden D., Saito J., Komura T.: A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics 35*, 4 (2016), 1–11.

[HZY*24] Hu L., Zhang Z., Ye Y., Xu Y., Xia S.: Diffusion-based human motion style transfer with semantic guidance. In *SCA '24: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Eurographics Association (2024), pp. 1–12.

[JCL*24] Jiang B., Chen X., Liu W., Yu J., Yu G., Chen T.: MotionGPT: Human motion as a foreign language. *Advances in Neural Information Processing Systems (NeurIPS) 36* (2024).

[KAK*22] Kwiatkowski A., Alvarado E., Kalogeiton V., Liu C. K., Pettré J., van de Panne M., Cani M.-P.: A survey on reinforcement learning methods in character animation. In *Computer Graphics Forum (CGF)*. John Wiley and Sons (2022), vol. *41*, pp. 613–639.

[KKC23] Kim J., Kim J., Choi S.: FLAME: Free-form language-based motion synthesis & editing. *Conference on Artificial Intelligence (AAAI)*. AAAI Press (2023), 8255–8263.

[KPST23] Karunratanakul K., Preechakul K., Suwajanakorn S., Tang S.: Guided motion diffusion for controllable human motion synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE (2023), pp. 2151–2162.

[KW13] Kingma D. P., Welling M.: Auto-encoding variational {*Bayes*}. In *Int. Conf. on Learning Representations* (2013).

[LBWR22] Lucas* T., Baradel* F., Weinzaepfel P., Rogez G.: PoseGPT: Quantization-based 3D human motion generation and forecasting. In *Computer Vision–ECCV*. Springer-Verlag (2022), pp. 417–435.

[LCM*24] Li J., Clegg A., Mottaghi R., Wu J., Puig X., Liu C. K.: Controllable human-object interaction synthesis. In *European Conference on Computer Vision*. Springer (2024), pp. 54–72.

[LYL*19] Lee H.-Y., Yang X., Liu M.-Y., Wang T.-C., Lu Y.-D., Yang M.-H., Kautz J.: Dancing to music. *Advances in Neural Information Processing Systems (NeurIPS) 32* (2019), 3586–3596.

[MHLC*21] Mourot L., Hoyet L., Le Clerc F., Schnitzler F., Hellier P.: A survey on deep learning for skeleton-based human animation. *Computer Graphics Forum (CGF) 41*, 1 (2021), 122–157.

[MSZ*18] Mason I., Starke S., Zhang H., Bilen H., Komura T.: Few-shot learning of homogeneous human locomotion styles. *Computer Graphics Forum (CGF) 37*, 7 (2018), 143–153.

[MTD*15] Mandery C., Terlemez O., Do M., Vahrenkamp N., Asfour T.: The kit whole-body human motion database. In *International Conference on Advanced Robotics*. IEEE (2015), 329–336.

[MZD*24] Mu Y., Zuo X., Dai P., Yan Y., Lu J., Cheng L., et al.: Generative human motion stylization in latent space. In *Intl. Conf. on Learning Representations* (2024).

[PBV21] Petrovich M., Black M. J., Varol G.: Action-conditioned 3D human motion synthesis with transformer VAE. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE (2021), pp. 10965–10975.

[PBV22] Petrovich M., Black M. J., Varol G.: TEMOS: Generating diverse human motions from textual descriptions. In *Computer Vision–ECCV*. Springer-Verlag (2022), pp. 480–497.

[PMA18] Plappert M., Mandery C., Asfour T.: Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks. *Robotics and Autonomous Systems* (2018).

[PSK21] Pan J., Sun H., Kong Y.: Fast human motion transfer based on a meta network. *Journal of Information Sciences 547* (2021), 367–383.

[RBH*21] Rempe D., Birdal T., Hertzmann A., Yang J., Sridhar S., Guibas L. J.: HuMoR: 3D human motion model for robust pose estimation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE (2021), pp. 11468–11479.

[RBL*22] Rombach R., Blattmann A., Lorenz D., Esser P., Ommer B.: High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE (2022), pp. 10674–10685.

[RFB15] Ronneberger O., Fischer P., Brox T.: U-net: Convolutional networks for biomedical image segmentation. In *MICCAI* (2015).

[RKH*21] Radford A., Kim J. W., Hallacy C., Ramesh A., Goh G., Agarwal S., Sastry G., Askell A., Mishkin P., Clark J., et al.: Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PLMR (2021), vol. *139*, pp. 8748–8763.

[RSR*20] Raffel C., Shazeer N., Roberts A., Lee K., Narang S., Matena M., Zhou Y., Li W., Liu P. J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research 21*, 140 (2020), 1–67.

[SZZK21] Starke S., Zhao Y., Zinno F., Komura T.: Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics 40*, 4 (2021), 1–16.

[TCL23] Tseng J., Castellon R., Liu K.: EDGE: Editable dance generation from music. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE (2023), pp. 448–458.

[TGH*22] Tevet G., Gordon B., Hertz A., Bermano A. H., Cohen-Or D.: MotionCLIP: Exposing human motion generation to CLIP space. In *Computer Vision–ECCV*. Springer-Verlag (2022), pp. 358–374.

[TRG*23] Tevet G., Raab S., Gordon B., Shafir Y., Bermano A. H., Cohen-Or D.: Human motion diffusion model. In *Intl. Conf. on Learning Representations* (2023).

[vdMH08] van der Maaten L., Hinton G.: Visualizing data using t-SNE. *JMLR* (2008).

[vdOVK17] van den Oord A., Vinyals O., Kavukcuoglu K.: Neural discrete representation learning. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*. Curran Associates Inc (2017).

[VSP*17] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I.: Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates Inc (2017), pp. 6000–6010.

[WK24] Wang X., Kalogeiton V.: Conditional gradient-based textual inversion. In *ECCV-W* (2024).

[WLWBL*23] Wang Y., Leng Z., W. B. Li F., Wu S. C., Liang X.: Fg-T2M: Fine-grained text-driven human motion generation via diffusion model. *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE (2023), 21978–21987.

[XJZ*23] Xie Y., Jampani V., Zhong L., Sun D., Jiang H.: Omnicontrol: Control any joint at any time for human motion generation. In *Intl. Conf. on Learning Representations* (2023).

[YLX*19] Yan S., Li Z., Xiong Y., Yan H., Lin D.: Convolutional sequence generation for skeleton-based action synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE (2019), pp. 4393–4401.

[ZBT20] Zhang Y., Black M. J., Tang S.: We are more than our joints: Predicting how 3D bodies move. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* (2020).

[ZCP*22] Zhang M., Cai Z., Pan L., Hong F., Guo X., Yang L., Liu Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. *ArXiv PrePrint* (2022).

[ZDC*24] Zhou W., Dou Z., Cao Z., Liao Z., Wang J., Wang W., Liu Y., Komura T., Wang W., Liu L.: Emdm: Efficient motion diffusion model for fast and high-quality motion generation. In *European Conference on Computer Vision*. Springer (2024), pp. 18–38.

[ZGP*23] Zhang M., Guo X., Pan L., Cai Z., Hong F., Li H., Yang L., Liu Z.: ReMoDiffuse: Retrieval-augmented motion diffusion model. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA (2023), pp. 364–373.

[ZSJ20] Zhao R., Su H., Ji Q.: Bayesian adversarial human motion synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE (2020), pp. 6224–6233.

[ZZC*23] Zhang J., Zhang Y., Cun X., Huang S., Zhang Y., Zhao H., Lu H., Shen X.: T2M-GPT: Generating human motion from textual descriptions with discrete representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE (2023), pp. 14730–14740.

Supporting Information

Additional supporting information may be found online in the Supporting Information section at the end of the article.

Supplemental Video S1

Supplemental Video S2