ARDY: Autoregressive Diffusion with Hybrid Representation for Interactive Human Motion Generation

KAIFENG ZHAO, NVIDIA, Switzerland and ETH Zürich, Switzerland
MATHIS PETROVICH, NVIDIA, Switzerland
HAOTIAN ZHANG, NVIDIA, USA
TINGWU WANG, NVIDIA, USA
SIYU TANG, ETH Zürich, Switzerland
DAVIS REMPE, NVIDIA, USA

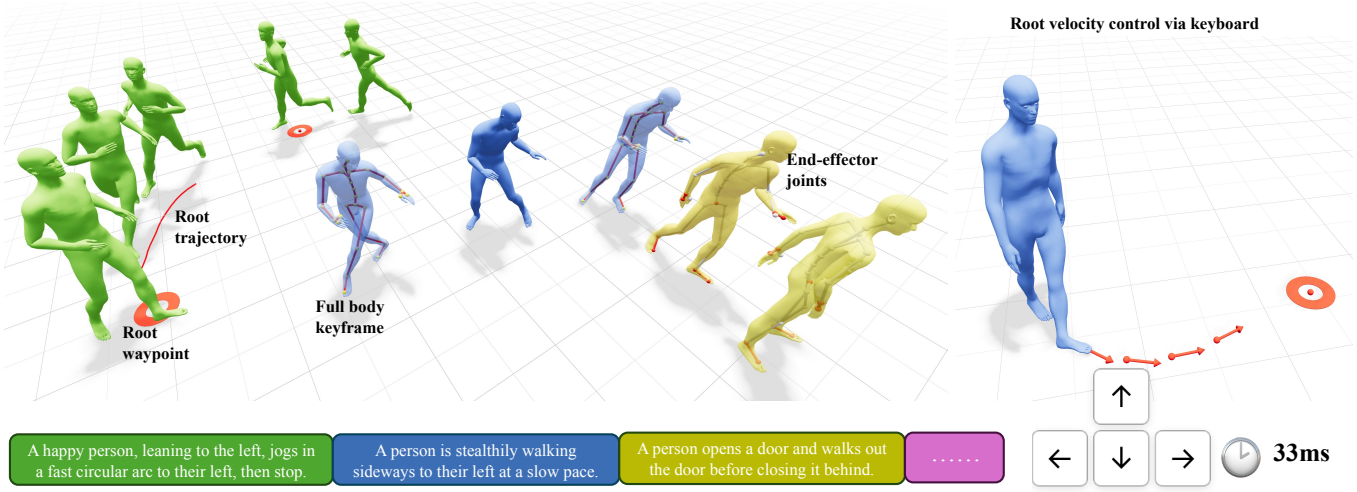

Fig. 1. We present **ARDY**, an autoregressive diffusion model designed for interactive human motion generation. Our approach natively supports online text prompting alongside a comprehensive suite of flexible kinematic constraints — including root waypoints and trajectories, full-body keyframes, and sparse joint positions and rotations — over long horizons. ARDY enables controllable and responsive interactive motion synthesis from real-time user inputs such as mouse and keyboard commands, with our efficient 4-step diffusion model achieving an average generation latency of 33 ms.

Generating realistic 3D human motions in real-time within interactive applications is key for animation, simulation, and humanoid robotics. While recent offline motion generation approaches offer precise control via text and kinematic constraints, they lack the inference speed required for interactive settings. Conversely, existing online methods enable real-time synthesis but often sacrifice controllability or struggle with complex text semantics and long-horizon goals due to limited context windows. In this work, we introduce ARDY, a streaming generation framework that bridges this gap by enabling high-fidelity motion generation controllable via online text prompts and flexible kinematic constraints. ARDY employs a hybrid representation that combines explicit root features with a latent body embedding, balancing precise trajectory control with efficient generative learning. We propose a two-stage autoregressive transformer denoiser that features variable history context and supports conditioning on flexible, long-horizon kinematic constraints. By training on a large-scale motion capture dataset and being directly conditioned on text labels and kinematic constraints sampled from ground truth poses, ARDY natively learns controllable generation that supports online prompting and flexible long-horizon goals. Extensive evaluations on the HumanML3D benchmark and the large-scale, high-fidelity Bones Rigplay dataset demonstrate ARDY's high motion quality and constraint adherence, validating the efficacy of our key architectural decisions. Finally, we demonstrate the method's practical versatility through an interactive demo featuring dynamic text control, diverse keyframe pose constraints, path following, and interactive locomotion control via mouse and keyboard. Supplementary video results, code, and model releases can be found at https://research.nvidia.com/labs/sil/projects/ardy/.

CCS Concepts: • **Computing methodologies → Motion processing**.

Authors' Contact Information: Kaifeng Zhao, kaifeng.zhao@inf.ethz.ch, NVIDIA, Switzerland and ETH Zürich, Switzerland; Mathis Petrovich, mpetrovich@nvidia.com, NVIDIA, Switzerland; Haotian Zhang, haotianz@nvidia.com, NVIDIA, USA; Tingwu Wang, tingwuw@nvidia.com, NVIDIA, USA; Siyu Tang, siyu.tang@inf.ethz.ch, ETH Zürich, Switzerland; Davis Rempe, drempe@nvidia.com, NVIDIA, USA.

ACM 1557-7368/2026/7-ART86
https://doi.org/10.1145/3811284

ACM Reference Format:
Kaifeng Zhao, Mathis Petrovich, Haotian Zhang, Tingwu Wang, Siyu Tang, and Davis Rempe. 2026. ARDY: Autoregressive Diffusion with Hybrid Representation for Interactive Human Motion Generation. *ACM Trans. Graph.* 45, 4, Article 86 (July 2026), 14 pages. https://doi.org/10.1145/3811284

1 Introduction

Learning to generate realistic 3D human motions has become a promising direction with applications ranging from character animation and simulation to humanoid robotics. Offline authoring models can benefit animators and game developers through intuitive controls like text and kinematic constraints [Pinyoanuntapong et al. 2025; Xie et al. 2024]. Meanwhile, interactive motion generators [Shi et al. 2024; Xiao et al. 2025] are key for characters in games and simulations to react to their environment and user inputs in real time. Besides digital humans, recent work in real-world humanoid robot control [He et al. 2025; Liao et al. 2025; Luo et al. 2025; Zhao et al. 2025b] relies heavily on high-quality human motions for supervision during training or planning at runtime.

Recent methods in *offline* motion modeling generate a full sequence of poses in parallel. Modern generative models such as diffusion [Karunratanakul et al. 2023; Rempe et al. 2026; Tevet et al. 2023; Zhang et al. 2024a] and generative masked modeling [Guo et al. 2024; Jiang et al. 2024a; Pinyoanuntapong et al. 2025] allow synthesized motions to follow complex text prompts and kinematic constraints such as pose keyframes and joint positions. While these methods are expressive and controllable, their spatiotemporal design and/or slow inference time are usually not suitable for interactive applications such as computer games or robot control.

In contrast, *online* models generate motion at runtime [Chen et al. 2024; Holden et al. 2017; Ling et al. 2020], usually in an autoregressive fashion. While these models are fast and capable of producing realistic animations, they tend to sacrifice controllability. Some approaches support text conditioning but lack kinematic control [Xiao et al. 2025], while others enable kinematic constraints but can not accept text input [Chen et al. 2024; Shi et al. 2024]. Although a few recent methods integrate both text and kinematic constraints control [Tevet et al. 2025; Zhao et al. 2025a], their restricted context windows limit the understanding of global text semantics and the execution of long-horizon kinematic goals.

In this work, we aim to get the best of both: controllability through complex text prompts and flexible kinematic goal constraints, while generating motion in a streaming fashion that enables online interactivity (see Fig. 1). To achieve this, we introduce **ARDY**, an **A**uto-**R**egressive **D**iffusion model that leverages a h**Y**brid pose representation to generate high-quality motion interactively, conditioned on online text prompts and flexible kinematic constraints from user inputs. ARDY is comprised of two main components. First, ARDY employs a hybrid motion representation that decomposes motion into an explicit root feature and a latent body embedding derived from a learned tokenizer. This hybrid representation enables explicit and accurate root control during generation while maintaining a compact representation for efficient generative learning. Second, ARDY utilizes an autoregressive transformer denoiser for interactive motion generation, conditioned on a text prompt and kinematic constraints that can be spatiotemporally sparse and span long horizons. To handle variable and potentially sparse constraints, we represent the constraints as a masked motion sequence that is injected as input conditioning to the autoregressive denoiser. The denoiser features a variable history context and supports kinematic goals extending beyond a single generation window, which are essential for complex long-term motion semantics and long-horizon kinematic goal reaching. Moreover, the autoregressive denoiser employs an interleaved two-stage architecture: it first predicts the clean explicit root, then predicts the clean latent body embedding conditioned on the first-stage root prediction. These two stages operate in an interleaved manner within the denoising loop, ensuring continuous mutual influence between root and body motion. This staged design is crucial for simultaneously satisfying text instructions and kinematic constraints. By training on a large-scale dataset with text labels and kinematic constraints sampled from the ground truth motion itself, ARDY learns conditional generation that supports online prompting and long-horizon kinematic goals, eliminating the need for additional control modules [Pinyoanuntapong et al. 2025; Shi et al. 2024; Zhao et al. 2025a] such as expensive test-time optimization or RL-based control policies.

We present an interactive demo that highlights the practical capabilities of our method, including dynamic text control, dense and sparse key-pose constraints, path following, and real-time locomotion control via mouse and keyboard. This demonstration showcases the potential for generative models to power next-generation interactive animation systems. Moreover, we validate our design choices on the Bones Rigplay [Bones Studio 2026] dataset—featuring a significantly larger scale and higher quality than the public HumanML3D dataset—to assess the impact of key architectural decisions. Furthermore, we evaluate ARDY against state-of-the-art offline and autoregressive conditional motion generation methods on the public HumanML3D [Guo et al. 2022] benchmark, validating its strong motion quality and kinematic constraint adherence in a controlled setting that isolates the effects of proprietary data.

In summary, the key contributions of this paper are (1) a hybrid latent-body explicit-root representation amenable to fast and controllable motion generation, (2) a two-stage autoregressive diffusion model featuring variable history context length and support for long-horizon kinematic constraint conditioning, including full-body keyframes, root waypoints, root paths, and end-effector positions/rotations, and (3) an extensive evaluation on a large-scale, production-quality dataset that highlights the efficacy of our design choices and demonstrates the strong capabilities of ARDY.

2 Related Work

In this section, we summarize relevant work in conditional 3D human motion generation and how our method fits in context. For this purpose, we define *offline* motion generation as a method that generates a full spatiotemporal sequence of poses in parallel, while *online/interactive/runtime/streaming* motion generation refers to an autoregressive method that generates poses sequentially (either individually or in chunks) and can therefore react to dynamically changing conditions (*e.g.*, new text prompts or constraints).

Offline Human Motion Generation. A primary focus of many recent offline motion generation works is text conditioning. Enabled by motion datasets with natural language descriptions [Plappert et al. 2016], early work on this problem employed VAE-based architectures for diverse generation [Guo et al. 2022; Petrovich et al. 2022]. More recently, diffusion models have proven to be effective at capturing the complex distribution of text and motion, enabling

Table 1. **Method Feature Comparison.** Comparison of the proposed ARDY with existing conditional 3D motion generation methods. We delineate various capabilities including real-time performance, online prompting, supported spatial control types, the architectural mechanism of control (*i.e.*, whether each method requires test-time optimization or RL policies), and the maximum history and future context length in model generation.

Method	Real-time generation	Online text prompting	Spatial control			Native control		Context length (s)	
			Root trajectory	Joint position	Joint rotation	No optimization	No RL policy	History	Future
MaskControl [Pinyoanuntapong et al. 2025]	✗	✗	✓	✓	✗	✗	✓	N/A	10.00
Kimodo [Rempe et al. 2026]	✗	✗	✓	✓	✓	✓	✓	N/A	10.00
AMDM [Shi et al. 2024]	✓	✗	✓	✓	✗	✓	✗	0.03	0.03
CAMDM [Chen et al. 2024]	✓	✗	✓	✗	✗	✓	✓	0.33	1.50
MotionStreamer [Xiao et al. 2025]	✓	✓	✗	✗	✗	N/A	N/A	10.00	10.00
DartControl [Zhao et al. 2025a]	✓	✓	✓	✓	✗	✗	✗	0.07	0.27
DiP [Tevet et al. 2025]	✓	✓	✓	✓	✗	✓	✓	1.00	2.00
ARDY (Ours)	✓	✓	✓	✓	✓	✓	✓	8.00	10.00

high-quality motion generation from prompts [Chen et al. 2023; Tevet et al. 2023; Zhang et al. 2024a]. Motion diffusion models are also capable of flexible kinematic control, enabling "any-joint-any-time" constraints on generated motions [Karunratanakul et al. 2024, 2023; Rempe et al. 2026; Xie et al. 2024]. However, the iterative denoising process for potentially long motions tends to be too slow for interactive applications. Some methods have considerably sped up the denoising process by reducing the number of required steps [Dai et al. 2025; Zhou et al. 2024], but are still designed to generate all poses in parallel. While some diffusion approaches can handle a temporal sequence of input prompts, these methods generate all prompts jointly offline [Barquero et al. 2024; Li et al. 2025; Petrovich et al. 2024], which is not suitable for interactive applications.

Another line of work leverages a discrete tokenized representation of human motion. Methods like MoMask [Guo et al. 2024] and MMM [Pinyoanuntapong et al. 2024b] generate motion from text by training a VQ-VAE motion tokenizer followed by a masked transformer that iteratively predicts masked poses, eventually resulting in a latent motion that can be decoded [Meng et al. 2025; Pinyoanuntapong et al. 2024a]. Some tokenized approaches also support precise kinematic controls through test-time-optimization [Pinyoanuntapong et al. 2025; Wan et al. 2024]. Besides masked models, several approaches take inspiration from language models [Radford et al. 2018] and use autoregressive transformers to generate a sequence of motion tokens that are decoded to human poses [Fan et al. 2025; Jiang et al. 2024a; Lu et al. 2025; Zhang et al. 2023]. While these methods are in fact autoregressive, they are generally large and slow models, designed for offline motion generation without support for precise kinematic control.

Our method ARDY delivers text-following and kinematic control capabilities on par with recent offline models, while operating within an interactive framework. This is achieved through a novel two-stage diffusion architecture that denoises a hybrid combination of latent (tokenized) body and explicit root representations.

Interactive Motion Generation. Early works in autoregressive motion modeling leveraged non-linear latent variable models [Taylor et al. 2006] and recurrent neural networks [Fragkiadaki et al. 2015]. Non-generative autoregressive prediction models [Holden et al. 2017; Starke et al. 2022, 2019] have been trained for reactive character control by conditioning on various combinations of past and future poses and trajectory information. In parallel, data-driven interactive animation systems such as Learned Motion Matching [Holden et al. 2020] and Control Operators [Gou et al. 2025] enable responsive real-time character control via learned similarity metrics and modular control primitives rather than explicit generative modeling. Moving into generative approaches, autoregressive VAE models learned a low-dimensional motion latent space for task-based RL control [Ling et al. 2020; Zhang and Tang 2022] and tracking via optimization [Rempe et al. 2021]. Similar approaches have learned human-object interactions [Hassan et al. 2021; Starke et al. 2019; Zhao et al. 2023] by conditioning the model on object geometry in addition to the future pose information.

Autoregressive motion diffusion models have taken the approaches developed for offline generation and made them amenable to interactive settings, primarily through shorter motion generation horizon and fewer denoising steps [Chen et al. 2024; Ji et al. 2025; Jiang et al. 2024b; Shi et al. 2024; Wu et al. 2025; Zhang et al. 2025, 2024b; Zhao et al. 2025a]. A-MDM learns to denoise the next pose in a motion given the previous pose, and allows flexible kinematic constraints through inpainting or RL control [Shi et al. 2024]. Similarly, CAMDM [Chen et al. 2024] and PRIMAL [Zhang et al. 2025] denoise a small window of future frames given a handful of past frames. CAMDM is conditioned on a future trajectory to follow while PRIMAL relies on guidance and an additional ControlNet for velocity, heading, and waypoint control. While CAMDM and PRIMAL show action label conditioning, none of these methods support complex text prompting. UniPhys [Wu et al. 2025] enables text control, but relies entirely on test-time guidance for kinematic controls, which is inefficient for interactive applications. Closest to our work is DiP [Tevet et al. 2025], which extends CAMDM by adding conditioning on text and 3D target joint locations provided every two seconds. However, DiP's short history and prediction horizon limit its ability to handle complex text prompts that require longer history context, and prevent it from satisfying kinematic constraints beyond its short generation horizon.

Latent diffusion has also been leveraged for interactive motion generation [Cen et al. 2025; Xiao et al. 2025; Zhao et al. 2025a]. DartControl [Zhao et al. 2025a] uses a VAE to learn a continuous latent representation of motion primitives, then a diffusion model that predicts future motion in this latent space. Similar to DiP, DartControl is limited by a short history context, and kinematic control such as 2D waypoint reaching or full-body in-betweening requires test-time-optimization or training an additional RL control policy. MotionStreamer [Xiao et al. 2025] also learns a continuous latent space using a causal convolutional autoencoder, then trains a causal

transformer denoiser to generate the next latent conditioned on the past and text input. Similar to our approach, MotionStreamer is trained on variable history length, making it more robust to complex prompts. However, it lacks support for kinematic goal constraints.

Several autoregressive diffusion models have been paired with physics-based controllers to carry out generated motions in simulation [Huang et al. 2025; Rempe et al. 2023; Ren et al. 2023; Tevet et al. 2025; Wu et al. 2025]. Fully physics-based runtime character control is also an active area of study [Luo et al. 2023; Peng et al. 2022], which has recently enabled both kinematic control and preliminary text prompting [Tessler et al. 2024; Wu et al. 2025].

As shown in Tab. 1, our approach enables real-time generation with native support for online text prompting, variable-length history contexts, and flexible long-horizon kinematic constraints—a combination of capabilities unmatched by prior works.

3 Method: ARDY

Our method ARDY consists of two main components: (1) a motion tokenizer first learns a compact latent representation of body motion, and then (2) an autoregressive two-stage motion diffusion model learns to denoise hybrid motion tokens containing latent body motion and explicit root motion. Our hybrid representation is introduced in Sec. 3.1 followed by the body motion tokenizer in Sec. 3.2. The autoregressive generation problem formulation is detailed in Sec. 3.3 and then the diffusion model that solves it is described in Sec. 3.4. Finally, Sec. 3.5 covers implementation details.

3.1 Hybrid Motion Representation

To balance the representational compactness required for efficient generative learning with the need for direct, precise control via explicit feature overwriting, we propose a hybrid motion representation that decouples root motion from body motion. Specifically, root trajectories are represented in an explicit, interpretable form, while body motion is encoded in a compact latent space. In this section, we give a high-level overview of the hybrid motion representation and its advantages for generation before detailing how the latent component is learned in Sec. 3.2.

Explicit Motion Representation. Our hybrid representation builds on an explicit motion representation, which we describe first for context. Each frame of a motion that uses this explicit representation $\mathbf{m} = (\mathbf{m}_{\text{root}}, \mathbf{m}_{\text{body}}) \in \mathbb{R}^M$ is defined as a tuple of root and body skeleton joint features

$$\mathbf{m}_{\text{root}} = (\mathbf{p}, \cos\psi, \sin\psi) \in \mathbb{R}^5, \quad \mathbf{m}_{\text{body}} = (\boldsymbol{\theta}, \mathbf{J}, \dot{\mathbf{J}}, \mathbf{c}), \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^3$ denotes the global root position, $\psi \in (-\pi, \pi]$ denotes the root heading angle, $\boldsymbol{\theta} \in \mathbb{R}^{6j}$ denotes the 6D representation [Zhou et al. 2019] of the global joint rotations for all j skeleton joints including the root, $\mathbf{J} \in \mathbb{R}^{3j-3}$ denotes the non-root joint positions subtracted by the planar root position, $\dot{\mathbf{J}} \in \mathbb{R}^{3j}$ denotes the global joint velocities, and $\mathbf{c} \in \mathbb{R}^4$ denotes the binary floor contact label for the feet joints. The explicit representation feature size M depends on the number of joints in the skeleton.

Hybrid Motion Representation. Our hybrid motion representation is formed by simply replacing the body component of the pose feature with a latent embedding. Concretely, a single pose $\mathbf{x}$ of a motion using the hybrid representation is a tuple

$$\mathbf{x} = (\mathbf{m}_{\text{root}}, \mathbf{x}_{\text{body}}) \tag{2}$$

where $\mathbf{x}_{\text{body}} \in \mathbb{R}^L$ is the latent body representation with dimensionality L, which has replaced $\mathbf{m}_{\text{body}}$ from the explicit representation. In practice, $\mathbf{x}_{\text{body}}$ is the output of a learned tokenizer (Sec. 3.2) and each token encodes multiple frames of motion. The diffusion model introduced in Sec. 3.4 learns to generate motion using the hybrid representation, which has several advantages. Maintaining root position features in global coordinates avoids potential compounding errors inherent to integrating local velocity-based representations. The global root also facilitates controllable motion generation conditioned on spatial constraints, which are often sparse and defined within the global scene space, as it enables direct overwriting of root features. Moreover, the latent body representation is more compact than explicit representations, and pre-defined after the tokenizer is trained. This makes it better suited for generative modeling, both computationally and in terms of learning efficiency.

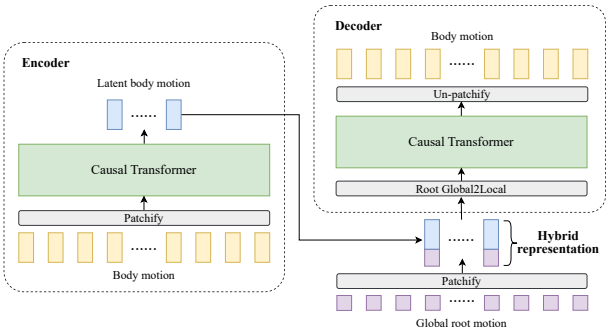

Fig. 2. **Motion Tokenizer.** The encoder first embeds the patchified body motion into a latent representation. This latent body motion is concatenated with the patchified global root motion to form our *hybrid representation*, which is decoded back to reconstruct the body motion.

3.2 Body Motion Tokenizer

We train a motion tokenization network to compress the high-dimensional explicit body features into a compact latent space, facilitating more efficient generative learning. As illustrated in Fig. 2, the tokenizer employs an asymmetric conditional autoencoder architecture. Given an explicit body motion $\mathbf{m}_{\text{body}}^{1:N}$ containing N frames, we treat each P consecutive frames as a patch by reshaping them into a single vector, resulting in $T = N/P$ input vectors to the encoder. The encoder compresses the body motion into latent tokens $\mathbf{x}_{\text{body}}^{1:T} \in \mathbb{R}^{T \times L}$, which are then concatenated along the feature dimension with the patchified explicit root motion $\mathbf{m}_{\text{root}}^{1:T} \in \mathbb{R}^{T \times 5P}$ to form the *hybrid motion tokens*:

$$\mathbf{x}^{1:T} = [\mathbf{m}_{\text{root}}^{1:T}; \mathbf{x}_{\text{body}}^{1:T}] \tag{3}$$

resulting in $\mathbf{x}^{1:T} \in \mathbb{R}^{T \times D}$ where $D = L + 5P$. The decoder subsequently reconstructs the body motion from these hybrid tokens. Crucially, the decoder first transforms the global root motion from

Eq. (1) into a local representation, which replaces the global root motion for the conditional input to the decoder network. Each root pose in the local representation is a tuple $(\dot{\psi}, \dot{\mathbf{p}}_x, \dot{\mathbf{p}}_z, \mathbf{p}_y)$ where $\dot{\psi}$ is the 1D angular velocity of the heading, $\dot{\mathbf{p}}_x$ and $\dot{\mathbf{p}}_z$ are the x and z components of the linear root velocity, and $\mathbf{p}_y$ is the y-component (height) of the root. Note that while the global root representation is useful for *generating* motion as discussed previously, in the tokenizer decoder we find the local representation is more suitable to significantly mitigate foot skating (discussed in Sec. 5.2 and Tab. 2).

We use transformer encoder layers with causal attention in both the encoder and decoder, which ensures that each frame embedding relies only on preceding frames and preserves temporal causality. We experimented with different autoencoder variants for the tokenizer, including variational autoencoder (VAE) [Kingma and Welling 2014] and finite scalar quantization (FSQ) [Mentzer et al. 2023] variants, as detailed in Sec. 5.3. While all variants perform similarly, we find that FSQ demonstrates better stability in training, making it the default tokenizer choice. Training details can be found in Sec. 3.5.

3.3 Controllable Interactive Motion Generation

We aim to develop a motion generation model that supports text and spatial conditions from real-time input streams. At runtime, the model should be reactive to any changes in the input streams like a new text prompt or shift in goal location. Similar to prior work [Chen et al. 2024; Tevet et al. 2025; Wu et al. 2025; Zhang et al. 2025], we formulate this problem as a conditional autoregressive generation task that synthesizes a short window of future motion starting from the current frame, conditioned on past history and optional goal inputs (*i.e.*, kinematic constraints). The synthesized future motion is then played back for the user until re-planning occurs and the model predicts future motion in the new window.

Our autoregressive model operates in the hybrid token space. Assuming that the prediction window starts at token index 1, then our goal is to train the generative model $\mathcal{F}$ to generate the next C tokens in the *current* prediction window:

$$\mathbf{x}^{1:C} = \mathcal{F}(s, \mathbf{x}^{(-H+1):0}, \mathbf{g}^{1:(C+F)}), \tag{4}$$

where s is the text prompt describing the motion semantics of the current generation window, $\mathbf{x}^{(-H+1):0}$ is the history motion spanning up to the previous H tokens, and $\mathbf{g}^{1:(C+F)}$ denotes the spatial goals to achieve. Note that the goals for the first C tokens $\mathbf{g}^{1:C}$ are within the current prediction horizon, while $\mathbf{g}^{(C+1):(C+F)}$ are goals beyond the prediction window, up to F additional *future* tokens.

Notably, H can vary in our formulation, so the model should expect to receive anywhere from 0 to a maximum of H history conditioning tokens. A long history context is crucial to handle text prompts that describe complex non-cyclic motions. For instance, the prompt "walk forward, then bend over and pick something up before continuing to walk" has walking before and after the pick-up action. In autoregressive formulations with limited history context [Tevet et al. 2025; Wu et al. 2025; Zhao et al. 2025a], a model conditioned only on recent walking frames cannot determine whether a preceding pick-up action has already occurred or still needs to be generated, leading to inaccurate generations with missing or duplicated actions.

While autoregressive generation maintains temporal causality (*i.e.*, there is no dependence on future frames), it can still be conditioned on future goals $\mathbf{g}^{1:(C+F)}$. Spatial goals encompass constraints on the motion that specify joint position and/or rotation values at specific timesteps in the future. These can be used to hit 2D waypoints or follow full paths on the ground, full-body pose keyframes, sparse end-effector position constraints, and more. Importantly, our formulation is not limited to goals within the current prediction window, but is also conditioned on goals further in the future. Out-of-window goal constraints implicitly determine the motion generation within the current window, even though they do not directly apply to the immediate frames. For example, when a human needs to run to a location in 10 seconds, the destination goal will determine in which direction the human should start moving from the first step. Supporting such long-horizon goals in previous works requires training an additional RL control policy on top of the autoregressive motion model [Shi et al. 2024; Zhao et al. 2025a], while our model architecture supports this natively.

3.4 Autoregressive Two-Stage Diffusion Model

Based on the hybrid motion representation, we design a transformer-based diffusion denoising model to learn the goal-conditioned autoregressive motion generation task. To further enable precise controllability without sacrificing motion fidelity, we introduce an interleaved two-stage diffusion framework that decomposes the generation of root motion and body motion.

For an introduction to human motion diffusion, we refer the interested reader to prior work [Tevet et al. 2023; Zhang et al. 2024a], and focus here on relevant details for our method. At step k in the denoising process, our diffusion model takes C noisy hybrid motion tokens $\mathbf{x}_k^{1:C} \in \mathbb{R}^{C \times D}$ within the current generation window, along with relevant conditioning, and outputs a prediction for the clean denoised hybrid tokens $\hat{\mathbf{x}}_0^{1:C}$. Mirroring Eq. (4), the denoising process at step k can be written as:

$$\hat{\mathbf{x}}_0^{1:C} = \mathcal{F}(k, s, \mathbf{x}_k^{1:C}, \mathbf{x}^{(-H+1):0}, \mathbf{g}^{1:(C+F)}). \tag{5}$$

The high-level architecture of the denoising network is illustrated in the left side of Fig. 3. The diffusion step and text conditioning are each a single token fed in alongside the sequence of history tokens and noisy tokens for the current prediction window. We use sinusoidal positional encodings for motion tokens to embed their temporal position within the motion sequence, while employing separate learned positional embeddings for text and diffusion tokens. Linear layers are used to project all token types to the same feature dimensionality before feeding to the denoiser.

Spatial Goal Conditioning. We represent spatial goal inputs $\mathbf{g}$ with a masked version of the explicit motion representation from Eq. (1). This allows handling arbitrarily sparse global signals on any pose feature, such as keyframed body or end-effector joints. Only the constrained features and timesteps in $\mathbf{g}$ contain non-zero values while other unconstrained entries are set to zero. We additionally define a corresponding binary mask $\mathbf{v}$ of the same shape, which indicates the dimensions that are constrained. To align with the temporal granularity of the motion tokens, we assume the goal

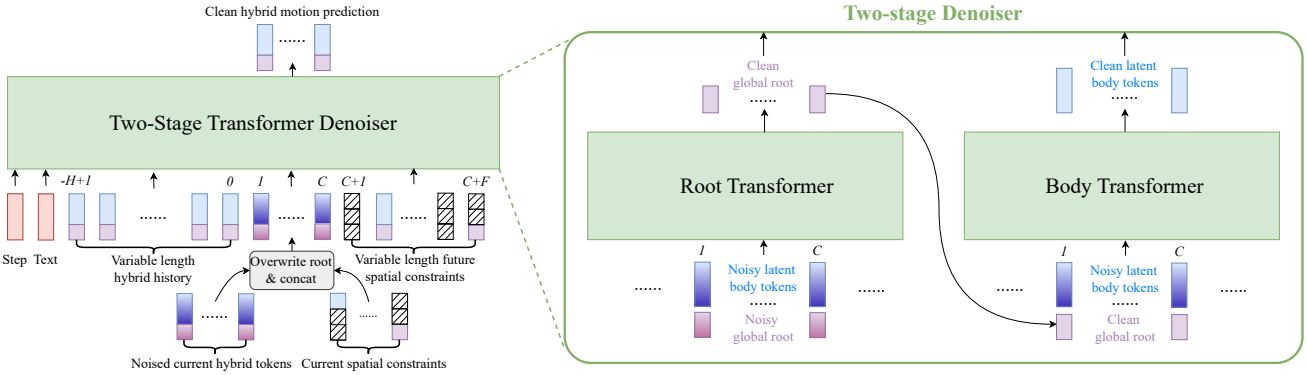

Fig. 3. **Autoregressive Two-Stage Transformer Denoiser.** (Left) Conditioned on a variable-length history context and optional spatial goal constraints, the autoregressive denoiser predicts a sequence of C clean motion tokens within the current generation window. Spatial goal constraints can be arbitrarily sparse and may be located within or beyond the current motion generation window. (Right) The two-stage denoiser first predicts clean global root motion, which then conditions the second stage to predict clean latent body tokens, together forming the complete hybrid motion prediction.

inputs are patchified, for example the short-term goals are $\mathbf{g}^{1:C} \in \mathbb{R}^{C \times MP}$ with patch size P and pose feature dimensionality M.

Before being given to the model, the root part of the noisy tokens $\mathbf{m}^{1:C}_{\text{root}}$ is *overwritten* with the root component of the constraint as $\tilde{\mathbf{m}}^{1:C}_{\text{root}} = (1 - \mathbf{v}_{\text{root}}) \odot \mathbf{m}^{1:C}_{\text{root}} + \mathbf{v}_{\text{root}} \odot \mathbf{g}^{1:C}_{\text{root}}$ where $\odot$ is the element-wise product. This root constraint overwriting [Rempe et al. 2026; Setareh et al. 2024] facilitates highly accurate control over the root trajectory, which governs the fundamental global movement of the human motion. To incorporate constraints on detailed body poses and make the model aware of all constraints, we concatenate the explicit body goal features and the full constraint mask with the input tokens along the feature dimension. In other words, the input noisy tokens are extended with masked constraints to form the augmented representation $[\tilde{\mathbf{m}}^{1:C}_{\text{root}}; \mathbf{x}^{1:C}_{\text{body}}; \mathbf{g}^{1:C}_{\text{body}}; \mathbf{v}]$ where $\mathbf{x}^{1:C}_{\text{body}}$ is the latent body part of the input noisy tokens. Since there are no noisy input tokens beyond the prediction horizon C, the patchified long-horizon constraints $\mathbf{g}^{(C+1):(C+F)} \in \mathbb{R}^{F \times MP}$ are simply concatenated with their corresponding binary mask and fed in as additional tokens to the transformer. These long-horizon goal tokens can vary in length and sparsity depending on user input, with unconstrained tokens masked out during transformer inference.

Interleaved Two-Stage Denoiser. Our autoregressive transformer denoiser employs an interleaved, two-stage design [Rempe et al. 2026] to sequentially predict clean root and body motions. The internals of our transformer-based denoiser are shown on the right side of Fig. 3. At each denoising step, the model first predicts the explicit clean global root motion $\hat{\mathbf{m}}^{1:C}_{\text{root}}$ with the root transformer. Next, the global root motion is detached and fed into the body transformer, which predicts the clean latent body tokens $\hat{\mathbf{x}}^{1:C}_{\text{body}}$. The outputs from both branches are concatenated to form the clean hybrid motion prediction $\hat{\mathbf{x}}^{1:C}_0 = [\hat{\mathbf{m}}^{1:C}_{\text{root}}; \hat{\mathbf{x}}^{1:C}_{\text{body}}]$. During inference, this concatenated hybrid prediction is re-noised for the subsequent diffusion step and fed back into the two-stage denoiser. This iterative and interleaved denoising process ensures continuous mutual influence between the root and body transformers throughout generation. Finally, the predicted hybrid motion representation is processed by the tokenizer's decoder to recover the explicit body motion and form the full, un-patchified explicit motion as $\hat{\mathbf{m}}^{1:G}_0 = [\hat{\mathbf{m}}^{1:G}_{\text{root}}; \hat{\mathbf{m}}^{1:G}_{\text{body}}]$, where the generation window size in frames is $G = C \cdot P$.

Our two-stage architecture is motivated by the hypothesis that predicting body motion conditioned on clean root motion is an easier task than generating both root and body jointly. This decomposition is designed to enable precise controllability without compromising the fidelity of the synthesized motion. As demonstrated in our ablation study in Tab. 2, the proposed two-stage architecture yields better results compared to a monolithic one-stage baseline that simultaneously predicts root and body motion.

3.5 Training and Implementation Details

Motion Tokenizer. In practice, our motion tokenizer uses a patch size of $P = 4$ frames. Both the encoder and decoder are implemented as 8-layer transformers with a latent dimension of 512, utilizing causal self-attention to preserve temporal consistency. The tokenizer is trained on motion clips of varying lengths (1–10 seconds) using a reconstruction loss and additional loss penalizing foot skating:

$$\mathcal{L}_{\text{skate}} = \frac{\sum_{j \in \mathcal{S}_f} \hat{\mathbf{c}}_j \|\hat{\dot{\mathbf{J}}}_j\|_2}{\sum_{j \in \mathcal{S}_f} \hat{\mathbf{c}}_j}, \tag{6}$$

where $\mathcal{S}_f$ represents the set of foot joint indices, $\hat{\mathbf{c}}_j$ denotes the predicted contact label for foot joint j, and $\|\hat{\dot{\mathbf{J}}}_j\|_2$ denotes the magnitude of predicted foot joint velocity. This foot-skating loss penalizes the velocities of joints predicted to be in contact with the ground, thereby enforcing stationary constraints during the contact phase. We set the weight for this loss term to 0.01. The exact implementation of the reconstruction loss depends on the framework being employed for the tokenizer. We test three different approaches including a vanilla continuous autoencoder, VAE, and finite scalar quantization (FSQ) [Mentzer et al. 2023] and compare them in experiments later (Sec. 5.3). For the FSQ variant, we apply finite quantization to the encoder output embedding, constraining each feature to

one of 64 discrete levels. These quantized vectors serve directly as the latent representation. For all tokenizer variations, we train with the AdamAtan2 [Everett et al. 2024] optimizer for 4 million steps using a learning rate of $2e{-}5$ and batch size of 128. We employ a cosine learning rate scheduler with a 10k-step linear warmup phase. Training is performed on a single NVIDIA A100-SXM4-80GB GPU.

Two-Stage Denoiser. Both the root and body transformer in our two-stage denoiser employ the same transformer encoder architecture. Each transformer contains 8 layers with 8 heads and a latent size of 1024, totaling around 156 million parameters for our deployed denoiser model in the interactive demo. For text encoding, we use LLM2Vec [BehnamGhader et al. 2024], which is an embedding model trained on top of Llama-3-8B-Instruct [AI@Meta 2024].

After training the tokenizer, we train the denoiser using the DDPM framework [Ho et al. 2020] with a modified version of the "simplified" loss function that contains several components. In the following discussion, we drop the token/frame index superscripts from all terms for simplicity. First, given the clean hybrid prediction $\hat{\mathbf{x}}_0 = [\hat{\mathbf{m}}_{\text{root}}; \hat{\mathbf{x}}_{\text{body}}]$ and ground truth $\mathbf{x}_0$, the hybrid loss

$$\mathcal{L}_{\text{hybrid}} = ||\hat{\mathbf{x}}_0 - \mathbf{x}_0||_1 \tag{7}$$

uses a smooth L1 loss [Girshick 2015] to penalize errors between the predicted and ground truth hybrid motion tokens. For the next loss, we decode the predicted tokens with the tokenizer decoder $\mathcal{D}$ resulting in the predicted explicit body motion $\hat{\mathbf{m}}_{\text{body}} = \mathcal{D}(\hat{\mathbf{x}}_0)$. Then, the decoded body loss

$$\mathcal{L}_{\text{dec}} = ||\hat{\mathbf{m}}_{\text{body}} - \mathbf{m}_{\text{body}}||_1 \tag{8}$$

compares the predicted explicit body motion to the ground truth $\mathbf{m}_{\text{body}}$. To place greater emphasis on accurately hitting the specified constraints, we add a goal loss

$$\mathcal{L}_{\text{goal}} = ||\mathbf{v} \odot (\hat{\mathbf{m}}_0 - \mathbf{g})||_1 \tag{9}$$

that specifically penalizes components in the full explicit motion prediction $\hat{\mathbf{m}}_0$ that do not hit the constraint goals in $\mathbf{g}$. Finally, we add a regularizer to ensure consistency between the directly predicted joint positions and those resulting from the predicted joint rotations via forward kinmeatics:

$$\mathcal{L}_{\text{consist}} = ||\hat{\mathbf{j}}_0 - \text{FK}(\hat{\boldsymbol{\theta}}_0)||_2 \tag{10}$$

where $\hat{\mathbf{j}}_0$ denotes the predicted joint positions, and the forward kinematics function (FK) outputs joint positions given the predicted joint rotations $\hat{\boldsymbol{\theta}}_0$. The final loss combines all these objectives as

$$\mathcal{L} = \mathcal{L}_{\text{hybrid}} + \mathcal{L}_{\text{dec}} + \mathcal{L}_{\text{goal}} + \mathcal{L}_{\text{consist}}. \tag{11}$$

The two-stage denoiser is trained on sequences with a maximum length of 10 seconds following existing offline motion generation works [Pinyoanuntapong et al. 2025; Tevet et al. 2023]. For each training motion sequence, a fixed-size generation window of G frames is sampled randomly. Consequently, the lengths of the available history (H) and future (F) contexts for each training sample vary dynamically, ranging from 0 to the maximum sequence length minus G. Moreover, we augment the motion sequences by applying random rotations around the y-axis. Spatial constraints for both in-horizon and out-of-horizon are randomly sampled from a set of common use cases including 2D root keyframes, 2D root trajectories, full-body

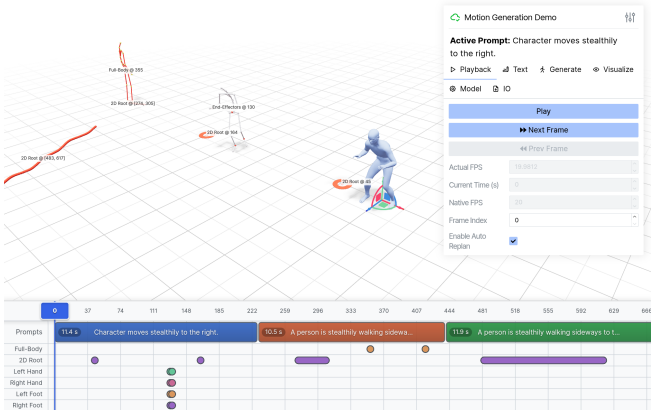

Fig. 4. **Interactive Demo Interface.** This web interface allows generating motion and interacting with ARDY in real time. The *control panel* at the top right allows dynamically changing the text prompt or input constraints. Input constraints are visualized in red within the *3D scene* as the model generates motion to follow them. The *timeline tracks* on the bottom of the interface intuitively show upcoming text prompts and constraints.

sparse keyframes, full-body keyframe blocks, sparse end-effector keyframes, and foot contact keyframes. To enable classifier-free guidance [Ho and Salimans 2021] during inference, we randomly drop the text prompts and spatial constraints with a 10% probability.

By default, we use ten diffusion steps during both train and test-time, which strikes a good balance between speed and accuracy. However, performance is still acceptable for most applications when going as low as four steps (see Sec. 5.3). Denoiser training uses the AdamAtan2 optimizer with a learning rate of $2e{-}5$. Importantly, we do not use dropout in the denoiser as this causes root constraint conditioning inputs to be partially lost. Our denoiser models are trained with a batch size of 512 across four NVIDIA A100-SXM4-80GB GPUs for one million optimization steps.

4 Interactive Motion Generation Demo

To showcase ARDY's versatility, we developed an interface using Viser [Yi et al. 2025] to interactively generate motion with our model. The system, shown in Fig. 4, enables real-time character control through a combination of streaming text prompts and interactive spatial constraints provided via mouse and keyboard inputs. In this section, we first detail ARDY's test-time operation, then qualitatively demonstrate key results through the interactive demo.

4.1 Test-Time Operation

During inference, ARDY operates autoregressively to synthesize motion in response to a dynamic stream of user inputs. In the first step of the motion roll-out, ARDY generates the first window of length G with no history poses as input. In subsequent steps, the previously predicted tokens become the history conditioning as the model predicts the next window of G motion frames. To facilitate autoregressive long motion generation, we employ a truncated sliding window to manage both historical and beyond-generation future contexts. The specific truncation lengths of these context

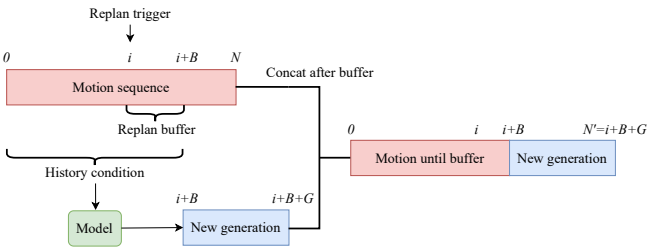

Fig. 5. **Latency-Aware Replanning.** We utilize a non-blocking strategy where a buffer of B frames is simultaneously played back and fed into the generation thread as history context. This buffer effectively hides the inference latency of slower models, ensuring that the transition to the newly generated sequence remains smooth and continuous.

windows are configurable in our interactive demo, up to a maximum of 8 seconds—a limit established by the longest context observed during training. Future constraints that fall beyond the truncation limit (e.g., a target location one minute ahead) are excluded from the input constraint tokens. They are only incorporated into the conditioning once the advancing generation window brings them within the truncated future context horizon. During the autoregressive generation, the root component of the previously predicted tokens are translated such that the last frame of the history coincides with the origin, which is what the model expects as input. The translation offset is preserved and subsequently applied to the generated motion to transform it back into global scene coordinates. This loop ensures high-quality motion with smooth temporal transitions.

To enable real-time interactivity, we incorporate a dynamic replanning mechanism that triggers immediately upon detecting new user input, such as updated text prompts or modified future kinematic constraints, or when the current motion buffer will soon be depleted. Our replanning scheme is latency-aware, facilitating the use of more powerful models even when their inference latency exceeds the inter-frame interval. As shown in Fig. 5, when a replan is triggered we utilize the subsequent B frames, which have already been generated, as a *replan buffer*. These frames are played back to the user while simultaneously serving as history context for the asynchronous generation thread. This replan buffer effectively masks the inference latency of slower models, ensuring smooth and continuous transitions to the new generation. We present this scheme as an optional mechanism to enable increased diffusion steps for enhanced motion quality and control accuracy. In our deployment setup, the 4-step model operates without buffer frames, while the 10-step model employs a single buffer frame.

4.2 Demo Results

The interactive motion generation demo uses ARDY trained on the Bones Rigplay dataset [Bones Studio 2026] described in detail later (see Sec. 5). The demo runs on a workstation equipped with an RTX 4090 GPU. The average generation latency is 33 ms for our efficient 4-step diffusion model and 63 ms for our 10-step diffusion model, with the latter providing slightly improved control accuracy. Both models use a generation window of $G = 40$ frames (2 seconds at 20 fps). All examples in Fig. 1 are generated using this interface, demonstrating that the system can process complex descriptions and seamlessly adapt to dynamic changes in user-specified text prompts. It also robustly satisfies diverse kinematic constraints, ranging from sparse long-term goals (e.g., reaching a target location in 10 seconds) to dense short-term constraints (e.g., trajectory following or full-body keyframes). Additional qualitative results for kinematic constraint-conditioned generation are shown in Fig. 6.

Our system also supports diverse locomotion interfaces: users can define target root trajectories in real time using mouse-based waypoints or modulate real-time velocity via keyboard commands. For mouse-based root path control, we derive the target trajectory by linearly interpolating between mouse-click waypoints and smoothing the resulting path. For keyboard-based root velocity control, we compute a target velocity from user input and the current velocity, then linearly interpolate between the two and integrate the resulting per-frame velocities to derive the root trajectory input to the model. Extensive video demonstrations of our interactive generation system are provided in the supplementary material, highlighting its responsiveness and high-fidelity motion quality.

5 Analysis on Large-Scale Mocap Data

Next, we thoroughly analyze key design choices of ARDY along with the effects of various hyperparameter settings.

5.1 Experiment Setting

Bones Rigplay Mocap Dataset. We leverage the large-scale proprietary Bones Rigplay dataset [Bones Studio 2026], which contains around 700 hours of diverse studio-quality human motion with text descriptions. The scale and quality of this data enables a more robust testbed for evaluating design variations compared to smaller public datasets like HumanML3D [Guo et al. 2022], which are saturated as indicated by methods scoring higher than ground truth data on metrics like R-precision. This dataset contains motions from more than 150 participants and is retargeted to a unified-proportion 27-joint skeleton to facilitate learning. The motions encapsulate thousands of distinct behaviors, each performed by multiple actors for multiple takes, resulting in a diverse distribution of semantics and kinematic variations. It includes common motion categories such as locomotion, everyday activities, gestures, and combat, performed in a variety of styles. Raw motion clips range from 1 to 180 seconds in length, but we clip motions to a maximum of 10 seconds and subsample to 20 fps for training. To improve generalization, we use LLM to generate diverse paraphrases of the original text labels. The dataset is split into training and test sets by first grouping motion clips according to semantic content (*i.e.*, action type, such as "eating_apple_right"), and then assigning disjoint groups to each split with an approximate 90/10 ratio, resulting in about 315k motion clips for training and 35k for testing. As a result, the test set contains motion categories that are entirely unseen during training, providing a stronger evaluation of generalization to novel actions.

Constraints Sampling. We evaluate text+constraint-conditioned generation across a comprehensive suite of test cases designed to simulate common downstream applications. These scenarios include dense root trajectory following, sparse waypoints navigation, full-body keyframes, and end-effector joints control (incorporating both

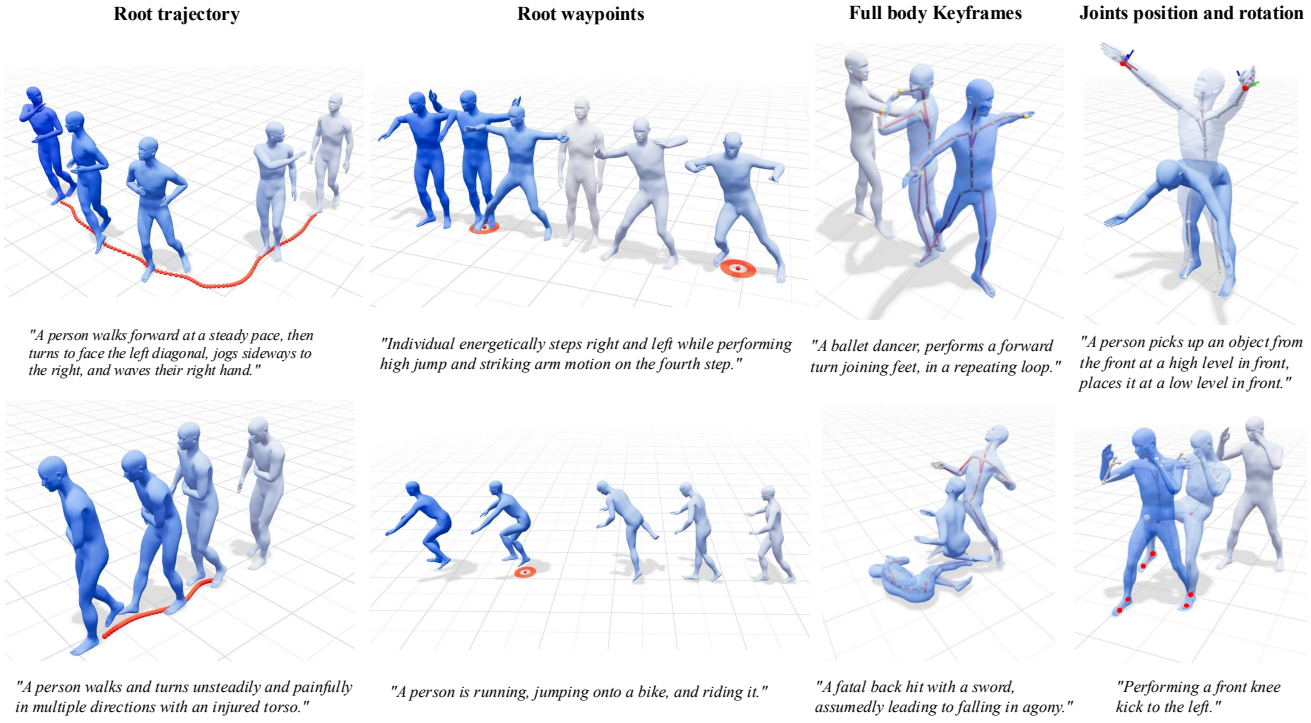

Fig. 6. **Motion Generation with Kinematic Constraints.** Qualitative results for motion generation conditioned on text prompts and diverse kinematic constraints, including dense root trajectories, sparse root waypoints (visualized as red rings), full-body keyframes (visualized as red skeletons), sparse joint positions (visualized as white skeletons with constrained joints highlighted as red spheres), and joint rotations (visualized as coordinate axes centered at the constrained joint). Motion temporal progression is indicated by a color gradient from gray to blue.

position and orientation goals). The spatial constraints are sampled directly from the ground-truth test set alongside their corresponding text prompts. Furthermore, to rigorously evaluate the model's robustness against constraint inputs, we introduce slight random perturbations to the global translation and heading of a subset of sampled constraints during the evaluation.

Evaluation Metrics. Following established protocols [Guo et al. 2022], we employ Fréchet Inception Distance (**FID**) to quantify the distributional similarity between generated and ground-truth motions, and Top-3 **R-precision** to assess text-motion alignment. To ensure a rigorous evaluation, we train a robust evaluator model based on TMR [Petrovich et al. 2023] using the large-scale Bones Rigplay dataset. Notably, we compute R-precision over a test dataset containing about 5k unique samples of diverse action types. This significantly increases the retrieval difficulty compared to the standard practice in benchmarks like HumanML3D [Guo et al. 2022], which computes the metric over batches of size 32 only. As a proxy for motion quality, we also report a heuristic **foot skating** metric that measures mean foot velocity when the foot is considered in-contact based on a height threshold. To assess spatial control accuracy in constraints-conditioned generation, we compute the **mean error** between the user-specified constraint targets (position and orientation) and the corresponding generated poses.

5.2 Ablation Study

Tab. 2 presents ablation results on three key design choices: the hybrid motion representation, the global-to-local root conversion within the tokenizer decoder, and the two-stage denoiser design.

Hybrid Motion Representation. We first compare our proposed hybrid motion representation (derived via the learned tokenizer) against the purely explicit motion representation. To ensure a fair comparison, we train an autoregressive baseline that uses explicit pose features, applying the same patching strategy to align the temporal granularity of the tokens. This explicit baseline uses masked overwriting (Sec. 3.4) to condition on all kinematic constraint inputs by overwriting both constrained root and body features. As demonstrated in Tab. 2, our autoregressive model utilizing the hybrid representation significantly outperforms its explicit counterpart in both motion quality and control accuracy. The high-dimensionality of explicit motion representations likely complicates the generative learning process, particularly under our few-step denoising setting. In contrast, the hybrid representation compresses high-dimensional body features into compact latent embeddings that are more amenable to efficient generative modeling.

Global-to-Local Conversion. Next, we evaluate the importance of our global-to-local root conversion process within the tokenizer decoder by training a baseline decoder that operates directly on the

Table 2. **Quantitative Ablation of Architectural Designs.** We evaluate performance across text-only and various kinematic constraints-conditioned generation scenarios, including end-effector joint rotation and position, full-body keyframe joints, dense root trajectories, and sparse root waypoints. ↑ denotes higher values are better; ↓ denotes lower values are better. **Bold** and underlined values indicate the best and second-best results, respectively.

	Text-only Generation			Constraints-conditioned Generation					
Model	Skate (m/s) ↓	R-prec. ↑	FID ↓	Skate (m/s) ↓	Joint rot. (deg.) ↓	Joint pos. (m) ↓	Keyframe body (m)↓	Traj. (m) ↓	Waypoint (m) ↓
Dataset	0.255	76.56	0.000	-	-	-	-	-	-
ARDY (Ours)	**0.264**	65.47	**0.027**	0.250	2.23	**0.025**	**0.023**	**0.015**	**0.024**
Explicit representation	0.365	53.90	0.065	0.281	**1.67**	0.130	0.136	0.033	0.203
Global root-conditioned decoder	0.303	64.94	0.028	0.284	2.88	0.048	0.044	0.024	0.060
One-stage architecture	**0.264**	**65.84**	0.029	**0.248**	2.46	0.101	0.079	0.017	0.164

Table 3. **Hyperparameter and Tokenizer Analysis.** The **best** results in each group are highlighted in bold, and the second best are underlined. The ablation table is divided into five sections, sequentially comparing: (1) generation horizon, (2) diffusion steps, (3) tokenizer patch sizes, (4) tokenizer latent space capacities (latent embedding quantization levels and dimensions), and (5) various tokenizer types. The default configuration in each section is marked with *.

Model			Text-only Generation			Constraints-conditioned Generation					
Tokenizer	Horizon	Diffusion step	Skate (m/s) ↓	R-prec. ↑	FID ↓	Skate (m/s) ↓	Joint rot. (deg.) ↓	Joint pos. (m) ↓	Keyframe body (m)↓	Traj. (m) ↓	Waypoint (m) ↓
Generation horizon											
FSQ 64-128, Patch 4	4	10	**0.151**	33.42	0.224	0.445	9.23	0.848	0.864	0.864	0.850
FSQ 64-128, Patch 4	8	10	0.258	56.70	0.037	**0.243**	3.45	0.031	0.026	**0.013**	**0.020**
FSQ 64-128, Patch 4	12	10	0.254	59.54	0.033	0.247	2.94	0.033	0.028	0.017	0.031
FSQ 64-128, Patch 4	20	10	0.255	63.80	0.030	0.250	2.61	0.046	0.037	0.014	0.059
FSQ 64-128, Patch 4	40*	10	0.264	**65.47**	**0.027**	0.250	**2.23**	**0.025**	**0.023**	0.015	0.024
Diffusion steps											
FSQ 64-128, Patch 4	40	1	0.411	56.74	0.079	1.405	25.39	1.040	1.054	1.037	1.002
FSQ 64-128, Patch 4	40	2	0.239	61.28	0.052	0.360	7.96	0.174	0.169	0.274	0.163
FSQ 64-128, Patch 4	40	3	0.231	63.59	0.041	0.254	3.58	0.053	0.051	0.046	0.044
FSQ 64-128, Patch 4	40	4	**0.230**	64.41	0.034	**0.249**	2.68	0.034	0.032	0.028	0.027
FSQ 64-128, Patch 4	40	10*	0.264	65.47	0.027	0.250	**2.23**	**0.025**	**0.023**	0.015	**0.024**
FSQ 64-128, Patch 4	40	100	0.282	**65.49**	**0.025**	0.257	2.71	0.030	0.027	**0.009**	0.028
Tokenizer patch size											
FSQ 64-128, Patch 1	40	10	0.298	44.45	0.152	0.355	2.31	0.764	0.816	0.790	0.775
FSQ 64-128, Patch 4*	40	10	**0.264**	65.47	0.027	**0.250**	**2.23**	**0.025**	**0.023**	**0.015**	**0.024**
FSQ 64-128, Patch 8	40	10	0.317	**68.01**	**0.022**	0.295	3.05	0.070	0.062	0.018	0.100
Tokenizer latent space capacity											
FSQ 16-32, Patch 4	40	10	0.283	**68.11**	**0.023**	0.261	4.57	0.031	0.026	0.016	0.020
FSQ 64-32, Patch 4	40	10	0.273	67.62	0.025	0.252	3.96	0.026	**0.023**	**0.014**	**0.017**
FSQ 64-128, Patch 4*	40	10	**0.264**	65.47	0.027	**0.250**	**2.23**	**0.025**	**0.023**	0.015	0.024
FSQ 64-256, Patch 4	40	10	0.268	64.04	0.031	0.257	2.31	0.030	0.025	0.015	0.032
Tokenizer type											
AE 128D, Patch 4	20	10	0.266	62.20	0.033	**0.246**	2.23	**0.044**	0.040	0.016	**0.057**
VAE 128D, Patch 4	20	10	0.259	63.35	0.031	0.250	**2.17**	0.046	0.042	**0.014**	0.058
FSQ 64-128, Patch 4*	20	10	**0.255**	**63.80**	**0.030**	0.250	2.61	0.046	**0.037**	**0.014**	0.059

global root representation. The ablation results reveal that removing the global-to-local root conversion leads to a notable increase in foot skating, confirming that local root representations are essential for preserving motion quality and physical plausibility.

Two-Stage Denoiser Design. To validate our two-stage model architecture, we train a one-stage baseline that jointly predicts the root trajectory and latent body motion tokens simultaneously. Experimental results show that the two-stage architecture achieves superior performance, yielding higher-fidelity text-conditioned motion and significantly lower spatial constraint errors. This suggests that decomposing root and body prediction facilitates the simultaneous learning of high-fidelity generation and precise spatial control.

5.3 Hyperparameter and Tokenizer Type Analysis

Tab. 3 provides an analysis of the generation horizon length, the number of diffusion steps, and the tokenizer configurations.

Generation Horizon. The generation horizon length is a critical hyperparameter impacting the model's performance. We observe that extending the horizon consistently improves motion fidelity (FID) and semantic alignment (R-Precision) metrics. Conversely, extremely narrow horizons (*e.g.*, 4 frames) lead to training instability and degraded performance, ultimately resulting in the generation of drifting motions. The text-only foot-skating metric for the 4-frame horizon is misleadingly low, as the model often fails to respond to text prompts. Regarding spatial control, we find that horizons of 8 and 40 frames effectively minimize the constraint errors. Qualitative analysis shows that models with an 8-frame horizon can transition between actions more rapidly in response to updated text prompts

compared to those with a 40-frame horizon. Furthermore, our experiments show that the 8-frame model learns constraint adherence faster during training than its 40-frame counterpart.

Diffusion Step. We ablate the impact of the number of diffusion steps used by the autoregressive denoiser. Using extremely few diffusion steps (*e.g.*, 1 or 2) leads to significantly worse generation quality and constraint adherence. Increasing diffusion steps provides slight gains in FID, R-Precision, and constraint accuracy. However, our few-step models still achieve highly competitive performance, demonstrating the robustness of the learned hybrid representation for efficient high-quality motion synthesis.

Tokenizer Patch Size. We also evaluate the effect of the tokenizer patch size. Using a minimum patch size of a single frame leads to faster learning in the early stages, but causes training instability later on, resulting in significantly worse overall performance in the end. Conversely, using a larger patch size of 8 slightly improves the FID and R-precision metrics, but at the cost of worse skating performance and constraint accuracy. This trade-off occurs because compressing more frames into a single token causes a greater loss of fine-grained pose details within each patch.

Tokenizer Latent Space Capacity. We evaluate tokenizers with varying latent space capacities. The capacity of a Finite Scalar Quantization (FSQ) latent space is determined jointly by the number of discrete quantization levels and the number of latent dimensions. By default, we use an FSQ configuration with 64 quantization levels and 128 dimensions, denoted as FSQ 64-128. While performance is relatively similar across configurations, there are some notable differences. Using FSQ 16-32 with a smaller latent capacity yields slightly better FID and R-precision metrics under the limited training budget of 1 million iterations, but it degrades performance on end-effector joint constraints and full-body errors. This trade-off arises because a smaller latent space lacks the capacity to represent fine-grained motion details accurately. On the other hand, expanding the number of dimensions to 256 slows model convergence and does not provide performance gains within the same train budget.

Tokenizer Type. We experiment with several tokenizer architectures, including Variational Autoencoders (VAE) and Finite Scalar Quantization (FSQ). For the VAE variant, we applied a KL-divergence loss with weight of 1×10^{-6} to regularize the latent distribution. Our results indicate that all tokenizer variants perform comparably to a vanilla autoencoder. However, the vanilla autoencoder suffers from severe training instability and diverges when trained with longer horizons, such as 40 frames. In contrast, the FSQ tokenizer demonstrates superior training stability over the vanilla autoencoder baseline, leading us to adopt FSQ as our default choice.

6 Benchmark Evaluation

Lastly, we evaluate ARDY against both offline and online state-of-the-art baselines for text+constraints-conditioned generation on the standard HumanML3D [Guo et al. 2022] dataset. For these experiments, our model is trained with a 40-frame generation horizon using 10 diffusion steps and a vanilla autoencoder tokenizer.

Table 4. **Offline Text and Constraint Control Comparison.** Evaluation results of text-conditioned motion generation with joint position goals on HumanML3D. * denotes methods without test-time optimization. ↑ denotes higher values are better; ↓ denotes lower values are better.

Method	R-Prec. ↑	FID ↓	Skate (%) ↓	Error (cm) ↓	Latency (s) ↓
Dataset (HumanML3D retarget)	0.739	0.000	7.92	0.00	-
Dataset (Our retarget)	0.732	0.011	6.87	0.00	-
Without optimization					
MaskControl* [Pinyoanuntapong et al. 2025]	**0.760**	0.050	7.27	46.18	0.46
ARDY (Ours)*	0.729	**0.044**	**6.28**	**4.15**	**0.15**
With optimization					
MaskControl [Pinyoanuntapong et al. 2025]	**0.758**	**0.047**	7.87	0.45	68.65
ARDY (Ours) Opt	0.721	0.088	**5.87**	**0.30**	**9.25**

6.1 Experiment Setting

HumanML3D Dataset. This public dataset contains around 30 hours of motion data with corresponding text descriptions. In our experiments, we exclude the HumanAct12 [Guo et al. 2020] subset of HumanML3D due to the absence of native joint rotation data and the severe motion artifacts introduced by the original preprocessing. During data processing, we preserve the original SMPL [Loper et al. 2015] joint rotations in the retargeting step, unlike the original HumanML3D pipeline, which discards native joint rotations. This makes our processed data compatible with real-time animation, since we can directly animate the body model with generated joint rotations instead of going through an expensive inverse kinematics post-process using generated joint positions.

Evaluation Metrics. We adopt the evaluation benchmark from prior work [Guo et al. 2022; Pinyoanuntapong et al. 2025] to assess various aspects of the generated motion. To evaluate text-following, we report the Top-3 **R-precision**. Motion quality is measured via Fréchet Inception Distance (**FID**), which indicates similarity to the ground-truth distribution, and the **foot skating ratio**, which quantifies the frequency of detected foot skating frames. To assess spatial control accuracy, we calculate the **mean joint error** for the constrained goal joint positions. We utilize the original HumanML3D evaluator models, which were trained on the original processed HumanML3D data, to calculate FID and R-precision metrics. As a result, our method is slightly disadvantaged on these metrics. Additionally, we report the motion generation **latency** for each method, measured on a single NVIDIA A100-SXM4-80GB GPU.

6.2 Offline Model Comparison

We first compare to MaskControl [Pinyoanuntapong et al. 2025], a SOTA offline motion generation model that specializes in accurate joint controls. Following the protocol in MaskControl, we evaluate the model's ability to satisfy arbitrary joint position constraints at any given frame. We first compare our raw generation results against MaskControl with its test-time optimization module disabled (denoted as MaskControl*). Subsequently, we apply a similar test-time optimization to our predicted hybrid motion to minimize joint errors. We then compare these refined results against the full MaskControl pipeline. As shown in Tab. 4, ARDY achieves competitive text-following (on par with ground truth R-prec) and motion quality while demonstrating a lower foot skating ratio. Notably, compared to the raw MaskControl output before optimization, our

Table 5. **Autoregressive Text and Constraint Control Comparison.** evaluation results of text-conditioned autoregressive motion generation with in-horizon and out-of-horizon sparse joint goals on HumanML3D. ↑ denotes higher values are better; ↓ denotes lower values are better.

Method	R-Prec. ↑	FID ↓	Skate (%) ↓	Error (cm) ↓	Latency (s) ↓
Dataset (HumanML3D retarget)	0.711	0.000	8.53	0.00	-
Dataset (Our retarget)	0.711	0.010	7.00	0.00	-
In-horizon goals					
DiP [Tevet et al. 2025]	0.609	0.967	12.29	9.20	0.15
ARDY (Ours)	**0.690**	**0.092**	**7.07**	**2.48**	0.15
Out-of-horizon goals					
DiP [Tevet et al. 2025]	0.599	1.453	11.07	17.64	0.15
ARDY (Ours)	**0.684**	**0.100**	**7.63**	**2.92**	0.15

method yields significantly lower spatial control errors, indicating a stronger underlying generative prior.

6.3 Autoregressive Model Comparison

Next, we compare ARDY to the closely related model DiP [Tevet et al. 2025], an autoregressive motion diffusion model. For the autoregressive model comparison, we evaluate constraints satisfaction by sampling goal joints using two distinct schemes. The first scheme, termed *in-horizon goals*, follows the original DiP setting by sampling one goal joint at the final frame of each autoregressive generation window. This scheme necessitates a goal input every 2 seconds, which is often impractical for applications relying on sparser control signals. The second scheme, *out-of-horizon goals*, involves sampling a single final goal joint at the very end of a long sequence which is beyond the initial autoregressive generation window. This configuration creates a challenging scenario requiring long-horizon planning, a task that the DiP system fails to handle effectively. Following the implementation of DiP, we sample the goal joints from the pelvis, wrists, and feet. We set the test sequence length to 9 seconds and provide 1 second of ground truth motion as initial history to adapt to the original DiP implementation.

As presented in Tab. 5, our approach surpasses DiP in both in-horizon and out-of-horizon scenarios. Notably, DiP exhibits a sharp increase in joint error under the out-of-horizon setting, indicating its limitation for long-term planning. In contrast, our method effectively resolves these long-context constraints, maintaining high accuracy even when goals are placed far into the future. Furthermore, to ensure our quantitative gains translate to actual human perception, we conduct a side-by-side perceptual study comparing the two methods on motion quality, semantic alignment, and joint goal accuracy for out-of-horizon goals. Participants are instructed to vote for the better result or indicate a tie. Across 240 pairwise human comparisons (Tab. 6), our approach ARDY is strongly and consistently preferred over DiP, confirming that the numerical improvements in Tab. 5 reflect genuine qualitative gains.

7 Discussion

We propose ARDY, an autoregressive motion diffusion model that enables interactive and controllable human motion generation. ARDY natively supports online text prompting and flexible kinematic goal constraints tailored to interactive applications, including long-horizon goals that extend beyond a single generation window. We present a real-time demonstration of interactive and instructable motion generation, underscoring the potential of generative models for future animation systems. We validate our architectural decisions through extensive ablation studies on the large-scale, studio-quality Bones Rigplay dataset. Furthermore, experiments on the public HumanML3D benchmark demonstrate that ARDY outperforms existing methods in terms of both motion fidelity and control accuracy.

Table 6. **Perceptual Study Results.** We report the percentage of human preferences comparing our method against DiP across three criteria. Our approach is consistently preferred over DiP, with a significant margin in motion quality, semantic alignment, and goal accuracy.

	Ours (%)	Tie (%)	DiP [Tevet et al. 2025] (%)
Motion Quality	65.8	25.0	9.2
Semantic Alignment	67.5	25.0	7.5
Goal Accuracy	64.6	31.2	4.2

Limitations. While ARDY demonstrates a promising system for interactive human motion generation, several aspects of the design remain open for future research improvement. First, ARDY explicitly utilizes all past motion frames as the history context during autoregressive generation, which can be inefficient for extremely long-horizon tasks. Exploring more efficient, structured memory representations and update mechanisms is an important future direction. Second, as a diffusion model, ARDY relies on a multi-step iterative generation process, which can be computationally demanding. This could potentially be further accelerated by combining our approach with recent advances in shortcut diffusion models [Geng et al. 2025; Lu and Song 2025]. Third, ARDY is a purely kinematic model and lacks awareness of physical dynamics. Consequently, artifacts such as foot skating and jittering can sometimes be observed in the generated motions. A crucial future direction is to integrate physics modelling into ARDY, proposing a unified generative model capable of predicting both the kinematics and dynamics of human motion, which is essential for physics-critical applications.

8 Acknowledgments

We would like to thank Edy Lim, Eugene Jeong, Sam Wu, Ehsan Hassani, Michael Huang, and Jin-Bey Yu for their help with data processing and cleaning, and Cyrus Hogg, Simon Yuen, Lindsey Pavao, Jenna Diamond, Rizwan Khan, Samantha Shinagawa, and Akanksha Shukla for their efforts on data acquisition and labeling. We also thank the anonymous reviewers for their valuable feedback.

References

AI@Meta. 2024. Llama 3 Model Card. (2024). https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

German Barquero, Sergio Escalera, and Cristina Palmero. 2024. Seamless Human Motion Composition with Blended Positional Encodings. (2024).

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders. In *First Conference on Language Modeling*. https://openreview.net/forum?id=IW1PR7vEBf

Bones Studio. 2026. AI Datasets for Machine Learning and Motion Capture. https://bones.studio/datasets. Accessed: 2026.

Zhi Cen, Huaijin Pi, Sida Peng, Qing Shuai, Yujun Shen, Hujun Bao, Xiaowei Zhou, and Ruizhen Hu. 2025. Ready-to-React: Online Reaction Policy for Two-Character Interaction Generation. In *ICLR*.

Rui Chen, Mingyi Shi, Shaoli Huang, Ping Tan, Taku Komura, and Xuelin Chen. 2024. Taming Diffusion Probabilistic Models for Character Control. In *ACM SIGGRAPH 2024 Conference Papers* (Denver, CO, USA) *(SIGGRAPH '24)*. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3641519.3657440

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. 2023. Executing your Commands via Motion Diffusion in Latent Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18000–18010.

Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. 2025. Motionlcm: Real-time controllable motion generation via latent consistency model. In *ECCV*. 390–408.

Katie Everett, Lechao Xiao, Mitchell Wortsman, Alexander A Alemi, Roman Novak, Peter J Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, et al. 2024. Scaling exponents across parameterizations and optimizers. *International Conference on Machine Learning* (2024).

Ke Fan, Shunlin Lu, Minyue Dai, Runyi Yu, Lixing Xiao, Zhiyang Dou, Junting Dong, Lizhuang Ma, and Jingbo Wang. 2025. Go to Zero: Towards Zero-shot Motion Generation with Million-scale Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. arXiv:2507.07095 [cs.CV] https://arxiv.org/abs/2507.07095

Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent network models for human dynamics. In *Proceedings of the IEEE international conference on computer vision*. 4346–4354.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. 2025. Mean Flows for One-step Generative Modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Ross Girshick. 2015. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*.

Ruiyu Gou, Michiel van de Panne, and Daniel Holden. 2025. Control Operators for Interactive Character Animation. *ACM Transactions on Graphics (TOG)* (2025).

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. 2024. Momask: Generative masked modeling of 3d human motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1900–1910.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. 2022. Generating Diverse and Natural 3D Human Motions From Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5152–5161.

Chuan Guo, Xinxin Zuo, Sen Wang, Shihao Zou, Qingyao Sun, Annan Deng, Minglun Gong, and Li Cheng. 2020. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM international conference on multimedia*. 2021–2029.

Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. 2021. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 11374–11384.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. 2025. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143* (2025).

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

Jonathan Ho and Tim Salimans. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. 2020. Learned motion matching. *ACM Transactions on Graphics (ToG)* (2020).

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–13.

Xiaoyu Huang, Takara Truong, Yunbo Zhang, Fangzhou Yu, Jean Pierre Sleiman, Jessica Hodgins, Koushil Sreenath, and Farbod Farshidian. 2025. Diffuse-cloc: Guided diffusion for physics-based character look-ahead control. *ACM Transactions on Graphics (TOG)* 44, 4 (2025), 1–12.

Kaiyang Ji, Ye Shi, Zichen Jin, Kangyi Chen, Lan Xu, Yuexin Ma, Jingyi Yu, and Jingya Wang. 2025. Towards immersive human-x interaction: A real-time framework for physically plausible motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10173–10183.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. 2024a. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems* 36 (2024).

Nan Jiang, Zimo He, Zi Wang, Hongjie Li, Yixin Chen, Siyuan Huang, and Yixin Zhu. 2024b. Autonomous character-scene interaction synthesis from text instruction. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. 2024. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1334–1345.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. 2023. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2151–2162.

Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

Chuqiao Li, Julian Chibane, Yannan He, Naama Pearl, Andreas Geiger, and Gerard Pons-Moll. 2025. Unimotion: Unifying 3d human motion synthesis and understanding. In *2025 International Conference on 3D Vision (3DV)*. IEEE, 240–249.

Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Guy Tevet, Koushil Sreenath, and C Karen Liu. 2025. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion. *arXiv preprint arXiv:2508.08241* (2025).

Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. 2020. Character controllers using motion vaes. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 40–1.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-Person Linear Model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)* (2015).

Cheng Lu and Yang Song. 2025. Simplifying, Stabilizing and Scaling Continuous-time Consistency Models. In *The Thirteenth International Conference on Learning Representations*.

Shunlin Lu, Jingbo Wang, Zeyu Lu, Ling-Hao Chen, Wenxun Dai, Junting Dong, Zhiyang Dou, Bo Dai, and Ruimao Zhang. 2025. Scamo: Exploring the scaling law in autoregressive motion generation model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 27872–27882.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. 2023. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582* (2023).

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castañeda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Zi Wang, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi Fan, and Yuke Zhu. 2025. SONIC: Supersizing Motion Tracking for Natural Humanoid Whole-Body Control. *arXiv preprint arXiv:2511.07820* (2025).

Zichong Meng, Yiming Xie, Xiaogang Peng, Zeyu Han, and Huaizu Jiang. 2025. Rethinking Diffusion for Text-Driven Human Motion Generation: Redundant Representations, Evaluation, and Masked Autoregression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 27859–27871.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. 2023. Finite scalar quantization: Vq-vae made simple. *arXiv preprint arXiv:2309.15505* (2023).

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. 2022. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)* 41, 4 (2022), 1–17.

Mathis Petrovich, Michael J. Black, and Gül Varol. 2022. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision (ECCV)*.

Mathis Petrovich, Michael J. Black, and Gül Varol. 2023. TMR: Text-to-Motion Retrieval Using Contrastive 3D Human Motion Synthesis. In *International Conference on Computer Vision (ICCV)*.

Mathis Petrovich, Or Litany, Umar Iqbal, Michael J. Black, Gül Varol, Xue Bin Peng, and Davis Rempe. 2024. Multi-Track Timeline Control for Text-Driven 3D Human Motion Generation. In *CVPR Workshop on Human Motion Generation*.

Ekkasit Pinyoanuntapong, Muhammad Saleem, Korrawe Karunratanakul, Pu Wang, Hongfei Xue, Chen Chen, Chuan Guo, Junli Cao, Jian Ren, and Sergey Tulyakov. 2025. MaskControl: Spatio-Temporal Control for Masked Motion Synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9955–9965.

Ekkasit Pinyoanuntapong, Muhammad Usama Saleem, Pu Wang, Minwoo Lee, Srijan Das, and Chen Chen. 2024a. Bamm: Bidirectional autoregressive motion model. In *European Conference on Computer Vision*. Springer, 172–190.

Ekkasit Pinyoanuntapong, Pu Wang, Minwoo Lee, and Chen Chen. 2024b. Mmm: Generative masked motion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1546–1555.

Matthias Plappert, Christian Mandery, and Tamim Asfour. 2016. The kit motion-language dataset. *Big data* 4, 4 (2016), 236–252.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. 2021. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11488–11499.

Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. 2023. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13756–13766.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, Jiefeng Li, Chen Tessler, Edy Lim, Eugene Jeong, Sam Wu, Ehsan Hassani, Michael Huang, Jin-Bey Yu, Chaeyeon Chung, Lina Song, Olivier Dionne, Jan Kautz, Simon Yuen, and Sanja Fidler. 2026. Kimodo: Scaling Controllable Human Motion Generation. *arXiv:2603.15546* (2026).

Jiawei Ren, Mingyuan Zhang, Cunjun Yu, Xiao Ma, Liang Pan, and Ziwei Liu. 2023. InsActor: Instruction-driven Physics-based Characters. *NeurIPS* (2023).

Cohan Setareh, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. 2024. Flexible Motion In-betweening with Diffusion Models. (2024).

Yi Shi, Jingbo Wang, Xuekun Jiang, Bingkun Lin, Bo Dai, and Xue Bin Peng. 2024. Interactive Character Control with Auto-Regressive Motion Diffusion Models. *ACM Trans. Graph.* 43 (jul 2024).

Sebastian Starke, Ian Mason, and Taku Komura. 2022. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (ToG)* 41, 4 (2022), 1–13.

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural state machine for character-scene interactions. *ACM Transactions on Graphics* 38, 6 (2019), 178.

Graham W Taylor, Geoffrey E Hinton, and Sam Roweis. 2006. Modeling human motion using binary latent variables. *Advances in neural information processing systems* 19 (2006).

Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. 2024. MaskedMimic: Unified Physics-Based Character Control Through Masked Motion Inpainting. *ACM Transactions on Graphics (TOG)* (2024).

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit Haim Bermano, and Michiel van de Panne. 2025. CLoSD: Closing the Loop between Simulation and Diffusion for multi-task character control. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=pZISppZSTv

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

Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. 2024. Tlcontrol: Trajectory and language control for human motion synthesis. In *European Conference on Computer Vision*. Springer, 37–54.

Yan Wu, Korrawe Karunratanakul, Zhengyi Luo, and Siyu Tang. 2025. UniPhys: Unified Planner and Controller with Diffusion for Flexible Physics-Based Character Control. *arXiv preprint arXiv:2504.12540* (2025).

Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. 2025. MotionStreamer: Streaming Motion Generation via Diffusion-based Autoregressive Model in Causal Latent Space. *arXiv preprint arXiv:2503.15451* (2025).

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. 2024. OmniControl: Control Any Joint at Any Time for Human Motion Generation. In *The Twelfth International Conference on Learning Representations*.

Brent Yi, Chung Min Kim, Justin Kerr, Gina Wu, Rebecca Feng, Anthony Zhang, Jonas Kulhanek, Hongsuk Choi, Yi Ma, Matthew Tancik, and Angjoo Kanazawa. 2025. Viser: Imperative, web-based 3d visualization in python. *arXiv preprint arXiv:2507.22885* (2025).

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Shaoli Huang, Yong Zhang, Hongwei Zhao, Hongtao Lu, and Xi Shen. 2023. T2M-GPT: Generating Human Motion from Textual Descriptions with Discrete Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. 2024a. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence* 46, 6 (2024), 4115–4128.

Yan Zhang, Yao Feng, Alpár Cseke, Nitin Saini, Nathan Bajandas, Nicolas Heron, and Michael J. Black. 2025. PRIMAL: Physically Reactive and Interactive Motor Model for Avatar Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Yan Zhang and Siyu Tang. 2022. The wanderings of odysseus in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20481–20491.

Zihan Zhang, Richard Liu, Rana Hanocka, and Kfir Aberman. 2024b. Tedi: Temporally-entangled diffusion for long-term motion synthesis. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

Kaifeng Zhao, Gen Li, and Siyu Tang. 2025a. DartControl: A Diffusion-Based Autoregressive Motion Model for Real-Time Text-Driven Motion Control. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. 2023. Synthesizing diverse human motions in 3d indoor scenes. In *Proceedings of the IEEE/CVF international conference on computer vision*. 14738–14749.

Siheng Zhao, Yanjie Ze, Yue Wang, C Karen Liu, Pieter Abbeel, Guanya Shi, and Rocky Duan. 2025b. ResMimic: From General Motion Tracking to Humanoid Whole-body Loco-Manipulation via Residual Learning. *arXiv preprint arXiv:2510.05070* (2025).

Wenyang Zhou, Zhiyang Dou, Zeyu Cao, Zhouyingcheng Liao, Jingbo Wang, Wenjia Wang, Yuan Liu, Taku Komura, Wenping Wang, and Lingjie Liu. 2024. Emdm: Efficient motion diffusion model for fast and high-quality motion generation. In *European Conference on Computer Vision*. Springer, 18–38.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5745–5753.