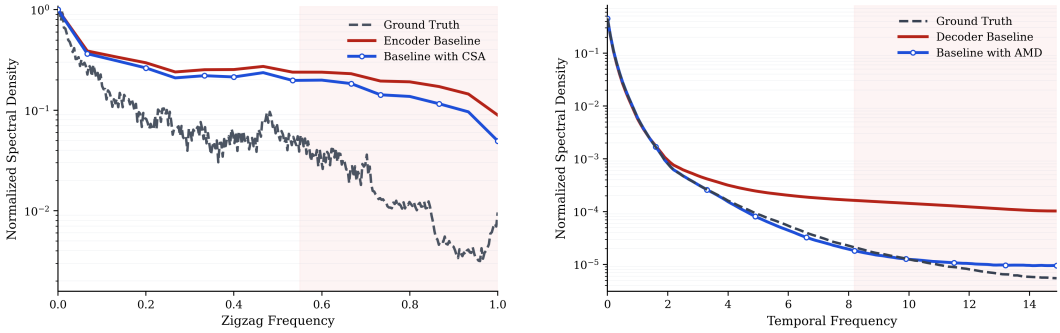
Improving the Diffusability of Motion Tokenizer

Guanhe Huang[1]* **Songqiao Han**[2]* **Tangzheng Lian**[1] **Oya Celiktutan**[1]
[1]Department of Engineering, [2]Department of Mathematics
King's College London, London, United Kingdom
{guanhe.huang,songqiao.han,lian.tangzheng,oya.celiktutan}@kcl.ac.uk



(a) Latent spectral profile (b) Motion spectral profile

Figure 1: **Spectral mismatch in motion tokenizers.** (a) The baseline encoder produces high-frequency shortcut outliers into the latent space. Our Critic Spectral Adaptation (CSA) uses the pre-trained diffusion as a frequency-specific critic to suppress this shortcut high-frequency energy. (b) The baseline deterministic decoder introduces severe high-frequency artifacts into the reconstructed motion. The Alpha-Flow Motion Decoder (AMD) calibrates the spectral profile to closely approximate the real motion dynamics.

Abstract

Latent diffusion models paired with a variational auto-encoder (VAE) tokenizer have shown promising performance for efficient text-driven human motion generation. However, motion latent diffusion suffers from a generation-reconstruction trade-off: simply increasing tokenizer capacity enhances reconstruction but fails to proportionally improve text-conditioned generation quality. We assume this trade-off stems from a spectral mismatch inside the VAE tokenizer: the encoder yields a latent space biased toward high frequencies, and the decoder introduces temporal artifacts. In this paper, we propose the Diffusable Motion Tokenizer (DiMoT) to improve the *diffusability* [1] of the tokenizer, rather than scaling up the diffusion backbone. To improve text-conditioned optimization, Critic Spectral Adaptation (CSA) leverages a pretrained diffusion model as the critic to suppress high-frequency latent energy. To preserve motion dynamics, the Alpha-Flow Motion Decoder (AMD) performs generative decoding via average-velocity prediction, enabling efficient one-step decoding. Extensive experiments demonstrate that DiMoT achieves state-of-the-art performance across various challenging text-to-motion benchmarks, yielding superior motion fidelity and text-motion alignment. Project page: `https://githubhgh.github.io/dimot/`.

*Equal contribution.
[1]Diffusability describes how easily the latent space can be fitted by diffusion.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Generative modeling is entering a *Token Era*, with text [1, 2, 3] and video [4, 5, 6] tokenization becoming standard practice in large generalist models. Conversely, motion sequences present unique challenges due to their inherent kinematic structure and strict temporal sensitivity, which are also more expensive and time-consuming to collect. For example, human joints have strict rotation limits, and any minor temporal misalignment could result in obvious artifacts such as jitter and foot sliding. Consequently, developing a robust motion tokenizer under these constraints is critical for advancing downstream applications in humanoid robotics [7, 8, 9, 10] and virtual reality [11, 12, 13, 14].

Latent Diffusion Models (LDMs) [15, 16, 17, 18, 19, 20] paired with a pre-trained Variational Autoencoder (VAE) tokenizer have advanced rapidly in motion generation, extending from single-person [21, 22, 23, 24, 25] to multi-person [26, 27, 28] scenarios. Compared with the diffusion backbone, the tokenizer has been less explored, yet it is equally important for high-quality motion generation. While continuous representations are naturally suited for producing smooth motion [29], they struggle to capture the discrete nature of conditioning signals (*e.g.*, text). Furthermore, the tokenizer and generative backbone are typically trained in a decoupled manner, where existing literature often relies on the VAE's reconstruction accuracy as a proxy for generation quality. However, such an optimal latent space for reconstruction struggles to achieve the same superiority in conditional diffusion generation [30]. Previous studies [31, 32] attribute this gap to the spectral mismatch: the latent space contains abundant high-frequency energy, whereas raw motion is predominantly low-frequency, which degrades downstream generation. In this paper, we propose the Diffusable Motion Tokenizer (DiMoT). Specifically, DiMoT couples Critic Spectral Adaptation (CSA) to regularize the encoder's latent space with Alpha-Flow Motion Decoder (AMD) to efficiently preserve motion dynamics.

Firstly, it is well-known that VAEs tend to over-smooth motion sequences into low-frequency content [33]. Surprisingly, we empirically find in Fig. 1a that the encoder yields a latent space biased toward high frequencies absent from the raw motion. We interpret this as an ill-posed shortcut code in the VAE that easily amplifies back into motion space, while the diffusion backbone struggles to model the text-conditioned distribution with such artificial outliers [31]. However, naively applying a uniform high-frequency penalty also suppresses the necessary dynamics for motion reconstruction. We therefore introduce CSA to use the pretrained diffusion model as a text-conditioned critic regularizer [34]. Moreover, CSA re-weights the regularization loss mode by mode (*i.e.*, Fourier) for spectral reorganization, which adaptively suppresses the VAE's shortcut outliers. By reweighting the latent space critic regularization with the spectral profile, CSA improves the diffusability of the tokenizer while maintaining comparable reconstruction performance.

Beyond the latent representation, the VAE decoder sets the performance ceiling for generation quality; decoupled training isolates it from the diffusion prior. Consequently, this mismatch implies that a well-optimized latent space does not guarantee accurate motion reconstruction during inference. Previous works [29, 35] employ iterative refinement within diffusion decoding, which uses the latent as conditioning signals to directly guide motion generation. However, the iterative nature of diffusion decoders introduces significant computational overhead, contradicting the primary efficiency of LDMs. To address it, we introduce the AMD, an efficient one-step generative decoder that learns the average velocity field [36, 37]. As illustrated in Fig. 1b, AMD's generative decoding matches the real motion frequency distribution, whereas deterministic decoding inherently introduces high-frequency artifacts. Therefore, the AMD improves the diffusability by calibrating the spectral profile to efficiently preserve the essential motion dynamics.

In summary, our core contributions are:

- We introduce **Critic Spectral Adaptation** (**CSA**), leveraging a pre-trained LDM as a critic for tokenizer encoder regularization. Guided by the text-conditioned distribution, CSA adaptively suppresses shortcut high-frequency energy via spectral reweighting. Furthermore, we provide a theoretical analysis detailing how CSA improves generalized latent diffusion frameworks.

- We propose the **Alpha-Flow Motion Decoder (AMD)** to replace the tokenizer decoder with the one-step generative decoder. By learning the latent-conditioned average velocity field, AMD calibrates essential motion dynamics without iterative refinement.

- We integrate these modules into the **Diffusable Motion Tokenizer (DiMoT)** to conduct comprehensive evaluations across multi-person and single-person benchmarks. Extensive results show DiMoT demonstrates superior fidelity and text following ability over existing LDM frameworks.

2 Related Works

Tokenizer Diffusability Bottleneck. Recent works [31, 38, 39, 40] have identified the spectral inconsistency between the latent space and raw data as a primary bottleneck for LDM generation quality. Specifically, vanilla VAE encoders tend to push the latent space into biased high-frequency modes, which diffusion models struggle to fit [31]. Existing solutions attempt to mitigate this by penalizing latent DCT amplitudes [31], matching input and latent power spectra [40], aligning latents with external features [30, 38], or jointly training the encoder with a diffusion decoder [39, 41]. Other approaches regularize latents toward lower frequencies [32] or apply a frequency curriculum during diffusion training without altering the autoencoder [42]. In contrast, CSA leverages the same pre-trained diffusion as a per-frequency critic to fine-tune the encoder. This requires no extra module, external feature extractor, or joint training.

Motion Generation in the Latent Space. LDMs [16] are widely adopted in text-to-motion generation due to their effective trade-off between computational efficiency and generation quality [21, 22, 26, 27, 28, 29, 35, 43, 44, 45, 46]. Early works focus on single-person generation [21], which are further improved by consistency distillation for real-time inference [22]. Different from individual motion generation, multi-person generation is more challenging due to the extra relative orientation and position modeling. Geng et al. [26] applies a decoupled latent space via disentangling the interaction context and individual motion. Recently, the focus has shifted to unified latent spaces capable of handling variable agents and tasks via shared policies [46] or domain-specific motion priors [45]. However, coherence between the tokenizer and the diffusion backbone remains underexplored. Concurrent works bridge this coherence gap via Gaussian mixture regularization [44] or generative discrete decoding [29]. These methods either ignore the inherent high-frequency outliers in the decoder output or introduce significant computational overhead. Instead, we replace the VAE decoder with AMD to preserve the essential motion dynamics, achieving high-fidelity, one-step generative decoding [41].

3 Methodology

Problem Statement. Given a user-input text y, the goal is to generate a natural and smooth motion sequence x that aligns with the language semantics. This can be modeled with a text-conditioned Latent Diffusion Model (LDM), where motion is generated in a latent space and decoded into the final motion sequence. Standard LDMs use a decoupled approach: training a diffusion model f_θ strictly on the frozen latents $z = E_\varphi(x)$ of a pre-trained VAE tokenizer (E_φ, D_ψ). Rather than scaling up the diffusion backbone, we target improving the tokenizer (E_φ and D_ψ) to maximize its *diffusability*.

To achieve this, we present the Diffusable Motion Tokenizer (DiMoT), a framework that addresses the spectral mismatch and bridges the performance gap between tokenizer reconstruction and diffusion generation. As illustrated in Fig. 2, DiMoT consists of two core components: (1) Critic Spectral Adaptation (CSA), which uses a pre-trained diffusion as the critic to adaptively regularize the latent spectral profile of encoder E_φ, and (2) Alpha-Flow Motion Decoder (AMD), which replaces the decoder D_ψ with one-step generative decoding to preserve temporal motion dynamics. In this way, DiMoT achieves high-quality text-conditioned generation while simultaneously preserving reconstruction fidelity.

3.1 Critic Spectral Adaptation (CSA)

We introduce Critic Spectral Adaptation (CSA) to fine-tune the encoder E_φ via spectral reweighting, leveraging a pre-trained text-conditioned diffusion model as a critic [34, 47]. We begin by formulating the spectral critic loss, followed by a theoretical analysis detailing how CSA facilitates the optimization of the downstream diffusion objective.

Diffusion as a Critic. By employing the pre-trained diffusion model f_{θ_0} as a generative critic, CSA optimizes the latent space manifold toward improved text-conditioned generability, rather than trivially fitting the reconstruction objective. Let E_φ be the pre-trained motion VAE encoder with the decoder D_{ψ_0} pinned at its baseline parameters ψ_0, and let $\mathcal{L}_{\text{rec}}(\varphi)$ denote the reconstruction loss as a function of the encoder. The frozen text-conditioned rectified-flow latent diffusion model $f_{\theta_0}(x_t, t, y)$ is trained on the baseline tokenizer's latents [16]. For motion-prompt pair $(x, y) \sim p_{\text{data}}$ and latent $z = E_\varphi(x)$, sample interpolation time $t \sim \mathcal{U}[0, 1]$ and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$. Set $x_t = (1-t)\,\text{sg}(z) + t\epsilon$. The prediction loss/residual is

$$r(z) \;=\; \text{sg}\big(f_{\theta_0}(x_t, t, y)\big) - (\epsilon - z), \tag{1}$$

where sg is the stop-gradient. The stop-gradients keep f_{θ_0} a fixed critic rather than a trainable model to differentiate through, restricting parameter updates solely to the encoder. Finally, the expected squared residual gives the (plain) critic loss $\mathcal{L}_{\text{crit}}(\varphi) = \mathbb{E}_{(x,y),t,\epsilon}\,\|r(z)\|^2$ (See Fig. 4a results).

Spectral Reweighting. Inspired by [31], we further improve the critic loss in the frequency domain. The Fast Fourier Transform (FFT) $\mathcal{F}$ decomposes $r(z)$ along the time axis into Fourier modes indexed by k, where $k = 0$ is the static component and larger k corresponds to faster temporal variations. The residual amplitude $|\mathcal{F}[r(z)](k)|$ is small at modes the diffusion handles well, and large at the shortcut high-frequency modes the encoder introduces (Fig. 1a). However, $\mathcal{L}_{\text{crit}}$ weights every mode equally, so we reweight it across modes to concentrate the encoder's response on the shortcut band:

$$\mathcal{L}_{\text{CSA}}(\varphi) \;=\; \mathbb{E}_{(x,y),\,t,\,\epsilon}\left[\frac{1}{N_t}\sum_k \tilde{w}(k)\,\big|\mathcal{F}[r(z)](k)\big|^2\right]. \tag{2}$$

Here, the weight $\tilde{w}(k) = w(k)/\bar{w}$ with $w(k) = 1 + \kappa\, d(k)^p$ is a static spectral weight ($d(k) \in [0, 1]$ denotes FFT distance from $k = 0$, p denotes polynomial exponent). The strength knob κ controls how aggressively the encoder pressure is biased toward the high-frequency band. The gradient on z at each mode is proportional to $\tilde{w}(k)\,\hat{r}(k)$ (Theorem 13, Appendix A.4), so modes already handled by the diffusion receive little encoder gradient. Algorithm 1 summarizes the resulting training step.

Algorithm 1 CSA Training Loop.

Require: encoder E_φ, frozen diffusion f_{θ_0}, FFT length N_t, total steps S

Pre-compute spectral weights $\tilde{w}(k) = (1 + \kappa\, d(k)^p)/\bar{w}$ for $k = 0, \ldots, N_t - 1$ ▷ static, FFT-geometry only

for step $i = 1, \ldots, S$ **do**

  Sample $(x, y) \sim p_{\text{data}}, \quad t \sim \mathcal{U}[0, 1], \quad \epsilon \sim \mathcal{N}(0, I)$ ▷ joint motion-prompt batch

  $z \leftarrow E_\varphi(x), \quad x_t \leftarrow (1-t)\,\text{sg}(z) + t\epsilon$ ▷ noised interpolant, encoder takes x only

  $r \leftarrow \text{sg}\big(f_{\theta_0}(x_t, t, y)\big) - (\epsilon - z)$ ▷ Eq. (1), text y enters here

  $\mathcal{L}_{\text{CSA}} \leftarrow N_t^{-1} \sum_k \tilde{w}(k)\,|\mathcal{F}[r](k)|^2$ ▷ Eq. (2), FFT along temporal axis

  Update φ on $\mathcal{L}_{\text{rec}}(\varphi) + \lambda_c \min(1, i/W)\,\mathcal{L}_{\text{CSA}}(\varphi)$ ▷ ψ_0, θ_0 frozen, W-step warmup

end for

CSA Improves Downstream Diffusion. While E_φ (φ denotes the encoder parameters) operates solely on motion inputs, CSA guides the encoder's gradients by incorporating the prompt y via the critic's text-conditioned prediction $\text{sg}(f_{\theta_0}(x_t, t, y))$. Moreover, we provide the theoretical foundations detailing how it improves general conditional diffusion optimization within this learned latent space.

Define the text-conditioned diffusion training loss as $\mathcal{L}_{\text{diff}}(\theta; \varphi) \coloneqq \mathbb{E}_{(x,y),t,\epsilon}\,\|f_\theta(x_t, t, y) - (\epsilon - z)\|^2$. Under reasonable conditions, we analyze the CSA regularization from the bilevel optimization theory [47, 48, 49]. Consequently, we prove that CSA translates into a strict improvement in $\mathcal{L}_{\text{diff}}$ as follows:

Theorem 1 (*Diffusion-loss improvement*)**.** *Adopt the assumptions and notation of Theorem 20 (Appendix A.6). Suppose also* $\theta_0 = \arg\min_\theta \mathcal{L}_{\text{diff}}(\theta; \varphi_0)$ *and* $g_{\text{crit}}^\top H^{-1} g > 0$*, where* $g_{\text{crit}} \coloneqq \nabla_\varphi \mathcal{L}_{\text{diff}}(\theta_0; \varphi_0)$ *and* $H \coloneqq \nabla_\varphi^2 \mathcal{L}_{\text{rec}}(\varphi_0)$*. Then for all sufficiently small* $\lambda_c > 0$*,*

$$\min_\theta \mathcal{L}_{\text{diff}}\big(\theta;\, \varphi^*(\lambda_c)\big) \;<\; \min_\theta \mathcal{L}_{\text{diff}}\big(\theta;\, \varphi_0\big). \tag{3}$$

The alignment condition is treated as a working assumption (Appendix A.7).

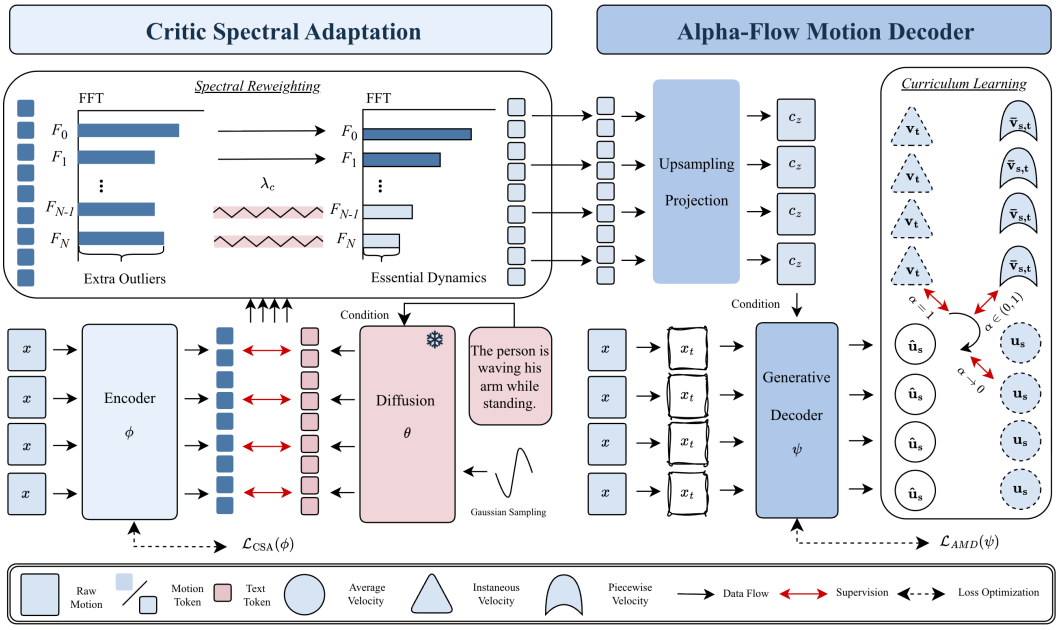


Figure 2: Overview of our DiMoT framework. Left: Critic Spectral Adaptation (CSA). CSA utilizes a pre-trained latent diffusion model as a text-conditioned critic, which adaptively suppresses shortcut high-frequency outliers via spectral reweighting. Right: Alpha-Flow Motion Decoder (AMD). By using curriculum learning over trajectory velocity fields in different time scales, AMD gradually learns one-step average-velocity generative decoding with latent conditions.

Intuitively, at the CSA-trained encoder $\varphi^*(\lambda_c)$, training the diffusion from scratch yields **a strictly lower training loss** than at the baseline φ_0. Meanwhile, when fine-tuning, reconstruction is theoretically preserved at first order in λ_c along the same trajectory (Theorem 20, Appendix A.6). Even without optimality of the baseline diffusion, Corollary 26 yields the same conclusion for fixed θ_0.

As a cross-modality check, we apply CSA to a class-conditional **ImageNet** [50, 51] pipeline (Appendix B). In line with Theorem 1, generation FID improves at the evaluated diffusion training checkpoints (i.e., $100k$, $200k$ and $300k$ steps). Reconstruction FID remains close to baseline, suggesting information is reorganized rather than removed.

3.2 Alpha-Flow Motion Decoder

As shown in Fig. 1, while deterministic decoding inherently produces this high-frequency noise, generative decoding can match the frequency distribution of real motion. Compared to the VAE encoder, the VAE decoder directly determines the upper bound of LDM generation performance, while it only plays the static role as a passive receiver in a mismatched way [41]. Such a mismatch becomes increasingly disruptive for the motion tokenizer. Iterative generative decoding can mitigate it, which also introduces prohibitive computational overhead. To this end, we propose the Alpha-Flow Motion Decoder (**AMD**) in Fig. 2, a latent-conditioned one-step generative decoder that preserves the efficiency while restoring motion details lost by standard VAE decoding [31, 32].

Latent Condition Upsampling Projection. Given latent tokens $z \in \mathbb{R}^{N_z \times d_z}$ from a frozen diffusion backbone, we aim to recover a frame-aligned condition sequence $c_z \in \mathbb{R}^{T \times d_C}$. Unlike [29], our continuous latent tokens avoid strict frame-to-frame correspondence, where each token is aligned with a coarse temporal segment. Rather than naively interpolating these tokens, we map the compressed memory to target frames using a Transformer decoder for upsampling projection. We first linearly project each latent token into the conditioning space, $m = \mathrm{Proj}(z) \in \mathbb{R}^{N_z \times d_C}$, which serves as the memory sequence. Meanwhile, we construct a set of learnable frame queries $q \in \mathbb{R}^{T \times d_C}$, where each query represents one target frame position. After adding positional encoding to both the memory and query sequences, we feed them into a transformer decoder, where the queries interact with each other through self-attention and retrieve motion information from the latent memory through

cross-attention: $c_z = \mathrm{TransDec}(q, m) \in \mathbb{R}^{T \times d_C}$. This global cross-attention ensures each condition vector c_z captures the full semantic context, providing the fine-grained temporal alignment required for accurate motion generation.

Generative Decoding on Average Velocity Field. After obtaining the latent condition, we apply AMD [52] as a generative backbone to generate the motion sequences. Compared to vanilla rectified flow matching [53], AMD learns to predict the average velocity field in a one-step setting, without relying on costly iterative refinement. In detail, vanilla flow matching learns local denoising directions instantaneously, whereas AMD predicts a long-range transport toward motion target X via velocity integration [37]. Moreover, AMD applies curriculum training to achieve smooth transitions between different trajectory flow matching, which shows better convergence and stability.

Following MeanFlow [37], we introduce the average velocity field for one-step diffusion decoding. Given a destination time $r_d < t$, the decoder u_ψ predicts an average transition velocity $u(x_t, r_d, t, z)$ over the interval $[r_d, t]$ conditioned on latent space z as follows:

$$u(x_t, r_d, t, z) = \frac{1}{t - r_d} \int_{r_d}^{t} v(x_\tau, \tau, z)\, d\tau, \tag{4}$$

where x is the clean motion sample and Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, $x_t = (1-t)x + t\epsilon, v_t = \epsilon - x$, where $t \in [0, 1]$, $x_0 = x$, and $x_1 = \epsilon$. By conditioning explicitly on the destination time, the model can represent large trajectory jumps that are necessary for direct one-step decoding based on *MeanFlow identity*. However, the original MeanFlow's training target is not tractable and depends on the network prediction. By differentiating the integral definition (see Appendix C for derivation), the average velocity derivative target is calculated via the Jacobian-vector product of the decoder network:

$$u(x_t, r_d, t, z) = v(x_t, t, z) - (t - r_d)\frac{d}{dt} u_{\psi^-}(x_t, r_d, t, z). \tag{5}$$

In practice, MeanFlow training configuration sets $t = r_d$ for 75% of samples, providing instantaneous flow-matching supervision [37]. AMD instead uses $t = r_d$ for 50% of each batch, while the remaining samples with $r_d < t$ learn interval-average velocities for one-step decoding. The α schedule and training details are provided in Appendix E.

Alpha-Flow Curriculum Learning To mitigate such reliance on trivial instantaneous velocity prediction, AMD performs one-step average velocity supervision by gradually shifting from fast-converging trajectory flow matching to intractable MeanFlow optimization. In practice, AMD enforces consistency between interval predictions defined over (t, s, r_d):

$$s = \alpha r_d + (1 - \alpha)t, \alpha \in (0, 1], \quad x_s = x_t - (t - s)\tilde{v}_{s,t}, \tag{6}$$

where α is the time-variant scheduler, and x_s is the motion state at timestep s. Based on this connection, we train the decoder with an α-curriculum rather than a fixed objective (*e.g.*, $s = t$ for vanilla flow matching). The AMD objective is formally defined as:

$$\mathcal{L}_{\mathrm{AMD}}(\psi) = \mathbb{E}_{x,\epsilon,t,r_d}\left[\frac{1}{\alpha}\left\| u_\psi(x_t, r_d, t, z) - \left(\alpha\tilde{v}_{s,t} + (1-\alpha)u_{\psi^-}(x_s, r_d, s, z)\right)\right\|_2^2\right], \tag{7}$$

where $\tilde{v}_{s,t} = v_t$ is used to form the shift target in the endpoint value. Specifically, when $\alpha = 1$, it reduces to trajectory flow matching, which provides stable transition supervision. When $\alpha \to 0$, it recovers the MeanFlow regime, which is better aligned with the final goal of one-step generation.

Notably, the latent can come from text-conditioned diffusion backbones or VAE encoders. A decoder trained with a tokenizer latent may still face a train-test mismatch at inference time, while the diffusion backbone could produce training samples with high variance. To reduce this mismatch, we use a mixed training scheme by randomly replacing the latent samples produced.

At inference time, we instantiate decoding with a single interval from $t = 1$ to $r_d = 0$, starting from Gaussian noise in raw motion space. Conditioned on the latent token, the decoder predicts the whole average velocity field of raw motion sequences in one step: $\hat{x} = x_1 - u_\psi(x_1, 0, 1, z)$. In this way, we use the AMD to replace deterministic decoding with efficient but more effective motion realization, preserving the coherent high-frequency dynamics (See the video in the supplementary material).

3.3 DiMoT Justification

Training the Tokenizer for Generation While DiMoT optimizes the tokenizer, it is not limited to maximizing standard reconstruction accuracy. Instead, DiMoT enhances motion generation by improving the tokenizer's diffusability. It structures the latent space to be more easily modeled by the diffusion process. We evaluate this by showing that DiMoT improves text-conditioned generation performance while maintaining comparable reconstruction fidelity. To isolate the tokenizer's contribution, we keep the downstream diffusion backbone architecture and training configuration strictly fixed (See Appendix E).

Optimal Latent Spectral Profile High-frequency energy in the latent space is more challenging to suppress, as latent spaces lack the tractable ground truth for spectral alignment. While excessively high-frequency energy degrades diffusion generation, naively eliminating it also compromises sample quality [31]. Consequently, prior methods [32] rely on mathematical transformations (*e.g.*, downsampling) as anchors. In contrast, CSA provides a flexible anchoring mechanism that adaptively penalizes high-frequency components, exclusively for samples misaligned with the critic distribution.

4 Experiments

4.1 Datasets

We utilize InterHuman [54] and Inter-X [55] datasets for the evaluation of text-conditioned interaction generation performance. We also provide HumanML3D [56] results in Tab. 3, but this paper focuses on the dual-person task. This setting imposes stricter constraints that more effectively expose the temporal artifacts and high-frequency outliers our method addresses.

The InterHuman and InterX datasets contain 7,779 and 11,388 interaction sequences, respectively, where each sequence is illustrated with 3 textual annotations. InterHuman follows the SMPL [57] skeleton representation with 22 joints, including the root joint. Each interaction sequence at the i-th frame is represented by: $x^i = \left[\mathbf{j}_g^p, \mathbf{j}_g^v, \mathbf{j}^r, \mathbf{c}^f\right]$ which is the collection of joint positions $\mathbf{j}_g^p \in \mathbb{R}^{2\times22\times3}$, joint velocities $\mathbf{j}_g^v \in \mathbb{R}^{2\times22\times3}$, 6D representation rotations $\mathrm{j}^r \in \mathbb{R}^{2\times21\times6}$, and binary foot-ground contact features $\mathbf{c}^f \in \mathbb{R}^{2\times4}$, resulting in a total input dimension of 524. Inter-X follows the SMPL-X [58] skeleton representation, comprising 54 body and hand joints, accompanied by root orientation and translation. Each interaction sequence could be represented by $x^i = \left[\mathbf{r}_g^p, \mathbf{j}^r\right]$ comprising root joint positions $\mathbf{r}_g^p \in \mathbb{R}^{2\times3}$ and Euler angle representation rotations $\mathrm{j}^r \in \mathbb{R}^{2\times55\times3}$, for a total input dimension of 336.

We employ the evaluation metrics following previous studies [54, 56]. (See Appendix D for more details). The implementation of CSA, AMD, and diffusion backbone can be found in Appendix E.

4.2 Main Results

Table 1 shows DiMoT achieves state-of-the-art performance on the InterHuman [54] benchmark. Compared to the MLD++ [22] baseline, DiMoT improves Top-3 R-Precision from 0.688 to 0.739 and reduces FID by 6.5% (5.046 to 4.716). Furthermore, DiMoT consistently outperforms recent methods (e.g., DHVAE [26], TIMotion [62]) across all R-Precision tiers while establishing the lowest overall FID. Similar to DHVAE, these methods based on LDMs suffer from low MModality, while our DiMoT still shows better diversity over the MLD++, with the Diversity metric closely matching the ground truth. In the Inter-X evaluations, compared to the MLD++ baseline, DiMoT improves R-Precision (0.732 to 0.746) and achieves a substantial 24.9% reduction in FID (0.446 to 0.335). DiMoT still outperforms the DHVAE across R-Precision metrics, achieving the lowest FID and MM Dist. These results confirm DiMoT's generalizability to both SMPL [57] and SMPL-X [58] data. While suppressing high-frequency shortcuts with CSA slightly degrades reconstruction FID, the MPJPE improvement demonstrates enhanced kinematic accuracy.

Figure 3 qualitatively compares DiMoT with the MLD++ [65] baseline across diverse interactive text prompts. In the first scenario, MLD++ fails to capture the interaction semantics with interpenetration. Conversely, DiMoT synthesizes the first agent's sprinting approach and the second's synchronized dismissing gesture. For the dancing prompt, MLD++ yields generic, rigid movements that ignore the target style. DiMoT instead generates expressive, coordinated poses, capturing complex kinematics

Table 1: Quantitative evaluation on the InterHuman [54] and InterX test sets. $\pm$ indicates a 95% confidence interval (10 runs) and $\rightarrow$ means the closer to ground truth the better. The deep blue background indicates the best, while the light blue background indicates the second-best. '†' denotes self-reproduced results.

Dataset	Method	R Precision↑			FID↓	MM Dist↓	Diversity→	MModality↑
		Top 1	Top 2	Top 3				
Inter Human	Ground Truth	$0.452^{\pm0.008}$	$0.610^{\pm0.009}$	$0.701^{\pm0.008}$	$0.273^{\pm0.007}$	$3.755^{\pm0.008}$	$7.948^{\pm0.064}$	-
	TEMOS [59]	$0.224^{\pm0.010}$	$0.316^{\pm0.013}$	$0.450^{\pm0.018}$	$17.375^{\pm0.043}$	$6.342^{\pm0.015}$	$6.939^{\pm0.071}$	$0.535^{\pm0.014}$
	T2M [56]	$0.238^{\pm0.012}$	$0.325^{\pm0.010}$	$0.464^{\pm0.014}$	$13.769^{\pm0.072}$	$5.731^{\pm0.013}$	$7.046^{\pm0.022}$	$1.387^{\pm0.076}$
	MDM [60]	$0.153^{\pm0.012}$	$0.260^{\pm0.009}$	$0.339^{\pm0.012}$	$9.167^{\pm0.056}$	$7.125^{\pm0.018}$	$7.602^{\pm0.045}$	$2.350^{\pm0.080}$
	ComMDM [61]	$0.223^{\pm0.009}$	$0.334^{\pm0.008}$	$0.466^{\pm0.010}$	$7.069^{\pm0.054}$	$6.212^{\pm0.021}$	$7.244^{\pm0.038}$	$1.822^{\pm0.052}$
	InterGen [54]	$0.371^{\pm0.010}$	$0.515^{\pm0.012}$	$0.624^{\pm0.010}$	$5.918^{\pm0.079}$	$5.108^{\pm0.014}$	$7.387^{\pm0.029}$	$2.141^{\pm0.063}$
	TIMotion [62]	$0.491^{\pm0.005}$	$0.648^{\pm0.004}$	$0.724^{\pm0.004}$	$5.433^{\pm0.080}$	$3.775^{\pm0.001}$	$8.032^{\pm0.030}$	$0.952^{\pm0.032}$
	in2IN [63]	$0.425^{\pm0.008}$	$0.576^{\pm0.008}$	$0.662^{\pm0.009}$	$5.535^{\pm0.120}$	$3.803^{\pm0.002}$	$7.953^{\pm0.047}$	$1.215^{\pm0.023}$
	InterMask [64]	$0.449^{\pm0.004}$	$0.599^{\pm0.005}$	$0.683^{\pm0.004}$	$5.154^{\pm0.061}$	$3.790^{\pm0.002}$	$7.944^{\pm0.033}$	$1.737^{\pm0.020}$
	DHVAE [26]	$0.496^{\pm0.004}$	$0.647^{\pm0.006}$	$0.720^{\pm0.005}$	$5.015^{\pm0.085}$	$3.772^{\pm0.002}$	$7.952^{\pm0.045}$	$0.804^{\pm0.030}$
	MLD++† [65]	$0.412^{\pm0.007}$	$0.594^{\pm0.003}$	$0.688^{\pm0.005}$	$5.046^{\pm0.084}$	$3.787^{\pm0.001}$	$8.084^{\pm0.023}$	$0.613^{\pm0.028}$
	DiMoT	$0.519^{\pm0.004}$	$0.663^{\pm0.008}$	$0.739^{\pm0.006}$	$4.716^{\pm0.080}$	$3.775^{\pm0.001}$	$7.970^{\pm0.035}$	$0.650^{\pm0.013}$
InterX	Ground Truth	$0.429^{\pm0.004}$	$0.626^{\pm0.003}$	$0.736^{\pm0.003}$	$0.002^{\pm0.002}$	$3.536^{\pm0.013}$	$9.734^{\pm0.078}$	-
	TEMOS [59]	$0.092^{\pm0.003}$	$0.171^{\pm0.003}$	$0.238^{\pm0.002}$	$29.258^{\pm0.069}$	$6.867^{\pm0.013}$	$4.738^{\pm0.078}$	$0.672^{\pm0.041}$
	T2M [56]	$0.184^{\pm0.010}$	$0.298^{\pm0.006}$	$0.396^{\pm0.005}$	$5.481^{\pm0.382}$	$9.576^{\pm0.006}$	$5.771^{\pm0.151}$	$2.761^{\pm0.042}$
	MDM [60]	$0.203^{\pm0.009}$	$0.329^{\pm0.007}$	$0.426^{\pm0.005}$	$23.701^{\pm0.057}$	$9.548^{\pm0.014}$	$5.856^{\pm0.077}$	$3.490^{\pm0.061}$
	MDM(GRU) [60]	$0.179^{\pm0.006}$	$0.299^{\pm0.005}$	$0.387^{\pm0.007}$	$32.617^{\pm0.122}$	$9.557^{\pm0.019}$	$7.003^{\pm0.134}$	$3.430^{\pm0.035}$
	ComMDM [61]	$0.090^{\pm0.002}$	$0.165^{\pm0.004}$	$0.236^{\pm0.004}$	$29.266^{\pm0.067}$	$6.870^{\pm0.017}$	$4.734^{\pm0.067}$	$0.771^{\pm0.053}$
	InterGen [54]	$0.207^{\pm0.004}$	$0.335^{\pm0.005}$	$0.429^{\pm0.005}$	$5.207^{\pm0.216}$	$9.580^{\pm0.011}$	$7.788^{\pm0.208}$	$3.686^{\pm0.052}$
	InterMask [64]	$0.403^{\pm0.005}$	$0.595^{\pm0.004}$	$0.705^{\pm0.005}$	$0.399^{\pm0.013}$	$3.705^{\pm0.017}$	$9.046^{\pm0.073}$	$2.261^{\pm0.081}$
	DHVAE [26]	$0.442^{\pm0.005}$	$0.638^{\pm0.005}$	$0.745^{\pm0.004}$	$0.339^{\pm0.015}$	$3.604^{\pm0.020}$	$9.378^{\pm0.065}$	$2.505^{\pm0.047}$
	MLD++† [65]	$0.433^{\pm0.005}$	$0.625^{\pm0.007}$	$0.732^{\pm0.006}$	$0.446^{\pm0.017}$	$3.659^{\pm0.028}$	$4.664^{\pm0.062}$	$1.611^{\pm0.115}$
	DiMoT	$0.450^{\pm0.007}$	$0.642^{\pm0.004}$	$0.746^{\pm0.004}$	$0.335^{\pm0.013}$	$3.562^{\pm0.030}$	$4.648^{\pm0.060}$	$2.427^{\pm0.112}$

Table 2: Ablation study of DiMoT components. Results are from the InterHuman [54] dataset.

Method	Tokenizer			Diffusion		
	rFID ↓	rR-Top3 ↑	MPJPE ↓	FID ↓	R-Top3 ↑	Foot Sliding ↓
MLD [21]	0.317	0.629	71.10	5.477	0.636	0.0209
GMM [44]	0.183	0.651	58.15	5.164	0.665	0.0191
MLD++ [65]	0.080	0.686	49.26	5.046	0.688	0.0186
MLD [21] w. CSA	0.315	0.661	67.70	5.322	0.689	0.0203
MLD [21] w. AMD	0.222	0.656	59.34	5.209	0.662	0.0194
MLD [21] w. DiMoT	**0.193** (-0.124)	**0.663** (+0.034)	**53.90** (-17.20)	**5.245** (-0.232)	**0.685** (+0.049)	**0.0197** (-0.0012)
MLD++ [65] w. CSA	0.086	0.713	47.18	4.967	0.746	0.0182
MLD++ [65] w. AMD	0.077	0.672	46.76	4.781	0.716	0.0175
MLD++ [65] w. DiMoT	**0.071** (-0.009)	**0.698** (+0.012)	**41.36** (-7.90)	**4.716** (-0.330)	**0.739** (+0.051)	**0.0168** (-0.0018)

like "stomping feet" and "swaying hips". In the "bumping" scenario, MLD++ poorly models the interaction, yielding unnatural spatial proximity. DiMoT models spatial boundaries and momentum strictly with natural post-contact reactions. These visualization results demonstrate that DiMoT mitigates temporal artifacts, showing smooth and semantically accurate multi-person motions. Furthermore, Appendix H reports a user study comparing DiMoT with InterMask.

4.3 Ablation Study

Table 2 ablates the DiMoT components across Latent Diffusion Model (LDM) baselines (MLD [21], MLD++ [65]). Integrating DiMoT consistently improves generation quality independent of the base architecture. By applying our tokenizer design to MLD++, DiMoT achieves the best generative performance with minimum temporal errors (Foot Sliding: 0.0168). DiMoT improves the diffusability with a superior trade-off between reconstruction and generation. The tokenizer maintains its fidelity with stronger MPJPE results while maximizing motion fidelity and text alignment.

Figure 4(a) compares different regularization methods applied to the tokenizer encoder. "No Critic" refers to the encoder that relies solely on a reconstruction loss, which evaluates the effectiveness of the reconstruction objective itself. CSA achieves optimal performance (R-Top3: 0.74) when fine-tuned with an additional adapter, which is further enhanced by spectral reweighting. Without an

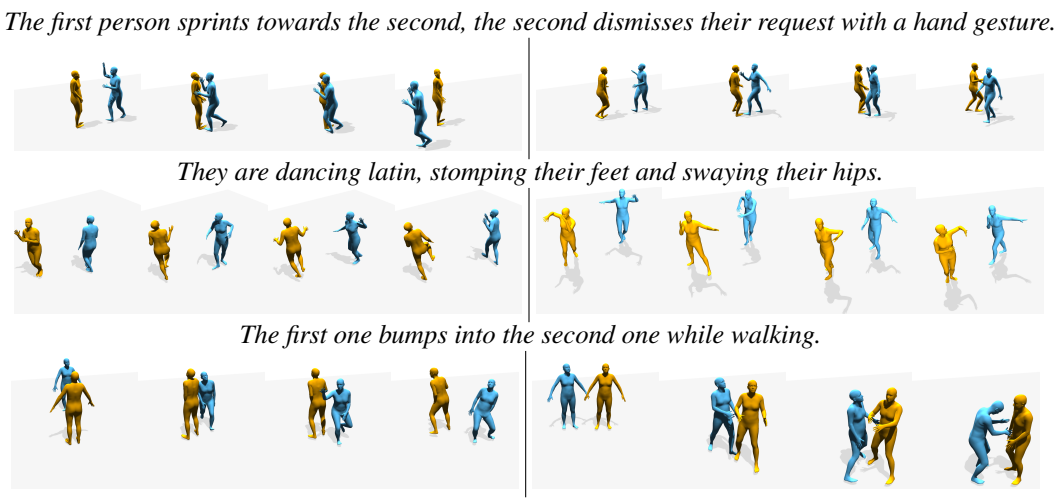
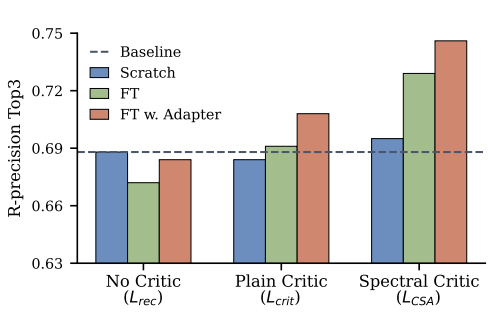
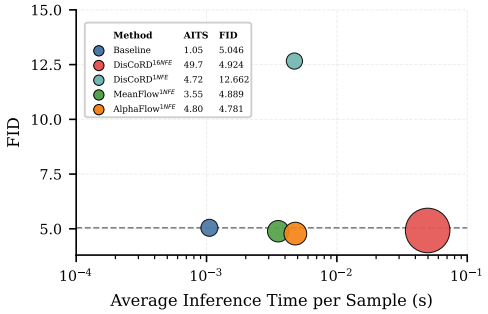


Figure 3: Qualitative comparison [66] (zoom in to see it better) between baseline MLD++ [22] and DiMoT, highlighting DiMoT's superior generation quality. (See the video in supplementary material.)



(a) R-precision results for different encoder designs. FT refers to fine-tuning without extra parameters.



(b) FID results for different decoder designs. We use the circle size to represent the decoder FLOPs.

Figure 4: Comparison of different design choices with DiMoT on the InterHuman dataset.

adapter, CSA functions as a parameter-free objective applied to the standard MLD++ [65] diffusion backbone, yet achieves better text-motion alignment. More spectral reweighting results are included in Appendix G.

Figure 4(b) evaluates various decoder generative schedulers in terms of latency and FID. We also highlight the inference step and FLOPs with the superscript and circle size, respectively. DisCoRD$^{16\text{NFE}}$ achieves better fidelity than the baseline but with excessive latency and FLOPs. In contrast, our AMD improves the generation fidelity with single-step efficiency, reducing computational overhead by orders of magnitude (See more results in Appendix F). Moreover, we compare linear and sigmoid curricula in Appendix G.1.

In Table 3, we also conduct an evaluation on the single-person benchmark (*i.e.*, HumanML3D [56]). By integrating DiMoT into MLD++ [65], it also improves the baseline performance. DiMoT dominates in fidelity with the lowest FID, and also shows gains in R-Top3 and MModality. These results further confirm that DiMoT's architectural benefits generalize to motion domains.

Table 3: Single-person results on HumanML3D.

Method	R-Top3↑	FID↓	MM Dist↓	MModality↑
GT	0.511	0.002	2.974	-
MLD++ [65]	0.544	0.049	2.828	1.672
MLD++ [65] w. CSA	0.552	0.047	2.808	1.715
MLD++ [65] w. AMD	0.542	0.042	2.842	1.675
MLD++ [65] w. DiMoT	0.550	0.038	2.809	1.689

5 Conclusion

In this paper, we address the generation-reconstruction trade-off in motion latent diffusion models by improving diffusability of the tokenizer, rather than scaling the diffusion backbone. We observe a spectral mismatch wherein both the latent and output profiles exhibit an abnormally high proportion of high-frequency components, potentially degrading text-to-motion generation. To address this, we propose the Diffusable Motion Tokenizer (DiMoT), consisting of Critic Spectral Adaptation (CSA) and Alpha-Flow Motion Decoder (AMD). Firstly, we demonstrate, both theoretically and empirically, that CSA regularizes the latent space via a diffusion-based spectral critic, suppressing high-frequency shortcuts without sacrificing reconstruction. Concurrently, AMD replaces deterministic decoding with a one-step average-velocity prediction, selectively preserving the essential dynamics required for temporal realism. Evaluations on standard motion benchmarks demonstrate that DiMoT successfully balances generation fidelity and reconstruction accuracy, validating our tokenizer refinement strategy.

Limitation DiMoT addresses spectral mismatch in text-to-motion generation, but defining the optimal spectral profile of the latent space remains an open problem, where CSA relies on using the diffusion as a critic with empirical choices. Moreover, motion datasets are limited in scale and use heterogeneous representations. Our evaluations demonstrate improvements within each dataset, but do not establish whether a tokenizer adapted on one dataset retains its performance on another. Future work will explore reinforcement or adversarial learning to replace the polynomial spectral weight, bypassing these rigid priors and optimality assumptions.

Acknowledgments

This work was supported by the K-CSC funding. The authors acknowledge the use of King's CREATE HPC. Retrieved November 05, 2026, from `https://doi.org/10.18742/rnvf-m076`.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[2] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[3] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[4] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[5] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.

[6] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[7] Wei-Jin Huang, Yue-Yi Zhang, Yi-Lin Wei, Zhi-Wei Xia, Juantao Tan, Yuan-Ming Li, Zhilin Zhao, and Wei-Shi Zheng. Learning whole-body human-humanoid interaction from human-human demonstrations. *arXiv preprint arXiv:2601.09518*, 2026.

[8] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[9] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

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

[11] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4D: Reconstructing and tracking humans with transformers. In *ICCV*, 2023.

[12] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. Avatarposer: Articulated full-body pose tracking from sparse motion sensing. In *European conference on computer vision*, pages 443–460. Springer, 2022.

[13] Yuming Du, Robin Kips, Albert Pumarola, Sebastian Starke, Ali Thabet, and Artsiom Sanakoyeu. Avatars grow legs: Generating smooth human motion from sparse tracking inputs with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2023.

[14] Jiaman Li, Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17142–17151, 2023.

[15] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[17] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

[18] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[19] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[20] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[21] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023.

[22] Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. *arXiv preprint arXiv:2404.19759*, 2024.

[23] Lei Jiang, Ye Wei, and Hao Ni. Motionpcm: Real-time motion synthesis with phased consistency model. *arXiv preprint arXiv:2501.19083*, 2025.

[24] Peng Jin, Yang Wu, Yanbo Fan, Zhongqian Sun, Wei Yang, and Li Yuan. Act as you wish: Fine-grained control of motion diffusion model with hierarchical semantic graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

[25] Longbin Ji, Pengfei Wei, Yi Ren, Jinglin Liu, Chen Zhang, and Xiang Yin. C2g2: Controllable co-speech gesture generation with latent diffusion model. *arXiv preprint arXiv:2308.15016*, 2023.

[26] Zichen Geng, Zeeshan Hayder, Bo Miao, Jian Liu, Wei Liu, and Ajmal Mian. Disentangled hierarchical vae for 3d human-human interaction generation. *arXiv preprint arXiv:2603.00144*, 2026.

[27] Guanhe Huang and Oya Celiktutan. Unified number-free text-to-motion generation via flow matching. *arXiv preprint arXiv:2603.27040*, 2026.

[28] Boyuan Li, Xihua Wang, Ruihua Song, and Wenbing Huang. Two-in-one: Unified multi-person interactive motion generation by latent diffusion transformer. *arXiv preprint arXiv:2412.16670*, 2024.

[29] Jungbin Cho, Junwan Kim, Jisoo Kim, Minseo Kim, Mingu Kang, Sungeun Hong, Tae-Hyun Oh, and Youngjae Yu. Discord: Discrete tokens to continuous motion via rectified flow decoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14602–14612, 2025.

[30] Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15703–15712, 2025.

[31] Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025.

[32] Shizhan Liu, Xinran Deng, Zhuoyi Yang, Jiayan Teng, Xiaotao Gu, and Jie Tang. Delving into latent spectral biasing of video vaes for superior diffusability. *arXiv preprint arXiv:2512.05394*, 2025.

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[34] Hanyu Wang, Saksham Suri, Yixuan Ren, Hao Chen, and Abhinav Shrivastava. Larp: Tokenizing videos with a learned autoregressive generative prior. *arXiv preprint arXiv:2410.21264*, 2024.

[35] Chenyang Gu, Mingyuan Zhang, Haozhe Xie, Zhongang Cai, Lei Yang, and Ziwei Liu. Bridging semantic and kinematic conditions with diffusion-based discrete motion tokenizer. *arXiv preprint arXiv:2603.19227*, 2026.

[36] Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J Zico Kolter, and Kaiming He. Improved mean flows: On the challenges of fastforward generative models. *arXiv preprint arXiv:2512.02012*, 2025.

[37] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

[38] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning of latent diffusion transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18262–18272, 2025.

[39] Yinbo Chen, Rohit Girdhar, Xiaolong Wang, Sai Saketh Rambhatla, and Ishan Misra. Diffusion autoencoders are scalable image tokenizers. *arXiv preprint arXiv:2501.18593*, 2025.

[40] Mang Ning, Mingxiao Li, Le Zhang, Lanmiao Liu, Matthew B Blaschko, Albert Ali Salah, and Itir Onal Ertugrul. Spectrum matching: a unified perspective for superior diffusability in latent diffusion. *arXiv preprint arXiv:2603.14645*, 2026.

[41] Long Zhao, Sanghyun Woo, Ziyu Wan, Yandong Li, Han Zhang, Boqing Gong, Hartwig Adam, Xuhui Jia, and Ting Liu. Epsilon-vae: Denoising as visual decoding. *arXiv preprint arXiv:2410.04081*, 2024.

[42] Bolin Lai, Xudong Wang, Saketh Rambhatla, James M Rehg, Zsolt Kira, Rohit Girdhar, and Ishan Misra. Toward diffusible high-dimensional latent spaces: A frequency perspective. *arXiv preprint arXiv:2511.22249*, 2025.

[43] Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit H Bermano, and Michiel van de Panne. Closd: Closing the loop between simulation and diffusion for multi-task character control. *arXiv preprint arXiv:2410.03441*, 2024.

[44] Linnan Tu, Lingwei Meng, Zongyi Li, Hefei Ling, and Shijuan Huang. Autoregressive motion generation with gaussian mixture-guided latent sampling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[45] Yaoqin Ye, Yiteng Xu, Qin Sun, Xinge Zhu, Yujing Sun, and Yuexin Ma. Remogen: Real-time human interaction-to-reaction generation via modular learning from diverse data. *arXiv preprint arXiv:2604.01082*, 2026.

[46] Liang Pan, Zeshi Yang, Zhiyang Dou, Wenjia Wang, Buzhen Huang, Bo Dai, Taku Komura, and Jingbo Wang. Tokenhsi: Unified synthesis of physical human-scene interactions through task tokenization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5379–5391, 2025.

[47] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[48] Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International conference on artificial intelligence and statistics*, pages 1540–1552. PMLR, 2020.

[49] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[50] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[52] Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. Alphaflow: Understanding and improving meanflow models. *arXiv preprint arXiv:2510.20771*, 2025.

[53] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[54] Han Liang, Wenqian Zhang, Wenxuan Li, Jingyi Yu, and Lan Xu. Intergen: Diffusion-based multi-human motion generation under complex interactions. *International Journal of Computer Vision*, pages 1–21, 2024.

[55] Liang Xu, Xintao Lv, Yichao Yan, Xin Jin, Shuwen Wu, Congsheng Xu, Yifan Liu, Yizhou Zhou, Fengyun Rao, Xingdong Sheng, et al. Inter-x: Towards versatile human-human interaction analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22260–22271, 2024.

[56] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5152–5161, 2022.

[57] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

[58] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

[59] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. *arXiv preprint arXiv:2204.14109*, 2022.

[60] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

[61] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *arXiv preprint arXiv:2303.01418*, 2023.

[62] Yabiao Wang, Shuo Wang, Jiangning Zhang, Ke Fan, Jiafu Wu, Zhucun Xue, and Yong Liu. Timotion: Temporal and interactive framework for efficient human-human motion generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7169–7178, 2025.

[63] Pablo Ruiz Ponce, German Barquero, Cristina Palmero, Sergio Escalera, and Jose Garcia-Rodriguez. in2in: Leveraging individual information to generate human interactions. *arXiv preprint arXiv:2404.09988*, 2024.

[64] Muhammad Gohar Javed, Chuan Guo, Li Cheng, and Xingyu Li. Intermask: 3d human interaction generation via collaborative masked modelling. *arXiv preprint arXiv:2410.10010*, 2024.

[65] Wenxun Dai, Ling-Hao Chen, Yufei Huo, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. Real-time controllable motion generation via latent consistency model.

[66] Manuel Kaufmann, Velko Vechev, and Dario Mylonopoulos. aitviewer, 7 2022. URL `https://github.com/eth-ait/aitviewer`.

[67] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[68] Bao Pham, Gabriel Raya, Matteo Negri, Mohammed J Zaki, Luca Ambrogioni, and Dmitry Krotov. Memorization to generalization: Emergence of diffusion models from associative memory. *arXiv preprint arXiv:2505.21777*, 2025.

[69] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[70] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

A Full theoretical development of CSA

A.1 Notation

Symbol	Meaning
$\mathcal{X}, \mathcal{Z}$	data and latent spaces, $\mathcal{Z} = \mathbb{R}^d$ with $d = C' \times N$, $N = n_1 \cdots n_D$
E_φ, D_{ψ_0}	encoder (with parameters φ) and decoder pinned at its pre-trained parameters ψ_0
p_{data}	data distribution
p_z^φ	induced latent distribution $(E_\varphi)_\# p_{\text{data}}$
f_θ, θ_0	velocity model (rectified flow) and its frozen pre-trained parameters
$\text{sg}(\cdot)$	stop-gradient (identity on values, zero on gradients)
x_t, ϵ, t	rectified-flow interpolant, noise, time
y, p_y, $p_{z\|y}^\varphi$	text prompt, text marginal, conditional latent law $(E_\varphi)_\# p(x\|y)$
$r(z)$	prediction residual $\text{sg}(v_{\text{pred}}) - (\epsilon - z)$, with $v_{\text{pred}} := f_{\theta_0}(x_t, t, y)$
$\mathcal{L}_{\text{crit}}$	plain critic loss
$\mathcal{L}_{\text{CSA}}$	static spectrally weighted CSA loss
$\mathcal{L}_{\text{rec}}$	reconstruction loss
$\mathcal{L}_{\text{diff}}(\theta; \varphi)$	diffusion training loss for encoder φ
$\mathcal{F}, \mathcal{F}^*$	unitary D-D Discrete Fourier Transform (DFT) and its conjugate transpose
$d(k)$	normalized Euclidean FFT distance from DC, $d(k) \in [0, 1]$
$w(k), \tilde{w}(k), \bar{w}$	static spectral weight, mean-1-normalized weight, mean of w
κ, p	spectral hyperparameters: HF strength and distance exponent
λ_c	weight on the spectral critic loss
M_w	spectral mass operator $\mathcal{F}^* \,\text{diag}(\tilde{w})\, \mathcal{F}$
N	total spatial DFT modes $n_1 \cdots n_D$ ($N = T$ for the 1D motion setting of Section 3.1)
H	$\mathcal{L}_{\text{rec}}$ Hessian at the baseline φ_0
g	stop-gradient pseudo-gradient of $\mathcal{L}_{\text{CSA}}$ at φ_0 (Assumption 18)
g_{crit}	true gradient $\nabla_\varphi \mathcal{L}_{\text{diff}}(\theta_0; \varphi_0)$ at the baseline (not the stop-gradient pseudo-gradient)
$\text{S}(k)$	per-mode power spectral density of the latent process
$\beta_{\text{eff}}(k, t)$	per-mode inverse effective temperature (Proposition 29)

A.2 Setup

Definition 2 (Stop-gradient operator)**.** The stop-gradient $\text{sg} : \mathbb{R}^N \to \mathbb{R}^N$ is the identity map on values: $\text{sg}(z) = z$ as elements of $\mathbb{R}^N$. In any expression where $\text{sg}(z(\varphi))$ appears, gradients with respect to φ treat $\text{sg}(z(\varphi))$ as a constant. Equivalently, introduce a frozen copy $\bar{z} = z$ with $\partial \bar{z} / \partial \varphi \equiv 0$. Replace $\text{sg}(z(\varphi))$ by $\bar{z}$ in every gradient computation. Operationally, in PyTorch this is `z.detach()`.

Let $\mathcal{X} = \mathbb{R}^{C \times H \times W}$ and $\mathcal{Z} = \mathbb{R}^d$ be the data and latent spaces. Set $d = C' \times n_1 \times \cdots \times n_D$ with $D \in \{1, 2, 3\}$ for motion, image, and video respectively. An autoencoder is a pair (E_φ, D_{ψ_0}) with text-agnostic encoder $E_\varphi(x)$ and a frozen pre-trained decoder D_{ψ_0}. Data is sampled jointly $(x, y) \sim p_{\text{data}}$ where y is the text prompt, with text marginal p_y. The conditional latent law is $p_{z|y}^\varphi := (E_\varphi)_\# p(x|y)$, and the marginal latent law is $p_z^\varphi = (E_\varphi)_\# p_{\text{data}}$. The rectified-flow interpolant [67] is $x_t = (1 - t)z_0 + t\epsilon$ with velocity $v = \epsilon - z_0$. A text-conditioned velocity model $f_\theta(x_t, t, y)$ is trained by

$$\mathcal{L}_{\text{diff}}(\theta; \varphi) = \mathbb{E}_{(x,y),\, t \sim \mathcal{U}[0,1],\, \epsilon \sim \mathcal{N}(0,I)} \left\| f_\theta((1 - t)E_\varphi(x) + t\epsilon,\, t,\, y) - (\epsilon - E_\varphi(x)) \right\|^2 . \quad (8)$$

Definition 3 (Plain critic loss)**.** For $z = E_\varphi(x)$, sample (t, ϵ) as above, set $x_t = (1 - t)\,\text{sg}(z) + t\epsilon$, and define the prediction residual $r(z) = \text{sg}(f_\theta(x_t, t, y)) - (\epsilon - z)$. The plain critic loss is

$$\mathcal{L}_{\text{crit}}(\varphi; \theta) = \mathbb{E}_{(x,y),t,\epsilon} \left\| r(z) \right\|^2 . \quad (9)$$

Lemma 4 (Value equivalence (values, not gradients))**.** *For every φ and every frozen θ, $\mathcal{L}_{\text{crit}}(\varphi; \theta) = \mathcal{L}_{\text{diff}}(\theta; \varphi)$ as values. Their gradients in φ differ under the stop-gradients (Remark 5, Assumption 18).*

Proof. sg preserves values, so dropping sg from (9) gives

$$\mathbb{E}_{(x,y),t,\epsilon} \left\| f_\theta\big((1 - t)E_\varphi(x) + t\epsilon,\, t,\, y\big) - \epsilon + E_\varphi(x) \right\|^2 ,$$

which matches (8) with $z_0 = E_\varphi(x)$. Equivalently, by the tower property $\mathbb{E}_{(x,y)} = \mathbb{E}_y \mathbb{E}_{x|y}$, the same quantity reads $\mathbb{E}_{y\sim p_y} \mathbb{E}_{z_0 \sim p^{\varphi}_{z|y},\, t,\, \epsilon} \left\| f_\theta(x_t, t, y) - (\epsilon - z_0) \right\|^2$, the form used in conditional diffusion training. □

Remark 5 (Stop-gradients: value equivalence and necessity of both stops). $\mathcal{L}_{\mathrm{crit}}$ and $\mathcal{L}_{\mathrm{diff}}$ agree as values (Lemma 4). Their gradients in φ differ under the stop-gradients. Autograd of $\mathcal{L}_{\mathrm{crit}}$ drops the Jacobian-of-f_{θ_0} contribution that $\nabla_\varphi \mathcal{L}_{\mathrm{diff}}$ retains. The pseudo-gradient g (Assumption 18) is the autograd direction. The true $g_{\mathrm{crit}} = \nabla_\varphi \mathcal{L}_{\mathrm{diff}}(\theta_0; \varphi_0)$ is the Taylor coefficient in Theorem 25.

Both stops are required. Removing sg from v_{pred} adds a $J^\top_{f_{\theta_0}} J_{f_{\theta_0}}$ term to the Hessian and breaks Theorem 10. Removing sg from x_t instead pushes the encoder to reshape the noised sample itself, a different objective.

A.3 FFT distance grid and CSA loss

Definition 6 (Normalized FFT distance grid). For spatial sizes $(n_1, \ldots, n_D)$ let $f^{(\ell)}_{k_\ell} = k_\ell / n_\ell \pmod 1 \in [-1/2, 1/2)$ in FFT layout. Define

$$d(k_1, \ldots, k_D) = \Big(\textstyle\sum_\ell (f^{(\ell)}_{k_\ell})^2 \Big)^{1/2} / d_{\max}, \qquad d_{\max} = \max_k \Big(\textstyle\sum_\ell (f^{(\ell)}_{k_\ell})^2 \Big)^{1/2}.$$

Then $d(0) = 0$ at DC, and $\max_k d(k) = 1$ by normalization. The maximum is attained at the Nyquist corner $k_\ell = n_\ell/2$ when all n_ℓ are even. Moreover, $d(\bar{k}) = d(k)$ where $\bar{k}_\ell \coloneqq n_\ell - k_\ell$ is the per-axis reflection, since $f^{(\ell)}_{n_\ell - k_\ell} \equiv -f^{(\ell)}_{k_\ell} \pmod 1$ in the FFT layout.

Definition 7 (CSA loss). With $w(k) = 1 + \kappa\, d(k)^p$ and $\tilde{w}(k) = w(k)/\bar{w}$, $\bar{w} = N^{-1} \sum_k w(k)$, and $\mathcal{F}$ the unitary D-D DFT acting on the spatial axes,

$$\mathcal{L}_{\mathrm{CSA}}(\varphi; \theta, w) = \mathbb{E}_{(x,y),t,\epsilon} \Big[\tfrac{1}{N} \sum_k \tilde{w}(k) \big| (\mathcal{F}\, r)(k) \big|^2 \Big].$$

Here $\big|(\mathcal{F}\, r)(k)\big|^2 = \sum_c \big|(\mathcal{F}\, r)(k, c)\big|^2$ sums the squared magnitude over channels at each spatial mode k, and $N = \prod_\ell n_\ell$ is the number of spatial modes.

Lemma 8 (Parseval recovery). *With $\kappa = 0$, so $w \equiv \tilde{w} \equiv 1$, $\mathcal{L}_{\mathrm{CSA}}(\varphi; \theta, 1) = \mathcal{L}_{\mathrm{crit}}(\varphi; \theta)/N$.*

Proof. Plancherel for the unitary DFT gives $\sum_k |(\mathcal{F}\, u)(k)|^2 = \sum_i u_i^2$ for any $u \in \mathbb{R}^N$. Apply per channel to $u = r$. □

A.4 Pointwise gradient, Hessian, strong convexity

Proposition 9 (Pointwise gradient). *Fix (x, y, t, ϵ) and let $c = \mathrm{sg}(f_\theta(x_t, t, y))$, constant in z under stop-gradient. Let $M_w = \mathcal{F}^* \operatorname{diag}(\tilde{w})\, \mathcal{F} \in \mathbb{R}^{N \times N}$. Then $M_w^\top = M_w$, M_w maps $\mathbb{R}^N$ to $\mathbb{R}^N$, and*

(i) $\ell_w(z) \coloneqq \frac{1}{N} \sum_k \tilde{w}(k) |(\mathcal{F}\, r(z))(k)|^2 = \frac{1}{N}\, r(z)^\top M_w r(z)$, *with* $r(z) = c - \epsilon + z$.
(ii) $\nabla_z \ell_w = \frac{2}{N} M_w r(z) = \frac{2}{N} M_w (z - (\epsilon - c))$.
(iii) $\nabla_z^2 \ell_w = \frac{2}{N} M_w$, *constant in* z.
(iv) $\nabla_\varphi \ell_w = \frac{2}{N} J_{E_\varphi}(x)^\top M_w r(z)$.

Proof. $\operatorname{diag}(\tilde{w})$ is real-diagonal. By Definition 6, $d(\bar{k}) = d(k)$, so $\tilde{w}(\bar{k}) = \tilde{w}(k)$ and $\tilde{w}$ is conjugate-symmetric in the FFT layout. For $u \in \mathbb{R}^N$, $\hat{u} \coloneqq \mathcal{F}\, u$ satisfies the Nyquist symmetry $\hat{u}(\bar{k}) = \overline{\hat{u}(k)}$. Then $\operatorname{diag}(\tilde{w})\hat{u}$ inherits the symmetry and its inverse DFT is real, so $M_w u \in \mathbb{R}^N$. The unitary conjugation $M_w = \mathcal{F}^* \operatorname{diag}(\tilde{w})\, \mathcal{F}$ of a real diagonal is Hermitian on $\mathbb{C}^N$. Restricting to $\mathbb{R}^N$ gives $M_w^\top = M_w$ with spectrum $\{\tilde{w}(k)\}_k$. For the multi-channel latent $r \in \mathbb{R}^{C' \times N}$, the same operator applies along each channel with overall Hessian $I_{C'} \otimes M_w$.

For (i), $\sum_k \tilde{w}(k) |(\mathcal{F}\, r)(k)|^2 = \langle \mathcal{F}\, r, \operatorname{diag}(\tilde{w})\, \mathcal{F}\, r \rangle = r^\top M_w r$ by unitarity. For (ii) and (iii), $\partial r / \partial z = I$ and M_w is constant, so $\nabla_z (r^\top M_w r)/N = 2 M_w r / N$ and $\nabla_z^2 \ell_w = 2 M_w / N$. For (iv), apply the chain rule at $z = E_\varphi(x)$. □

Theorem 10 (Strong convexity of the detached surrogate in z)**.** *Let $L_w(z) = \mathbb{E}_{t,\epsilon}[\ell_w(z)]$ at fixed (x, y), where ℓ_w is the pointwise loss of Proposition 9 with $c = \mathrm{sg}(f_\theta(x_t, t, y))$ frozen via Definition 2. Then $\nabla_z^2 L_w = 2M_w/N$, and L_w is μ-strongly convex and Λ-smooth in z with*

$$\mu = 2N^{-1} \min_k \tilde{w}(k), \qquad \Lambda = 2N^{-1} \max_k \tilde{w}(k).$$

For $w(k) = 1 + \kappa\, d(k)^p$ with $\kappa \geq 0$, $\tilde{w} > 0$, so $\mu > 0$ and the global minimizer z^* is unique. Gradient descent on L_w at fixed c with $\eta \leq 1/\Lambda$ contracts in the M_w-norm at rate $1 - \eta\mu$.

Remark 11 (Strong convexity is a frozen-c statement). Theorem 10 bounds the Hessian of the detached surrogate at fixed c. Real training re-evaluates f_{θ_0} at each step, so c moves with z and contributes a non-constant Jacobian term to the full Hessian. The contraction rate $1 - \eta\mu$ is therefore a property of the surrogate, not a guarantee on the full training dynamics. Training also acts on the encoder parameters φ, not on z directly. The z-space convexity at fixed (x, y, c) does not transfer to a corresponding property in φ.

Proof. Proposition 9(iii) gives a constant Hessian at every (x, y, t, ϵ), and linearity of expectation preserves it. M_w is the unitary conjugation of a positive diagonal, so its spectrum is exactly $\{\tilde{w}(k)\}_k$, giving μ and Λ. Strict positivity of $\tilde{w}$ gives a unique minimizer. The contraction statement is the standard rate for quadratic objectives with constant Hessian in the M_w-norm. □

Remark 12 (Fixed-point set is invariant under $\tilde{w}$). z^* minimizes L_w at fixed detached $c = \mathrm{sg}(v_{\mathrm{pred}})$ if and only if $\mathbb{E}_{t,\epsilon}[c - (\epsilon - z^*)] = 0$. This follows from setting $\nabla_z L_w = 0$ in Proposition 9(ii) and inverting M_w. The fixed-point set is therefore the same under any positive weighting. Spectral weights steer the dynamics toward this set but do not change which z^* are stationary. Plain critic and CSA share the fixed points. What differs is the gradient field at non-stationary z. The practical benefit of CSA therefore arises on the way to convergence, not from changing the asymptotic optimum.

Theorem 13 (Per-frequency selectivity)**.** *Let $\hat{r} = \mathcal{F}\, r$. The CSA gradient in the Fourier domain is*

$$(\mathcal{F}\, \nabla_z \ell_w)(k) = \tfrac{2}{N} \tilde{w}(k)\, \hat{r}(k),$$

and the per-mode magnitude is $2\tilde{w}(k)|\hat{r}(k)|/N$. The gradient at mode k vanishes if and only if $\tilde{w}(k) = 0$ or $\hat{r}(k) = 0$.

Proof. Apply $\mathcal{F}$ to Proposition 9(ii) and use unitarity $\mathcal{F}\,\mathcal{F}^* = I$, so $\mathcal{F}\, M_w = \mathrm{diag}(\tilde{w})\, \mathcal{F}$. □

A.5 Bayes-optimal sanity check

This subsection is stated for the marginal latent law $p_z^{\varphi_0}$ to keep the analysis self-contained. The conditional case follows by the same argument with $p_y \mathbb{E}_{x|y}$ replacing $\mathbb{E}_x$ and $\Delta(z, y)$ replacing $\Delta(z)$, but is not used by Theorem 1.

Remark 14 (Spring-force interpretation). The pointwise gradient $\nabla_z \ell_w = 2N^{-1} M_w (z - (\epsilon - v_{\mathrm{pred}}))$ acts as a spring pulling z toward $\epsilon - v_{\mathrm{pred}}$, the diffusion's implicit reconstruction of the clean code from x_t. Stiffness M_w upweights the modes selected by $\tilde{w}$. The fixed point in Remark 12 is the configuration where encoder outputs and diffusion denoising agree on every mode.

Proposition 15 (Bayes-optimal sanity check)**.** *Let $f^*(x_t, t) \coloneqq \mathbb{E}[\epsilon - z_0 \mid x_t, t]$ be the Bayes-optimal velocity predictor for $p_z^{\varphi_0}$. Define the population residual*

$$\Delta(z) \coloneqq \mathbb{E}_{t,\epsilon}[f^*((1-t)z + t\epsilon, t) - (\epsilon - z)],$$

and the per-time slice

$$\Delta_t(z) \coloneqq \mathbb{E}_{\epsilon}[f^*((1-t)z + t\epsilon, t) - (\epsilon - z)],$$

so $\Delta(z) = \mathbb{E}_t[\Delta_t(z)]$. Then $\mathbb{E}_{z \sim p_z^{\varphi_0}}[\Delta(z)] = 0$. Pointwise, $\Delta(z) \neq 0$ in general. For $z \in \mathrm{supp}(p_z)$ with $p_z \in C^2$ and $p_z(z) > 0$, $\Delta_t(z) \to 0$ as $t \to 0^+$ at rate $O(t)$, uniformly on compact subsets of $\mathrm{supp}(p_z)$.

The vanishing of $\mathbb{E}_z[\Delta(z)]$ is a sanity check on the residual: its mean over the training distribution is zero. It does not assert encoder-parameter stationarity. The actual encoder gradient at φ_0 is, by Proposition 9(iv),

$$\nabla_\varphi \mathcal{L}_{\mathrm{CSA}}(\varphi_0; \theta_0, w) = \tfrac{2}{N}\, \mathbb{E}_x \big[J_{E_{\varphi_0}}(x)^\top M_w\, \Delta(E_{\varphi_0}(x)) \big],$$

which need not equal zero even when $\mathbb{E}_z[\Delta(z)] = 0$, since the Jacobian-weighted expectation differs from the unweighted residual mean.

Proof. **Population identity.** Apply the tower property with respect to (x_t, t):

$$\mathbb{E}_{z,t,\epsilon}[f^*(x_t,t)] = \mathbb{E}_{z,t,\epsilon}[\mathbb{E}[\epsilon - z \mid x_t, t]] = \mathbb{E}_{z,t,\epsilon}[\epsilon - z].$$

This equals the right-hand side of $\mathbb{E}[\epsilon - z]$ inside Δ, so $\mathbb{E}_z[\Delta(z)] = 0$.

Pointwise nonvanishing. At fixed z and $t > 0$, $f^*(x_t, t) = \mathbb{E}[\epsilon' - z_0' \mid x_t, t]$ averages over all (z_0', ϵ') pairs whose interpolant lands at x_t, not just (z, ϵ). The posterior has nontrivial spread for $t > 0$, so $\Delta_t(z) \neq 0$ in general.

Slice-wise low-noise limit. Fix $z \in \mathrm{supp}(p_z)$ with $p_z \in C^2$ and $p_z(z) > 0$. The Bayes posterior is $p(z_0' \mid x_t, t) \propto p_z(z_0') \exp(-\|x_t - (1-t)z_0'\|^2/(2t^2))$, a Gaussian likelihood centered at $x_t/(1-t)$ with covariance $\Sigma_t = t^2/(1-t)^2 I$ tilted by the prior. The Laplace expansion at small $\|\Sigma_t\|$ gives

$$\mathbb{E}[z_0' \mid x_t, t] = \frac{x_t}{1-t} + \Sigma_t \nabla \log p_z\Big(\tfrac{x_t}{1-t}\Big) + O(\|\Sigma_t\|^2), \tag{10}$$

uniformly on compact $K \subset \mathrm{supp}(p_z)$ by $p_z \in C^2(K)$ and $\inf_K p_z > 0$. Substituting $x_t = (1-t)z + t\epsilon$, expanding $x_t/(1-t) = z + t\epsilon + t^2\epsilon + O(t^3)$ and $\Sigma_t = t^2 I + O(t^3)$, and Taylor of $\nabla \log p_z$ at z gives $\mathbb{E}[z_0' \mid x_t, t] = z + t\epsilon + t^2\epsilon + t^2 \nabla \log p_z(z) + O(t^3)$. Using $\epsilon' - z_0' = (x_t - z_0')/t$, $f^*(x_t, t) = -z - t\epsilon - t\nabla \log p_z(z) + O(t^2)$. Averaging over ϵ kills the $t\epsilon$ term, leaving

$$\Delta_t(z) = -t\, \nabla \log p_z(z) + O(t^2),$$

which is $O(t)$ and vanishes pointwise on $\mathrm{supp}(p_z)$ as $t \to 0^+$. For $z \notin \mathrm{supp}(p_z)$, $\Delta_t(z) \not\to 0$ in general. □

A.6 Information preservation and Pareto regime

Assumption 16. $\mathcal{L}_{\mathrm{rec}} \in C^3$ on a neighbourhood U of φ_0, with $\nabla_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi_0) = 0$ and Hessian $H \coloneqq \nabla^2_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi_0) \succeq \mu_{\mathrm{rec}} I \succ 0$.

Remark 17 (Reading $H \succ 0$ in over-parameterized networks). Strict positivity is a simplifying assumption. For over-parameterized encoders, $\mathcal{L}_{\mathrm{rec}}$ has flat directions from reparameterization symmetries (permutation of hidden units, weight-norm rescaling, and similar things). The analysis should be read on the corresponding quotient, equivalently as $H \succ 0$ restricted to the orthogonal complement of the symmetry-group tangent at φ_0.

Assumption 18. Both $\mathcal{L}_{\mathrm{CSA}}(\,\cdot\,; \theta_0, w)$ and $\mathcal{L}_{\mathrm{crit}}(\,\cdot\,; \theta_0)$ are C^2 on U. Let $g \coloneqq \nabla_\varphi \mathcal{L}_{\mathrm{CSA}}(\varphi_0; \theta_0, w)$ denote the stop-gradient pseudo-gradient. It is computed by autograd with v_{pred} treated as constant in φ. Assume $g \neq 0$.

Remark 19 (g is a vector field, not necessarily a scalar gradient). The pseudo-gradient g is a vector field on Φ. The implicit function theorem (IFT) framework does not require g to be the gradient of a scalar function. It only needs $F(\varphi, \lambda_c) = \nabla_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi) + \lambda_c \nabla_\varphi \mathcal{L}_{\mathrm{CSA}}(\varphi; \theta_0, w)$ to be C^1 with invertible $\partial_\varphi F$ at $(\varphi_0, 0)$. The trajectory $\varphi^*(\lambda_c)$ in Theorem 20 is well-defined regardless.

Theorem 20 (First-order Pareto improvement)**.** *Under Assumptions 16 and 18, let $\varphi^*(\lambda_c)$ be the local stationary point of $\nabla_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi) + \lambda_c \nabla_\varphi \mathcal{L}_{\mathrm{CSA}}(\varphi; \theta_0, w) = 0$ near φ_0. Throughout, g denotes the stop-gradient pseudo-gradient of Assumption 18. For all sufficiently small $\lambda_c > 0$:*

(i) $\varphi^*(\lambda_c) = \varphi_0 - \lambda_c H^{-1} g + O(\lambda_c^2)$.
(ii) $\mathcal{L}_{\mathrm{rec}}(\varphi^*) - \mathcal{L}_{\mathrm{rec}}(\varphi_0) = \frac{\lambda_c^2}{2} g^\top H^{-1} g + O(\lambda_c^3)$.
(iii) $g^\top(\varphi^* - \varphi_0) = -\lambda_c g^\top H^{-1} g + O(\lambda_c^2)$.

Since $g^\top H^{-1} g > 0$, the surrogate progress-to-cost ratio $|g^\top(\varphi^* - \varphi_0)|/\Delta\mathcal{L}_{\mathrm{rec}} = 2/\lambda_c + O(1) \to \infty$ as $\lambda_c \to 0^+$.

Proof. Define $F : \Phi \times \mathbb{R} \to \Phi$ by

$$F(\varphi, \lambda_c) \coloneqq \nabla_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi) + \lambda_c\, \nabla_\varphi \mathcal{L}_{\mathrm{CSA}}(\varphi; \theta_0, w).$$

F is C^1 jointly in (φ, λ_c) from $\mathcal{L}_{\mathrm{rec}} \in C^3$ (Assumption 16) and $\mathcal{L}_{\mathrm{CSA}} \in C^2$ (Assumption 18). Also $F(\varphi_0, 0) = 0$ and $\partial_\varphi F|_{(\varphi_0, 0)} = H \succ 0$ by Assumption 16 (read on the symmetry quotient as in Remark 17). The IFT then applies. It gives open $V \subset \mathbb{R}$ around 0 and $U' \subset U$ around φ_0. For every $\lambda_c \in V$, a unique $\varphi^*(\lambda_c) \in U'$ satisfies $F(\varphi^*(\lambda_c), \lambda_c) = 0$. The map $\lambda_c \mapsto \varphi^*(\lambda_c)$

is C^1. Implicit differentiation at $\lambda_c = 0$ gives $\mathrm{d}\varphi^*/\,\mathrm{d}\lambda_c|_0 = -H^{-1}g$, hence $\|\varphi^* - \varphi_0\| = O(\lambda_c)$. To upgrade the remainder to $O(\lambda_c^2)$, expand the gradients in the stationarity equation $\nabla_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi^*) + \lambda_c \nabla_\varphi \mathcal{L}_{\mathrm{CSA}}(\varphi^*; \theta_0, w) = 0$ via Taylor at φ_0. Using $\nabla_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi_0) = 0$, $\nabla^2_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi_0) = H$, $\mathcal{L}_{\mathrm{rec}} \in C^3$, and $\mathcal{L}_{\mathrm{CSA}} \in C^2$,

$$H(\varphi^* - \varphi_0) + \lambda_c g = O(\|\varphi^* - \varphi_0\|^2) + \lambda_c\, O(\|\varphi^* - \varphi_0\|) = O(\lambda_c^2),$$

where the last equality uses $\|\varphi^* - \varphi_0\| = O(\lambda_c)$. Inverting H,

$$\varphi^*(\lambda_c) = \varphi_0 - \lambda_c H^{-1} g + O(\lambda_c^2),$$

which is (i).

Taylor for $\mathcal{L}_{\mathrm{rec}}$. Using $\nabla_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi_0) = 0$, $\nabla^2_\varphi \mathcal{L}_{\mathrm{rec}}(\varphi_0) = H$, and $\mathcal{L}_{\mathrm{rec}} \in C^3$,

$$\mathcal{L}_{\mathrm{rec}}(\varphi^*) - \mathcal{L}_{\mathrm{rec}}(\varphi_0) = \tfrac{1}{2}(\varphi^* - \varphi_0)^\top H(\varphi^* - \varphi_0) + O(\|\varphi^* - \varphi_0\|^3).$$

Substituting $\varphi^* - \varphi_0 = -\lambda_c H^{-1} g + O(\lambda_c^2)$ and using $HH^{-1} = I$ on the quadratic term gives $(\varphi^* - \varphi_0)^\top H(\varphi^* - \varphi_0) = \lambda_c^2 g^\top H^{-1} g + O(\lambda_c^3)$, and $\|\varphi^* - \varphi_0\|^3 = O(\lambda_c^3)$. Combining, (ii) holds with

$$\left|\mathcal{L}_{\mathrm{rec}}(\varphi^*) - \mathcal{L}_{\mathrm{rec}}(\varphi_0) - \tfrac{\lambda_c^2}{2} g^\top H^{-1} g\right| \le C_{\mathrm{rec}}\, \lambda_c^3.$$

Surrogate pairing for (iii). Taking the inner product of $\varphi^* - \varphi_0 = -\lambda_c H^{-1} g + O(\lambda_c^2)$ with g gives (iii):

$$\left|g^\top(\varphi^* - \varphi_0) + \lambda_c\, g^\top H^{-1} g\right| \le C_{\mathrm{CSA}}\, \lambda_c^2.$$

The constant $C_{\mathrm{rec}} > 0$ depends on third derivatives of $\mathcal{L}_{\mathrm{rec}}$ on a closed ball $\bar{B}(\varphi_0, r) \subset U'$ that contains $\varphi^*(\lambda_c)$ for all $\lambda_c \in [0, \lambda_c^{\max}]$, and $C_{\mathrm{CSA}} > 0$ tracks the implicit constant in the trajectory expansion. $H \succ 0$ and $g \neq 0$ give $g^\top H^{-1} g > 0$, and the surrogate progress-to-cost ratio follows from dividing (iii) by (ii). □

Remark 21 (Reading of (iii)). Statement (iii) is the first-order pairing of the CSA direction g with the trajectory displacement $\varphi^* - \varphi_0$. This is the surrogate progress that training reduces by construction. The corresponding decrease in $\mathcal{L}_{\mathrm{crit}}$ is bridged in Theorem 25 through the alignment assumption Assumption 23. Lemma 4 then carries it over to $\mathcal{L}_{\mathrm{diff}}$.

A.7 Diffusion loss improvement

Assumption 22. $\theta_0 = \arg\min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi_0)$. The baseline diffusion model is optimal for the baseline encoder.

Assumption 23 (Critic-CSA gradient alignment)**.** Let $g \coloneqq \nabla_\varphi \mathcal{L}_{\mathrm{CSA}}(\varphi_0; \theta_0, w)$ be the stop-gradient pseudo-gradient that training optimizes. It is the gradient computed by autograd with $\mathrm{sg}(v_{\mathrm{pred}})$ treated as a constant in φ, as in Proposition 9. Let $g_{\mathrm{crit}} \coloneqq \nabla_\varphi \mathcal{L}_{\mathrm{diff}}(\theta_0; \varphi_0)$ be the true gradient of the conditional diffusion loss with respect to the encoder. We assume

$$g_{\mathrm{crit}}^\top H^{-1} g \;>\; 0.$$

The descent direction $-H^{-1}g$ in Theorem 20 comes from the stop-gradient pseudo-gradient. This is what training actually follows. The true gradient g_{crit} of the conditional diffusion loss is a different object. It includes a term involving the Jacobian of f_{θ_0} that the stop-gradient drops from the training signal. Assumption 23 says that this dropped term does not flip the sign at φ_0. So the descent direction used by training also descends the true $\mathcal{L}_{\mathrm{diff}}(\theta_0; \cdot)$ at φ_0. We treat Assumption 23 as a working assumption. Empirically the reconstruction quality stays flat while the downstream generation quality improves on motion and ImageNet, consistent with the Pareto regime predicted by the analysis.

Remark 24 (Decomposition of the alignment condition). Write $\delta \coloneqq g_{\mathrm{crit}} - g$. By construction δ collects the Jacobian term that sg removes from the training signal. It also captures the structural difference between $\mathcal{L}_{\mathrm{diff}}$ and $\mathcal{L}_{\mathrm{CSA}}$. Then

$$g_{\mathrm{crit}}^\top H^{-1} g \;=\; \|g\|^2_{H^{-1}} \;+\; \delta^\top H^{-1} g.$$

The first term is non-negative and vanishes only at $g = 0$. So Assumption 23 is equivalent to $\delta^\top H^{-1} g > -\|g\|^2_{H^{-1}}$. A simple sufficient (but not necessary) condition is $\delta^\top H^{-1} g \ge 0$. A more

testable condition is $\|\delta\|_{H^{-1}} \le c\,\|g\|_{H^{-1}}$ with $c < 1$. Cauchy-Schwarz then gives $|\delta^\top H^{-1} g| \le c\|g\|^2_{H^{-1}}$. In standard diffusion architectures f_{θ_0} is locally Lipschitz with $\|J_{f_{\theta_0}}\|_{op} \le L$ on $\mathrm{supp}(p_{x_t,t})$. This gives the per-instance bound $\|\delta\| \le 2(1-t)\,L\,\|J_{E_\varphi}\|_{op}\,\|r\|$. The same baseline-residual scale appears in $\|g\|$, so $\|\delta\|/\|g\|$ stays bounded. Assumption 23 reduces to a non-extreme-anti-alignment condition. The ratio $\|\delta\|/\|g\|$ can be probed directly during fine-tuning. To probe the Euclidean alignment between g and g_{crit}, we compared their minibatch estimates on 128 single-image batches from ImageNet [50, 51] across two seeds, using the baseline encoder and a frozen critic. Their Euclidean inner product was positive on every batch (mean cosine 0.93), and $\|\delta\|_2/\|g\|_2$ was below 1 throughout (mean 0.40).

Theorem 25 (Conditional diffusion-loss improvement under gradient alignment; formal form of Theorem 1)**.** *Under Assumptions 16, 18, 22 and 23 and $\lambda_c > 0$ sufficiently small,*

$$\min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi^*(\lambda_c)) < \min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi_0).$$

Proof. Four-step chain.

(i) θ_0 is fixed (the baseline of Assumption 22). By definition of the minimum, $\min_{\theta'} \mathcal{L}_{\mathrm{diff}}(\theta'; \varphi^*) \le \mathcal{L}_{\mathrm{diff}}(\theta_0; \varphi^*)$.

(ii) By Lemma 4, $\mathcal{L}_{\mathrm{diff}}(\theta_0; \varphi^*) = \mathcal{L}_{\mathrm{crit}}(\varphi^*; \theta_0)$.

(iii) The IFT direction is the pseudo-gradient of $\mathcal{L}_{\mathrm{CSA}}$. From Theorem 20, $\varphi^*(\lambda_c) - \varphi_0 = -\lambda_c H^{-1} g + O(\lambda_c^2)$ with $g = \nabla_\varphi \mathcal{L}_{\mathrm{CSA}}(\varphi_0; \theta_0, w)$ from Assumption 18. This is what training follows, and we Taylor-expand $\mathcal{L}_{\mathrm{crit}}$, not $\mathcal{L}_{\mathrm{CSA}}$. By Lemma 4, $\mathcal{L}_{\mathrm{crit}}(\,\cdot\,; \theta_0)$ and $\mathcal{L}_{\mathrm{diff}}(\theta_0; \,\cdot\,)$ agree as value functions of φ. Their true gradients at φ_0 both equal $g_{\mathrm{crit}} = \nabla_\varphi \mathcal{L}_{\mathrm{diff}}(\theta_0; \varphi_0)$. Substituting,

$$\mathcal{L}_{\mathrm{crit}}(\varphi^*; \theta_0) - \mathcal{L}_{\mathrm{crit}}(\varphi_0; \theta_0) = -\lambda_c\, g_{\mathrm{crit}}^\top H^{-1} g + O(\lambda_c^2).$$

The form is mixed. g comes from $\mathcal{L}_{\mathrm{CSA}}$ as the training trajectory (Theorem 20). g_{crit} comes from $\mathcal{L}_{\mathrm{crit}} \equiv \mathcal{L}_{\mathrm{diff}}$ in value as the Taylor coefficient (Lemma 4). $\mathcal{L}_{\mathrm{CSA}}$ never enters the Taylor expansion. Assumption 23 assumes $g_{\mathrm{crit}}^\top H^{-1} g > 0$. The difference is then strictly negative for all sufficiently small $\lambda_c > 0$.

(iv) By Lemma 4 and Assumption 22, $\mathcal{L}_{\mathrm{crit}}(\varphi_0; \theta_0) = \min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi_0)$.

Concatenating gives the strict inequality. □

Corollary 26 (Fixed-θ_0 improvement)**.** *Under Assumptions 16, 18 and 23 and $\lambda_c > 0$ sufficiently small,*

$$\mathcal{L}_{\mathrm{diff}}(\theta_0; \varphi^*) \;<\; \mathcal{L}_{\mathrm{diff}}(\theta_0; \varphi_0).$$

Steps (i)–(iii) of the proof of Theorem 25 do not invoke Assumption 22 and give this directly.

Remark 27 (Fine-tuning inherits the bound). At fixed θ_0, the diffusion loss strictly decreases on the post-trained latents. No retraining of θ is required. Theorem 25 adds Assumption 22 to upgrade this to a min-over-θ statement. Fine-tuning θ from θ_0 on φ^* can only decrease the loss further. The practical fine-tune setting therefore inherits the strict bound either way.

Remark 28 (Approximate baseline optimality). If $\mathcal{L}_{\mathrm{diff}}(\theta_0; \varphi_0) = \min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi_0) + \epsilon_{\mathrm{opt}}$, then Theorem 25 relaxes to

$$\min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi^*) \le \min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi_0) + \epsilon_{\mathrm{opt}} - \lambda_c\, g_{\mathrm{crit}}^\top H^{-1} g + O(\lambda_c^2).$$

The improvement persists as long as $\lambda_c\, g_{\mathrm{crit}}^\top H^{-1} g > \epsilon_{\mathrm{opt}}$.

A.8 Frequency-dependent effective temperature (DenseAM extension)

Pham et al. [68] identify the empirical diffusion marginal

$$p(x_t, t) = \tfrac{1}{K} \sum_\mu (2\pi\sigma^2 t)^{-N/2} \exp\!\big(-\|x_t - \xi_\mu\|^2 / (2\sigma^2 t)\big)$$

with a Dense Associative Memory at scalar inverse temperature $\beta(t) = 1/(2\sigma^2 t)$. We extend this identification to a per-mode effective temperature under spatial stationarity.

Proposition 29 (Per-mode effective temperature)**.** *Assume the memories $\{\xi_\mu\}_{\mu=1}^K$ form a temporally wide-sense stationary (WSS) process: constant mean $\mathbb{E}[\xi_\mu(s)] = \mu_0$ for all s, and autocovariance $\mathrm{Cov}(\xi_\mu(s), \xi_\mu(t))$ depending only on the lag $|s-t|$. By Wiener-Khinchin, the population power spectral density (PSD) is the temporal Fourier transform of the autocovariance:*

$$\mathrm{S}(k) \;:=\; \mathcal{F}\big[\mathrm{Cov}(\xi_\mu(0), \xi_\mu(\cdot))\big](k).$$

The empirical estimator $\hat{\mathrm{S}}(k) := K^{-1}\sum_{\mu=1}^K |(\mathcal{F}\,\xi_\mu)(k)|^2$ converges to $\mathrm{S}(k)$ as $K \to \infty$ under ergodicity, with $O(K^{-1/2})$ deviation at finite K.

Because $\mathcal{F}$ is unitary, Parseval gives $\|x_t - \xi_\mu\|^2 = \sum_k |\hat{x}_{t,k} - \hat{\xi}_{\mu,k}|^2$, and each Boltzmann factor splits across modes:

$$p(x_t, t) = \frac{1}{K}\sum_\mu (2\pi\sigma^2 t)^{-N/2} \prod_k \exp\Big(-\frac{|\hat{x}_{t,k} - \hat{\xi}_{\mu,k}|^2}{2\sigma^2 t}\Big).$$

The sum over μ does not factor across k. WSS gives second-order spectral uncorrelatedness across modes, not joint independence. We treat modes as independent as a working approximation. The per-mode signal and noise energies are then $E_{\mathrm{sig}}(k) = \mathrm{S}(k)$ and $E_{\mathrm{noise}}(k,t) = \sigma^2 t$, giving a per-mode SNR:

$$\beta_{\mathrm{eff}}(k,t) \;\propto\; E_{\mathrm{sig}}(k)/E_{\mathrm{noise}}(k,t) \;=\; \mathrm{S}(k)/(\sigma^2 t). \tag{11}$$

Proof. Apply the scalar identification of Pham et al. [68] per mode to each Boltzmann factor above. The per-mode signal energy scales as $\mathrm{S}(k)$ by the Wiener-Khinchin definition. □

Remark 30 (Reading of Proposition 29). The mode-independence step is a working approximation, not an exact identity. WSS gives only second-order uncorrelatedness across modes. Joint independence is a separate assumption that we adopt for tractability. The polynomial weight $w(k) = 1 + \kappa d(k)^p$ should be read as motivated by this approximation, not derived as a unique identity. The empirical spectral profile in Fig. 1a supports this choice in our deployment regime.

Remark 31 (Static weight as SNR compensation). For a power-law PSD $\mathrm{S}(k) \propto \|k\|^{-\gamma}$, the SNR-compensation choice from (11) is $\tilde{w}(k) \propto \beta_{\mathrm{eff}}^{-1} \propto \|k\|^\gamma$. This makes $\tilde{w}(k)\beta_{\mathrm{eff}}(k,t)$ approximately constant across k. The encoder pressure $\frac{2}{N}\tilde{w}(k)|\hat{r}(k)|$ from Theorem 13 then applies more force at modes with smaller per-mode SNR. For $D > 1$ the same construction applies axis-wise. The additive 1 in $w(k) = 1 + \kappa\, d(k)^p$ does not come from (11). It keeps $\tilde{w}(k) > 0$, which Theorem 10 uses. It also keeps a baseline pressure on low-frequency modes.

A.9 Practical considerations for text-to-motion deployment

The setup in Appendix A.2 is already text-conditioned. This subsection collects practical remarks for deployment.

Remark 32 (Scope of the improvement). Theorem 25 states $\min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi^*) < \min_\theta \mathcal{L}_{\mathrm{diff}}(\theta; \varphi_0)$ where $\mathcal{L}_{\mathrm{diff}}$ is the joint-expectation loss $\mathbb{E}_{(x,y),t,\epsilon}[\cdot]$ (equivalently $\mathbb{E}_{y\sim p_y}[\mathcal{L}^y_{\mathrm{diff}}]$ via tower property, with $\mathcal{L}^y_{\mathrm{diff}}$ the per-prompt loss). The result is a min-over-θ statement about the prompt-averaged loss. It does not say a fixed θ achieves this drop, nor does it say each fixed prompt y improves. Text alignment, R-Precision, FID, and related metrics depend on the full sampler and decoder pipeline, so the theorem does not predict them.

Remark 33 (Classifier-free guidance and approximate optimality). Pretrained text-to-motion diffusion models are usually trained with classifier-free guidance (CFG) [69]. With probability p_{uncond} the text y is replaced by a null token during training, so the actual training objective is a mixture of conditional and unconditional losses. Hence θ_0 does not exactly minimize the conditional $\mathcal{L}_{\mathrm{diff}}(\theta; \varphi_0)$ in Assumption 22. Remark 28 covers this case. The improvement of Theorem 25 persists when the conditional optimality gap is small relative to $\lambda_c\, g_{\mathrm{crit}}^\top H^{-1} g$. CSA uses the raw conditional prediction $f_{\theta_0}(x_t, t, y)$, not the CFG-guided combination used at sampling time.

Remark 34 (Encoder stays text-free). The conditioning on y comes from the data distribution $p(x|y)$, not from the encoder. The map $E_\varphi(x)$ takes only motion. Different texts induce different latent distributions because they select different motions, and the critic gradient picks up the text signal through $f_{\theta_0}(\cdot, \cdot, y)$.

B ImageNet generalization study

We apply the CSA pipeline to class-conditional ImageNet [50, 51]. The autoencoder is the Stable Diffusion VAE (SD-VAE) backbone [16], fine-tuned with CSA at $\kappa = 1.0$, $\lambda_c = 0.1$, $p = 2.0$, for $5{,}000$ fine-tune steps with $1{,}000$-step warmup on λ_c, learning rate 10^{-6}, batch size 8. The diffusion model is a 200M-parameter class-conditional UNet at 256×256 resolution. The architecture has channels $(192, 384, 576, 768)$, three layers per block, dropout 0.1, and $1{,}001$ class embeddings. We train from scratch on the 1.28M ImageNet training images for $300{,}000$ steps at batch size 128. Peak learning rate is 10^{-4} with linear warmup over $5{,}000$ steps followed by cosine decay. We use exponential moving average (EMA) decay 0.9999 and CFG dropout probability 0.10. Reconstruction FID is measured on $50{,}000$ validation images. Generation FID uses 50 samples per class generated via 50-step deterministic sampling at CFG scale $w = 2.0$. We use a single seed (42).

Table 4 reports the paired baseline-vs-CSA result. Three observations stand out. Reconstruction FID stays close to baseline ($\Delta = +0.08$), and generation FID improves at all three checkpoints we evaluate ($\Delta = -2.24$ at $100{,}000$ training steps, $\Delta = -1.69$ at $200{,}000$ training steps, and $\Delta = -1.35$ at $300{,}000$ training steps). Moreover, the diffusion training loss curves separate by about 0.30 from step $1{,}000$ onward (Figure 5). Taken together, all three are consistent with the prediction of Theorem 1. The post-trained latent is easier for the diffusion to fit, and reconstruction is preserved at first order in λ_c.

Table 4: ImageNet generalization study. Single seed (42), 200M class-conditional UNet, CFG scale $w = 2.0$. Reconstruction FID is computed on $50{,}000$ validation images. Generation FID is computed from $50{,}000$ CFG samples (50 per class).

Metric	Baseline	CSA	Δ
Reconstruction FID	2.79	**2.87**	$+0.08$
Generation FID at 100k steps	36.29	**34.05**	-2.24
Generation FID at 200k steps	25.46	**23.77**	-1.69
Generation FID at 300k steps	20.82	**19.47**	-1.35

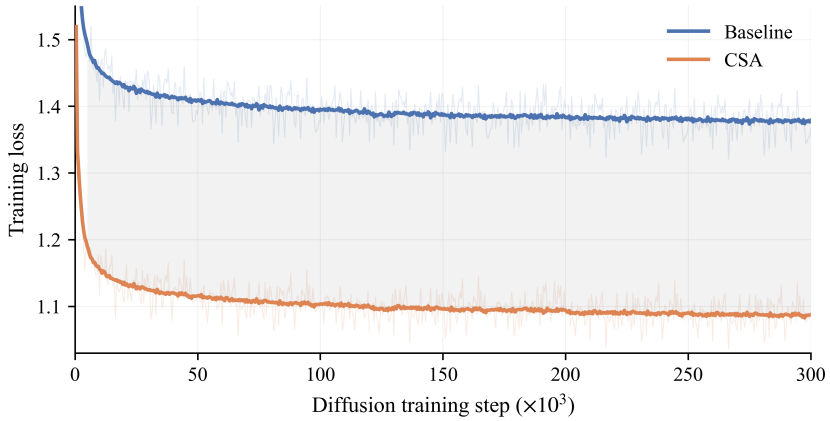


Figure 5: Diffusion training loss on class-conditional ImageNet latents. CSA-fine-tuned latents (orange) yield consistently lower training loss than the SD-VAE baseline (blue) from step $1{,}000$ onward. Faded curves show raw step-wise loss. Solid curves show 0.999-EMA-smoothed loss. The shaded band marks the sustained gap of about 0.30.

C MeanFlow Identity Derivation

This appendix derives the JVP form of the average velocity (5) used in Section 3.2 from the integral definition (4), following Geng et al. [37].

Recall the average velocity definition:

$$u(x_t, r_d, t, z) = \frac{1}{t - r_d} \int_{r_d}^{t} v(x_\tau, \tau, z)\, d\tau. \tag{12}$$

Multiplying both sides by $(t - r_d)$ removes the prefactor and gives an integral identity:

$$(t - r_d)\, u(x_t, r_d, t, z) = \int_{r_d}^{t} v(x_\tau, \tau, z)\, d\tau. \tag{13}$$

We now differentiate both sides of (13) with respect to t, treating r_d as independent of t.

The right-hand side reduces by the fundamental theorem of calculus to

$$\frac{d}{dt} \int_{r_d}^{t} v(x_\tau, \tau, z)\, d\tau = v(x_t, t, z). \tag{14}$$

The left-hand side expands by the product rule:

$$\frac{d}{dt} \big[(t - r_d)\, u(x_t, r_d, t, z)\big] = u(x_t, r_d, t, z) + (t - r_d)\, \frac{d}{dt} u(x_t, r_d, t, z). \tag{15}$$

Equating (14) and (15) and rearranging recovers the MeanFlow identity:

$$u(x_t, r_d, t, z) = v(x_t, t, z) - (t - r_d)\, \frac{d}{dt} u(x_t, r_d, t, z). \tag{16}$$

This matches (5) in the main text.

At training time, the total derivative $\frac{d}{dt} u(x_t, r_d, t, z)$ is computed via the Jacobian-vector product (JVP) of the decoder network with the tangent vector $(v(x_t, t, z),\ 0,\ 1,\ 0)$ along the (x, r_d, t, z) axes. The instance of u inside the time derivative is detached, denoted u_{ψ^-} in (5), to prevent recursive backpropagation through the network's own prediction [37]. Without this stop-gradient, the loss would induce double backpropagation through the JVP and require higher-order optimization.

D Generation Metric

Frechet Inception Distance (FID) The FID [70] measures the distribution distance between the generated and real interaction features.

$$\text{FID} = \|\mu_{\text{gt}} - \mu_{\text{pred}}\|^2 + \text{Tr}(\Sigma_{\text{gt}} + \Sigma_{\text{pred}} - 2(\Sigma_{\text{gt}}\Sigma_{\text{pred}})^{\frac{1}{2}})$$

where μ_{gt} and μ_{pred} are the mean ground-truth and generated interaction features, and Σ represents the covariance matrix.

Multimodal Distance (MM-Dist) This metric calculates the average Euclidean distance between each text feature and the generated interaction feature.

$$\text{MM-Dist} = \frac{1}{N} \sum_{i=1}^{N} \|f_{t,i} - f_{m,i}\|$$

where $f_{t,i}$ and $f_{m,i}$ are the features of the ith text-interaction pair.

Diversity All generated interactions are randomly sampled to calculate the average Euclidean distance between two subsets.

$$\text{Diversity} = \frac{1}{X_d} \sum_{i=1}^{X_d} \|x_i - x'_i\|$$

Multimodality (MModality) This metric assesses the variability given multiple text descriptions by calculating the average pairwise Euclidean distance between motion features.

$$\text{MModality} = \frac{1}{J_m \times X_m} \sum_{j=1}^{J_m} \sum_{i=1}^{X_m} \|x_{j,i} - x'_{j,i}\|$$

where $x_{j,i}$ and $x'_{j,i}$ are the features of the jth pair of the ith text description.

E Implementation Details

Diffusion backbone Training. In DiMoT, we train the diffusion backbone for different tokenizers in a fixed way. Firstly, each motion sequence is encoded into a set of 16 latent tokens with a channel dimension of 32, and the diffusion backbone denoises them conditioned on text. The denoiser is implemented as a transformer with an encoder-only topology. Each latent token is first projected to a 512-dimensional hidden space, processed by an 11-layer skip-connected Transformer encoder with 8 attention heads, a feed-forward dimension of 1024, GELU activations, and dropout of 0.1, and then projected back to the original latent dimension. We use learned 1D positional embeddings for the latent token sequence. Conditioning is injected by prepending two condition tokens to the latent sequence: (1) a timestep token obtained from a sinusoidal timestep embedding followed by an MLP projection, and (2) a text token produced by a frozen CLIP text encoder followed by a 2-layer Transformer Encoder for feature refinement. Thus, the denoiser processes a joint sequence consisting of 16 motion latent tokens, one timestep token, and one global text token. Notably, the Transformer backbone is not a plain stacked encoder: it adopts a skip-encoder design, in which hidden states from the early encoder blocks are concatenated with their symmetric late blocks via learnable linear fusion, thereby improving cross-layer information propagation while preserving global self-attention. By default, the model is trained with a flow-matching velocity prediction objective and sampled with classifier-free guidance at guidance scale 3.0 in 10 steps. The training process uses a batch size of 192, a peak learning rate of 1e-4, a 1K-iteration linear warm-up, and a cosine decay schedule. In the CSA stage, we use the same denoiser backbone, which remains frozen.

Tokenizer Training. We reproduce MLD++ [65] and adopt it as our baseline with multi-latent-token training. The baseline tokenizer is trained for 6K epochs with a peak learning rate of 1×10^{-4}, a batch size of 96, and a cosine decay schedule following a 1K-epoch linear warmup. During the CSA stage, we train the encoder for 1K epochs and gradually introduce spectral reweighting over a 200-epoch warmup. The encoder is optimized with an ℓ_2 reconstruction objective, and the CSA loss is weighted by $\lambda_c = 0.1$. We evaluate CSA under both training-from-scratch and baseline fine-tuning settings. For the fine-tuning variant, we additionally study an encoder-side adapter implemented as a 2-layer residual block. The spectral hyperparameters are set to $\kappa = 1.0$ and $p = 2.0$. The critic is pretrained on the baseline tokenizer's latents and remains frozen throughout CSA. After CSA, we freeze the adapted encoder and train a separate generative diffusion model from scratch on its latents, using the same architecture and training configuration as the baseline diffusion model. This generative model supplies diffusion-generated latent conditions for AMD training. In detail, we replace deterministic decoding with a latent-conditioned generative decoder. The decoder uses a 9-layer Transformer [71], while the latent-to-frame upsampling projection is implemented as a 2-layer Transformer. We train the AMD decoder from scratch for 2K epochs, where latent conditions are sampled from the pretrained diffusion output with probability 0.25 and from the encoder output with probability 0.75. For 50% of each batch, we apply marginal flow-matching supervision by setting $t = r_d$. For the remaining samples, we use an α-curriculum that linearly decreases the generative mixing factor from 1.0 to 0.005 over the first 1K epochs and then sets it to 0 for the final 1K epochs. We further apply single-step classifier-free guidance [37] with guidance scale 3.

Compute resources. All experiments were conducted on the High-Performance Computing cluster. We use a Linux server equipped with a 30-core Intel Xeon 6767P CPU, 235 GiB of system memory, and an NVIDIA H200 GPU.

F AMD ablation

Table 5: AMD ablation.

Variant	Proj.	Step	Target	α Sched. (Ratio)	Evaluation Metrics				
					FID↓	R-Top3↑	MPJPE↓	Foot Sliding↓	AITS↓
MLD++ (Baseline)	–	–	–	–	5.046	0.688	49.26	0.0186	1.05
Target v Flow Matching Velocity Baseline									
V1: + Interpo.	Interpolation	1 NFE	v	None (0%)	7.625	0.551	66.12	0.0255	3.05
V2: + Trans.	Transformer	16 NFE	v	None (0%)	4.951	0.708	47.12	0.0181	45.16
V3: + Trans.	Transformer	1 NFE	v	None (0%)	6.842	0.589	58.45	0.0221	4.82
Target u Projection Search									
V4: Interpolation	Interpolation	1 NFE	u	None (0%)	5.215	0.675	49.85	0.0191	3.13
V5: MLP	MLP	1 NFE	u	None (0%)	5.012	0.694	47.53	0.0178	3.64
Target u + Marginal Distribution ratio $t = r$									
V6: Ratio 25%	Transformer	1 NFE	u	Annealed (25%)	4.832	0.710	46.90	0.0183	4.80
V7: Ratio 50% (Default)	Transformer	1 NFE	u	Annealed (50%)	4.781	0.716	46.76	0.0175	4.80
V8: Ratio 75%	Transformer	1 NFE	u	Annealed (75%)	4.805	0.714	46.82	0.0176	4.80

G CSA ablation

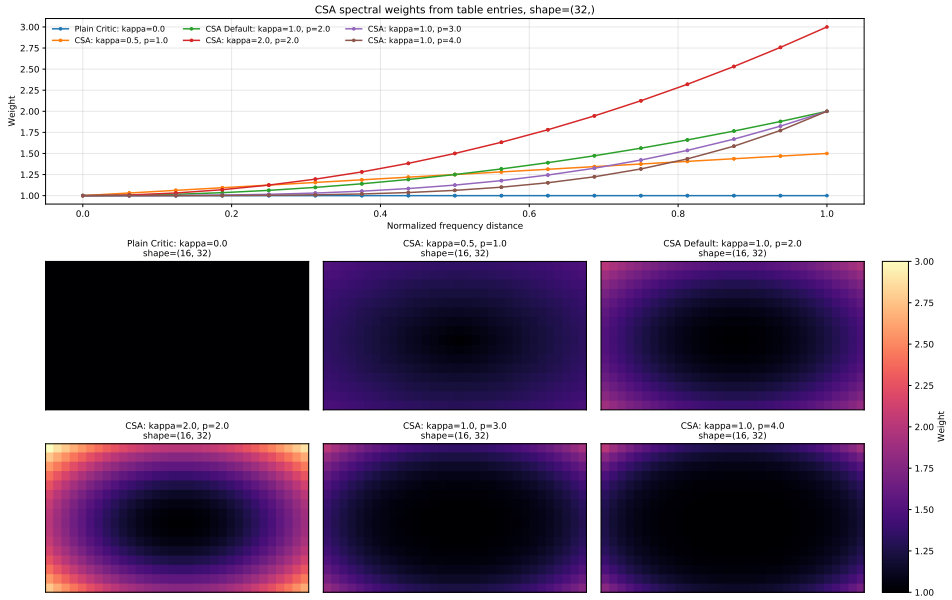


Figure 6: Spectral Reweighting Curve.

Table 6: Ablation of CSA Spectral Weights (κ, p) and Critic Scale (λ_c)

Method	κ	p	λ_c	rFID ↓	MPJPE ↓	FID ↓	R-Precision ↑
Baseline	-	-	0.0	0.080	49.26	5.046	0.688
Plain Critic	0.0	-	0.1	0.082	48.50	5.012	0.708
CSA	0.5	1.0	0.1	0.084	47.85	4.985	0.728
CSA (Default)	**1.0**	**2.0**	**0.1**	**0.086**	**47.18**	**4.967**	**0.746**
CSA	2.0	2.0	0.1	0.125	51.32	5.215	0.712
CSA	1.0	3.0	0.1	0.188	58.45	5.682	0.665
CSA	1.0	4.0	0.1	0.245	66.80	5.932	0.621
CSA	1.0	2.0	0.01	0.081	48.60	5.035	0.698
CSA	1.0	2.0	0.5	0.142	54.20	5.410	0.685

G.1 Effect of the Alpha Curriculum Schedule

We compare the default linear curriculum with an AlphaFlow-style sigmoid curriculum [52], keeping the remaining AMD settings unchanged. AMD is trained for 2K epochs. For samples with non-degenerate intervals ($r_d < t$), both schedules decrease α from 1 to $\alpha_{\min} = 0.005$ over the first 1K epochs, followed by the exact MeanFlow objective ($\alpha = 0$) for the final 1K epochs. The degenerate-interval ratio remains 50% for both schedules.

Let e denote training progress in epochs and $q = e/1000$. For schedule S, we use

$$\alpha_S(e) = \begin{cases} \alpha_{\min} + (1 - \alpha_{\min}) g_S(q), & 0 \le e \le 1000, \\ 0, & 1000 < e \le 2000. \end{cases} \tag{17}$$

The linear schedule uses $g_{\text{lin}}(q) = 1 - q$. For the sigmoid schedule, with $\sigma(a) = (1 + \exp(-a))^{-1}$ and $\gamma = 25$, we use

$$g_{\text{sig}}(q) = \frac{\sigma(\gamma/2) - \sigma(\gamma(q - 1/2))}{\sigma(\gamma/2) - \sigma(-\gamma/2)}. \tag{18}$$

This normalization gives the same initial and final α values for both curricula.

Table 7: Comparison of α curriculum schedules. Both schedules use the same training duration and endpoint values.

Schedule	FID↓	R-Top3↑	MPJPE↓	Foot Sliding↓
Linear (default)	4.781	0.716	46.76	0.0175
Sigmoid ($\gamma = 25$)	4.856	0.687	49.52	0.0178

The linear schedule performs better on all four reported metrics in this comparison. The sigmoid schedule concentrates the transition around the middle of the first training stage, whereas the linear schedule changes α at a constant rate. These results support the default linear curriculum under the evaluated setting.

H User Study against InterMask

We conduct a user study comparing DiMoT with InterMask [64]. The study involves 11 participants and 20 randomly selected samples. We evaluate preferences along four dimensions: text alignment, physical realism, interaction quality, and overall quality.

Table 8: User preferences for InterMask and DiMoT. Higher percentages indicate greater preference. Ties are excluded from the overall-quality comparison.

Criterion	InterMask	DiMoT
Text alignment	53.33%	46.67%
Physical realism	46.15%	53.85%
Interaction quality	43.80%	56.20%
Overall quality	47.39%	52.61%

DiMoT receives higher preference percentages for physical realism, interaction quality, and overall quality, while InterMask receives higher preference for text alignment. The overall-quality difference is modest, and the study is limited in participant and sample size. We therefore interpret these results as descriptive evidence of user preference in this comparison, rather than as a statistically established advantage.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We demonstrate our novel DiMoT design.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the impact of the CSA assumption.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have provided a thorough theoretical explanation for CSA theory in Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The code will be made publicly available after the review process.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code will be made publicly available after the review process.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We repeated the description of the experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We presented the results in detail.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the relevant content in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have included the relevant content.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: We cite the original paper of the used assets in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: we communicate the details of the model as part of submissions via structured templates.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the proposed method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.