

Streaming Generation of Co-Speech Gestures via Accelerated Rolling Diffusion

Evgeniia Vu^{1*}, Andrei Boiarov^{2*}, Dmitry Vetrov¹

¹Constructor University, Bremen, Germany

²Constructor Tech, Sofia, Bulgaria
andrei.boiarov@constructor.tech

Abstract

Generating co-speech gestures in real time requires both temporal coherence and efficient sampling. We introduce a novel framework for streaming gesture generation that extends Rolling Diffusion models with structured progressive noise scheduling, enabling seamless long-sequence motion synthesis while preserving realism and diversity. Our framework is universally compatible with existing diffusion-based gesture generation models, transforming them into streaming methods capable of continuous generation without requiring post-processing. We evaluate our framework on ZEGGS and BEAT, strong benchmarks for real-world applicability. Applied to state-of-the-art baselines on both datasets, it consistently outperforms them, demonstrating its effectiveness as a generalizable and efficient solution for real-time co-speech gesture synthesis. We further propose Rolling Diffusion Ladder Acceleration (RDLA), a new approach that employs a ladder-based noise scheduling strategy to simultaneously denoise multiple frames. This significantly improves sampling efficiency while maintaining motion consistency, achieving up to a 4× speedup with high visual fidelity and temporal coherence in our experiments. Comprehensive user studies further validate our framework’s ability to generate realistic, diverse gestures closely synchronized with the audio input.

Code — <https://github.com/andrewbo29/co-speech-gestures-rolling-diffusion>

1 Introduction

Co-speech gestures enhance non-verbal communication by reinforcing spoken content, and are essential for realism in virtual avatars, video conferencing, gaming, and embodied AI (Nyatsanga et al. 2023). Real-time, or streaming, generation is particularly important for interactive settings such as virtual assistants, gaming, and telepresence systems. Recent work largely relies on data-driven deep learning to produce realistic motion, with diffusion-based models standing out for their strong realism and diversity. Approaches such as DiffuseStyleGesture (Yang et al. 2023), which uses cross-local and self-attention for gesture–speech synchronization, and Taming Diffusion (Zhu et al. 2023), which

applies audio-conditioned diffusion with transformer guidance, both rely on seed-pose conditioning to ensure smooth transitions and temporal coherence.

Despite recent progress, diffusion-based methods still struggle in real-time settings. To manage long sequences, many models generate fixed-length chunks or incremental extensions. PersonaGestor (Zhang et al. 2024) uses fuzzy feature inference yet still stitches chunked outputs, while DiffSHEG (Chen et al. 2024) depends on incremental outpainting. Such non-streaming strategies often cause visual discontinuities and add latency due to post-processing. Methods that condition on previous frames improve continuity but incur substantial computational overhead, limiting real-time use (see Supplementary Material in the extended version (Vu, Boiarov, and Vetrov 2025)).

A promising alternative is the Rolling Diffusion framework (Ruhe et al. 2024), which turns diffusion models into an autoregressive process and improves temporal consistency for long sequences. However, applying it to real-time co-speech gesture generation remains difficult, primarily because the autoregressive loop is computationally expensive.

To address these challenges, we introduce a novel Rolling Diffusion based framework integrating diffusion models with real-time capabilities for co-speech gesture generation. By employing a structured ladder-based noise scheduling strategy, our Rolling Diffusion Ladder Acceleration (RDLA) approach simultaneously denoises multiple frames, significantly improving sampling efficiency. This achieves generation speeds of up to 200 FPS without compromising visual fidelity or temporal coherence. Extensive experiments on the ZEGGS (Ghorbani et al. 2023) and BEAT (Liu et al. 2022) benchmarks confirm our method’s effectiveness for realistic, diverse gesture generation in streaming contexts. In summary, our key contributions are:

1. We are the first, to our knowledge, to successfully adapt a rolling diffusion framework to a practical application, specifically demonstrating its effectiveness in real-time co-speech gesture generation.
2. We propose a universal framework for converting any diffusion-based gesture generation approach into a real-time streaming model without requiring post-processing.
3. We provide comprehensive evaluations on standard benchmarks and user studies, showing that our method

*These authors contributed equally.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

consistently outperforms baselines across the benchmarks, demonstrating its effectiveness as a generalizable solution for real-time, high-fidelity co-speech gesture synthesis.

4. We introduce RDLA, substantially increasing inference speed with minimal impact on gesture quality.

2 Proposed Approach

Rolling Diffusion Models

Diffusion models (Song and Ermon 2019; Ho, Jain, and Abbeel 2020) consist of a forward (diffusion) process and reverse process. The forward process gradually adds Gaussian noise to a data sample x^0 over T steps such that $x^T \sim \mathcal{N}(0, 1)$ and each transition is defined as:

$$q(x^t | x^{t-1}) = \mathcal{N}(x^t; \sqrt{1 - \beta^t}x^{t-1}, \beta^t I) \quad (1)$$

where β^t is a variance schedule controlling noise intensity. The marginal distribution after t steps is:

$$q(x^t | x^0) = \mathcal{N}(x^t; \sqrt{\bar{\alpha}^t}x^0, (1 - \bar{\alpha}^t)I) \quad (2)$$

where $\alpha^t = 1 - \beta^t$ and $\bar{\alpha}^t = \prod_{s=1}^t \alpha^s$.

The reverse process learns to denoise x^t using a neural network $f_\theta(x^t, t) = \hat{x}$. The model is trained by minimizing the objective: $\mathcal{L}(\theta) := \mathbb{E}_{t, x^0} [a(t) \|x^0 - \hat{x}\|^2]$, where $a(t)$ is a weighting function that can be specified to control the importance of different timesteps during training. This objective encourages $f_\theta(x^t, t)$ to accurately estimate the initial signal of data sample.

Rolling Diffusion Models (RDMs) (Ruhe et al. 2024) introduce a modification to standard diffusion models by incorporating a progressive corruption process along the temporal axis, making them particularly well-suited for sequential data $\mathbf{X} = \{x_k^0\}_{k=0}^{L-1}$, where x_l is k -th element of sequence. Unlike standard diffusion models that apply noise uniformly across all frames, RDMs operate with a rolling window of size N . In this setup, the noise level gradually increases from the first to the last frame in the window, enabling a seamless transition. In our implementation, we discretize time which indicates the noise level using $t \in \{0, \dots, T\}$ instead of the continuous range $t \in [0, 1]$, where $T = 1000$ represents the total number of noise steps. Since the generation window size N is typically much smaller than T , the noise level difference between adjacent frames is greater than 1. We define this difference as a step s , calculated as $s = \frac{T}{N}$. To ensure uniform noise distribution across frames, we select the generation window as a divisor of T , ensuring consistent step sizes and preventing uneven noise application. This structured noise scheduling allows for a more controlled and stable generation process, improving the overall quality of generated sequences. Then the rolling window takes the form $\mathbf{x}_j^{t_0} = \{x_{j+n}^{t_n}\}_{n=0}^{N-1}$, where $t_0 \in \{1, \dots, s\}$ is a noise level of the first frame in the window, $t_n = t_0 + ns$ is a noise level of n^{th} frame.¹

¹Since noise levels of all frames within a rolling window are deterministic functions of t_0 we may use this index to define the noise levels of the entire rolling window $\mathbf{x}_j^{t_0}$.

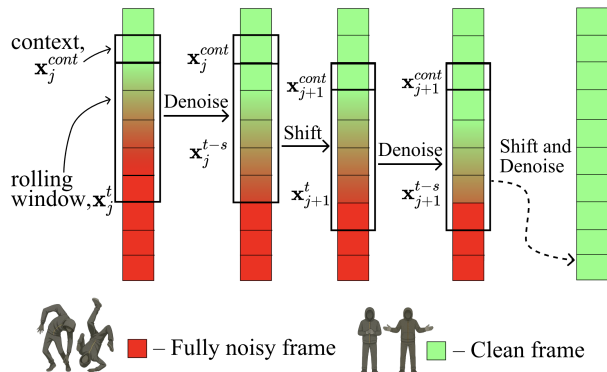


Figure 1: Visualization of the rolling denoising process with parameters $T = 5$, $N = 5$, $n^{\text{cont}} = 1$, $s = 1$

In the forward process, noise is applied progressively as:

$$q(\mathbf{x}_j^{t_0} | \mathbf{x}_j) = \prod_{n=0}^{N-1} \mathcal{N}(x_{j+n}^{t_n} | \alpha^{t_n} x_{j+n}^0, (1 - \alpha^{t_n})^2 I) \quad (3)$$

During training, the model $f_\theta(\mathbf{x}_j^{t_0}, t) = \hat{\mathbf{x}}$ processes only the frames within the rolling window. The parametrized reverse process $p_\theta(\mathbf{x}_j^{t_0-1} | \mathbf{x}_j^{t_0})$ is defined as:

$$p_\theta(\mathbf{x}_j^{t_0-1} | \mathbf{x}_j^{t_0}) = \prod_{n=0}^{N-1} q(x_{j+n}^{t_n-1} | x_{j+n}^{t_n}, \hat{x}_{j+n}) \quad (4)$$

Once the first frame is fully denoised, i.e. t_0 reaches 0, a new frame, sampled from Gaussian noise, is introduced, and the window shifts forward ($j = j + 1$, $t_0 = s$) to continue the denoising process. This approach allows for continuous and unbounded generation, making RDMs particularly effective for producing arbitrarily long sequences.

The training objective is

$$\mathcal{L}(\theta) := \mathbb{E}_j \mathbb{E}_{\mathbf{x}_j^0} \sum_{t^0=1}^s \mathbb{E}_{\mathbf{x}_j^{t_0}} \sum_{n=0}^{N-1} a(t_n) \|x_{j+n}^0 - \hat{x}_{j+n}\|^2, \quad (5)$$

where x_k^t represents a single frame with the upper index t denoting the noise level and the lower index k representing the index in the sequence and $a(t_n)$ is a weighting function (e.g. signal-to-noise ratio).

To achieve progressive noise scheduling, RDMs operate in two distinct phases: initialization and rolling. In the initialization phase, the model starts with a fully noisy sequence and gradually denoises it to the partially clean state required by the rolling window. Once this point is established, the model enters the rolling phase, where the rolling denoising process described above is applied.

Method

In our work, we adapt rolling diffusion models for co-speech gesture generation, introducing a novel framework that transforms any diffusion-based architecture into a streaming

model. Our approach enables seamless and continuous gesture generation of arbitrary length by modifying the model architecture and integrating a structured noise scheduling mechanism, which, combined with the rolling denoising process, ensures smooth temporal transitions and prevents abrupt motion discontinuities. As illustrated in figure 1, the model generates a new clean frame in each s -step and shifts the generation window forward to include the new frame at the end.

As a condition, the model receives audio features as input $U = \{u_k\}_{k=0}^{L-1}$. Audio processing depends on the implementation of the baseline model, but the most popular approach is to use a pre-trained model such as WavLM (Chen et al. 2022). Also, depending on the baseline’s architecture, the model can receive a speech style or speaker ID as an additional condition.

The architecture of the baseline model is mostly preserved, and the main change affects the integration of time embedding. All models incorporate time embeddings into each frame, either by summing or concatenating them with the frame embeddings. In the original models, a single time embedding is replicated across all frames. In contrast, our approach assigns a unique time embedding to each frame within the rolling window.

Training Description

During training, we sample the initial noise level t_0 for the first frame from a uniform distribution $t_0 \sim \text{Uniform}(\{1, \dots, s\})$ and select the starting index j for the rolling window uniformly $j \sim \text{Uniform}(\{1, \dots, L - N\})$. We then determine the noise levels for subsequent frames according to $t_n = t_0 + ns$. These noise levels are applied using eq. (3). As a result, the noise level for each frame n in the sequence falls within the range $t_n \in [s \cdot n, s \cdot (n + 1))$. To improve the performance of the model, we include several clean frames at the beginning of the sequence as an additional context, represented by $\mathbf{x}_j^{\text{cont}} = (x_{j-n}^{\text{cont}}, \dots, x_{j-1}^{\text{cont}})$ with a length n^{cont} . We discovered that applying a minimal level of noise $t = 1$ to these context frames is essential. This process acts as a form of regularization and improves stability (see the Supplementary Material for more details). Therefore, at each time step, the input to the model consists of a concatenated sequence $[\mathbf{x}_j^{\text{cont}}, \mathbf{x}_j^t]$. Here, \mathbf{x}_j^t denotes a rolling window whose first frame corresponds to the t -th noise level, and the associated audio features for this sequence are given by $u_j = (u_{j-n^{\text{cont}}}, \dots, u_{j+N-1})$.

Unlike prior work, our model is trained exclusively for the rolling phase, omitting an initial descent phase. Furthermore, we use $a(t_n) = 1$ in the training objective for all t , instead of using the signal-to-noise ratio (SNR). The training procedure is summarized in algorithm 1.

Sampling Process Description

To create a progressively noisy sequence within the rolling window, we begin by padding the sequence with idle poses, each characteristic of the style and corresponding to silence. These initial poses are then noised according to a schedule

Algorithm 1: Training

- 1: **repeat**
- 2: $j \sim \text{Uniform}(\{1, \dots, L - N\})$
- 3: $t_0 \sim \text{Uniform}(\{1, \dots, s\})$
- 4: $\mathbf{x}_j^0 \leftarrow (x_j^0, \dots, x_{j+N-1}^0)$
- 5: $\mathbf{u}_j \leftarrow (u_{j-n^{\text{cont}}}, \dots, u_{j+N-1})$
- 6: Sample $\mathbf{x}_j^{t_0} \sim q(\mathbf{x}_j^{t_0} | \mathbf{x}_j)$ (see eq. (3))
- 7: Compute $\hat{\mathbf{x}}_j \leftarrow f_\theta([\mathbf{x}_j^{\text{cont}}, \mathbf{x}_j^{t_0}], t_0, \mathbf{u}_j)$
- 8: Take gradient descent step on

$$\nabla_\theta \sum_{n=0}^{N-1} a(t_n) \|x_{j+n}^0 - \hat{x}_{j+n}\|^2$$

- 9: **until** Converged
-

starting at $t_0 = s$. At each s -th step, we get a fully denoised frame, which is appended to the output sequence. Subsequently, a new frame is sampled from Gaussian noise, added to the end of the rolling window, and the window is shifted. The sampling process is outlined in algorithm 2.

3 Rolling Diffusion Ladder Acceleration

In the standard rolling diffusion sampling process (Section 2), only a single frame is fully denoised at each s -th step, leading to a sequential bottleneck that slows down the overall generation. Hence we cannot make T smaller than N , which corresponds to $s = 1$. To overcome this limitation, we introduce Rolling Diffusion Ladder Acceleration

Algorithm 2: Sampling

Require: audio $\{u_j\}_{j=0}^{L-1}$, idle x_{idle}

Ensure: Resulted prediction y

- 1: **for** $n = -N, \dots, -1$ **do**
 - 2: $t_n = s(N + n + 1)$
 - 3: $u_n = 0$
 - 4: Sample $x_n^{t_n} \sim q(x_n^{t_n} | x_{\text{idle}})$
 - 5: **end for**
 - 6: $\mathbf{u}_{-N} \leftarrow (u_{-N-n^{\text{cont}}}, \dots, u_{-1})$
 - 7: $\mathbf{x}_{-N}^s \leftarrow (x_{-N}^s, \dots, x_{-1}^s)$
 - 8: $\mathbf{x}_{-N}^{\text{cont}} \leftarrow (x_{\text{idle}}, \dots, x_{\text{idle}})$
 - 9: $j = -N$
 - 10: **repeat**
 - 11: **for** $t_0 = s, s - 1, \dots, 1$ **do**
 - 12: $\hat{\mathbf{x}}_j \leftarrow f_\theta([\mathbf{x}_j^{\text{cont}}, \mathbf{x}_j^{t_0}], t_0, \mathbf{u}_j)$
 - 13: Sample $\mathbf{x}_j^{t_0-1} \sim p_\theta(\mathbf{x}_j^{t_0-1} | \mathbf{x}_j^{t_0})$ (see eq. (4))
 - 14: **end for**
 - 15: $y_j = \hat{x}_j^0; j = j + 1$
 - 16: Sample $x_{j-1}^1 \sim q(x_{j-1}^1 | x_{j-1}^0); x_{j+N}^T \sim \mathcal{N}(0, I)$
 - 17: $\mathbf{x}_j^{\text{cont}} \leftarrow (x_{j-n^{\text{cont}}}^1, \dots, x_{j-1}^1)$
 - 18: $\mathbf{x}_j^s \leftarrow (x_j^s, \dots, x_{j+N}^s)$
 - 19: $\mathbf{u}_j \leftarrow (u_{j-n^{\text{cont}}}, \dots, u_{j+N})$
 - 20: **until** Completed
-

(RDLA), a novel approach that transforms the original noise schedule into a ladder with step size l (see fig. 2), enabling the simultaneous denoising of l frames from the same noise level. We introduce a new sequence of stepsize noise levels t_i^l according to the following rule:

$$t_i^l = t_0^l + (k + 1) \cdot l - 1, \quad kl \leq i < (k + 1)l - 1,$$

where $k \in \{0, \frac{N}{l} - 1\}$, $t_0^l \in \{1, \dots, l\}$.

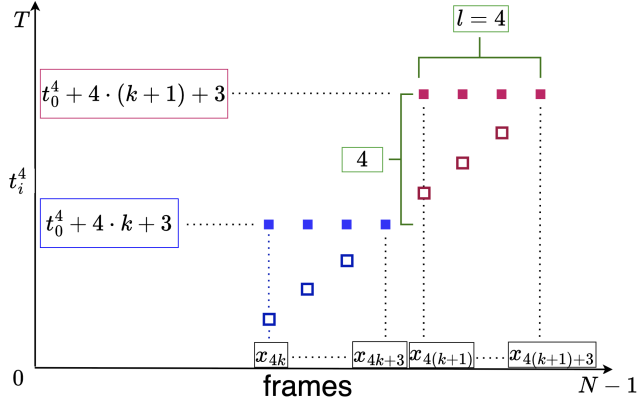


Figure 2: Rolling ladder steps k (blue bottom squares) and $k + 1$ (red upper squares) for the ladder step size $l = 4$ with corresponding noise level values and frames in the rolling window $\mathbf{x}_j^{t_0^l}$. The hollow squares of the corresponding color show the initial positions of the noise levels for a ladder of step size $l = 1$.

This modification allows multiple frames to be jointly denoised in each iteration, accelerating the process. The conventional rolling diffusion model can be seen as a special case of RDLA with $l = 1$. The process of constructing the ladder noise schedule $l = 4$ for steps k and $k + 1$ is illustrated in fig. 2. This proposed process can be viewed as a transformation from the noise schedule with $l = 1$ to the noise schedule with $l > 1$ ($l = 4$ for fig. 2). All noise levels within a constructing ladder step are set equal to the last noise level in that step. This design choice ensures a consistent step height across all levels of the ladder and guarantees that the last ladder step noise level equals to T . Thus, the sampling process in a rolling window starts from T , and has zero signal-to-noise ratio, which, as demonstrated in (Lin et al. 2024), is important for maintaining output quality.

During inference, RDLA processes l frames simultaneously by fully denoising an entire block at each iteration. At the same time, l new frames are initialized from Gaussian noise and appended to the rolling window, ensuring continuous sequence expansion. This modification follows algorithm 2, but with enhanced efficiency due to the structured noise scheduling of RDLA.

RDLA Training Strategy

To further enhance RDLA’s effectiveness, we introduce a progressive fine-tuning approach where the ladder step size is gradually increased during training (e.g., $l = 2, 4, \dots$). The

model architecture f_θ remains unchanged, but its weights are initialized from the previous iteration as l increases.

One key challenge is the loss of coherence between context and newly generated frames when l becomes large. To address this, we increase the context window size $N + n^{\text{cont}}$ while keeping the total rolling window length N fixed. To preserve the divisibility of T by N , we also needed to reduce T . This ensures stable training while maintaining performance across various step sizes.

Additionally, we introduce an inertial loss function $L_{\text{RDLA}}(\theta)$ to regularize the transition between ladder steps:

$$\mathbb{E}_j \mathbb{E}_{\mathbf{x}_j^0} \sum_{t_0^l=1}^l \mathbb{E}_{\mathbf{x}_j^{t_0^l}} \left[\sum_{n=0}^{N-1} \|x_{j+n}^0 - \hat{x}_{j+n}\|^2 - 2\lambda \sum_{n=0}^{N-1} \langle x_{j+n}^0 - \hat{x}_{j+n}, x_{j+n+1}^0 - \hat{x}_{j+n+1} \rangle \right] \quad (6)$$

where the second term penalizes abrupt changes between adjacent denoised frames, reducing jitter.

By implementing RDLA, we achieve a substantial reduction in inference time while maintaining high visual fidelity and temporal consistency. Empirical results (Section 4) demonstrate that RDLA accelerates gesture synthesis by up to 4× compared to standard rolling diffusion.

4 Experiments

To thoroughly examine the impact of our method, we integrate our progressive noise scheduling technique into multiple baseline models and conduct comparisons across two datasets: ZEGGS (Ghorbani et al. 2023) and BEAT (Liu et al. 2022). ZEGGS was chosen for its clean, high-quality motion capture data, recorded with a motion capture suit, ensuring precise gesture representation. Additionally, it includes diverse speaking styles, making it well-suited for evaluating stylistic consistency. In contrast, BEAT is one of the largest and most widely used datasets in the field, providing a broader range of conversational gestures. However, it contains more noise, presenting a greater challenge for generative models. Gestures in these datasets are represented in the BioVision Hierarchy (BVH) format and processed as vectors encoding joint positions and rotations in 3D space, along with additional features such as velocities. The processing method used in our work follows the baseline approaches.

This enables systematic evaluation of gesture-generation improvements across architectures. Our aim is to assess whether the method can serve as a generalizable framework for enhancing diverse gesture-synthesis models, ensuring robustness across varying data conditions. All evaluations used long sequences (1.5–2-min clips from ZEGGS/BEAT), from which metrics and user-study samples were derived.

Experimental Setup

Baseline approaches. As the primary baselines for our work, we selected state-of-the-art diffusion-based models for gesture generation: Taming Diffusion (Zhu et al.

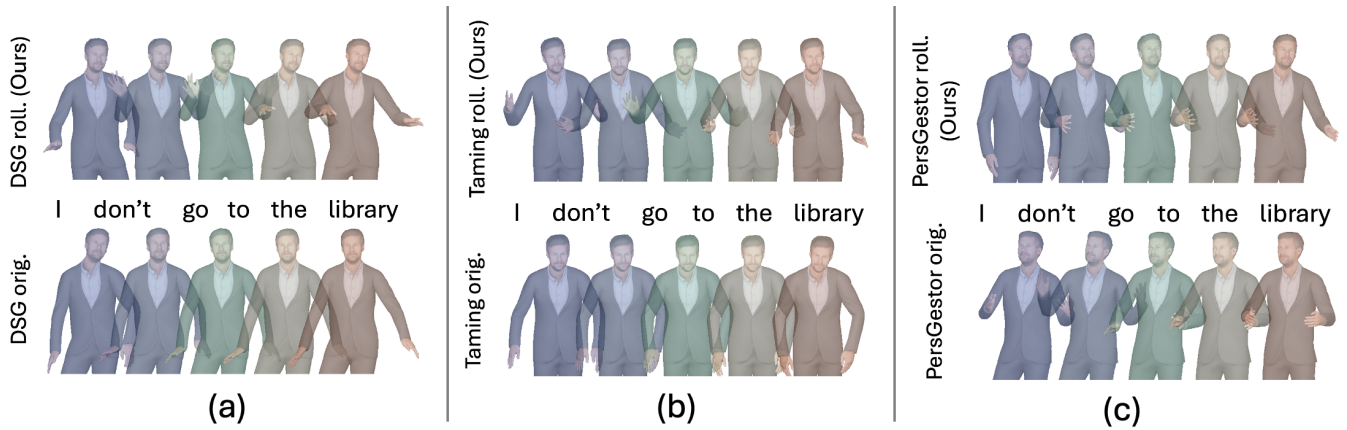


Figure 3: Qualitative Comparison on ZEGGS Dataset. Columns (a–c) are DiffuseStyleGesture, Taming, and PersonaGestor; the baseline is on the bottom row and our version on the top. Our approach generates a broader range of natural, diverse gestures that are tightly synchronized with the speech signal. Specifically, it responds to the negation “don’t” with a rejecting/withdrawing motion, whereas the baselines largely repeat neutral poses and miss this semantic cue.

2023), DiffuseStyleGesture (Yang et al. 2023) (DSG), PersonaGestor (Zhang et al. 2024) and DiffSHEG (Chen et al. 2024). These models were chosen because they represent leading approaches in data-driven gesture synthesis, each tackling different aspects of the task. Taming functions as a general-purpose co-speech gesture model, while DiffuseStyleGesture and DiffSHEG incorporate stylistic control through conditioning on predefined style labels. In contrast, PersonaGestor also accounts for stylistic variations, but derives them from features extracted directly from the audio rather than relying on explicit style labels. Notably, both DSG and Taming inherently employ direct conditioning on seed poses to ensure smooth transitions during generation. DiffSHEG, on the other hand, uses an outpainting strategy to concatenate adjacent clips, which also enables streaming generation. In each experiment, we adopt the same architecture as the baseline model, incorporating the modifications outlined in section 2. Since the DiffSHEG model was not originally developed with the ZEGGS dataset in mind, and its implementation relies on additional information unavailable in ZEGGS, we adapted and tested the model only on a single dataset where all necessary inputs were provided.

For consistency with prior work, we follow the DiffuseStyleGesture baseline setup, using six distinct speaking styles from the ZEGGS dataset (Happy, Sad, Neutral, Old, Angry, and Relaxed) across all three baselines. This ensures that our results remain comparable. For the BEAT dataset, we include all 30 available speaking styles, leveraging its extensive stylistic diversity to assess how well our method generalizes across a broader range of conversational gestures. All experiments were conducted on a workstation with an NVIDIA A40 GPU (48 GB), 126 GB RAM, 64-core CPU, and Ubuntu 22.04.

Evaluation metrics. To evaluate the quality of our generated gestures, we utilize the metrics FD_g , FD_k , Div_g , and Div_k introduced by Ng et al. (Ng et al. 2024). These metrics are computed in the space of 3D joint coordinates. The

Fréchet distance (FD) metrics, FD_g and FD_k , measure the similarity between the generated gestures and the real motion data. Specifically, FD_g evaluates the spatial distribution of poses, while FD_k assesses motion dynamics by analyzing frame-to-frame differences. Lower values indicate a closer resemblance to real-world gestures. The Diversity (Div) metrics, Div_g and Div_k , quantify the variability within the generated gestures. Higher diversity values suggest a richer and more varied set of gestures, preventing repetitive or overly uniform motion. Kinetic-based metrics help ensure that the generated gestures exhibit realistic movement patterns, maintaining a natural motion flow without sudden, unnatural position changes. Together, these evaluation measures provide a comprehensive assessment of both the realism and diversity of the generated co-speech gestures.

Training hyperparameters. Across all models, we retain most baseline hyperparameters to ensure fair comparison. However, the generation window length is a critical setting due to framework constraints. For the DSG rolling model, we use $N = 100$, with $n^{\text{cont}} = 8$ for ZEGGS and 30 for BEAT. Regularization is applied with weight decay $wd = 0.005$ and dropout 0.2 for ZEGGS, and $wd = 0.01$ for BEAT. For Taming, we adjust the generation window to $N = 50$, $n^{\text{cont}} = 4$, and increase training to 2000 epochs. In PersonaGestor, we use $N = 200$, $n^{\text{cont}} = 20$, and $wd = 0.01$ for both datasets. The rolling DiffSHEG model uses $N = 25$ and $n^{\text{cont}} = 9$ to match the baseline window length of 34. We preserve all inference settings and do not use resampling for last timesteps. Training is run for 700 epochs. All other hyperparameters remain unchanged.

RDLA experiments. To evaluate the effectiveness of Rolling Diffusion Ladder Acceleration (RDLA) in improving inference efficiency, we conducted a series of experiments on ZEGGS and BEAT dataset. To ensure a fair comparison, we applied RDLA to the DiffuseStyleGesture (DSG) model (Yang et al. 2023), which has demonstrated

Method	Div _g ↑	Div _k ↑	FD _g ↓	FD _k ↓
GT	272.34	213.97	–	–
DSG orig.	239.37 _{±2.4}	161.07 _{±3.3}	6393.99 _{±39.0}	14.24 _{±1.2}
DSG roll.	251.35 _{±0.8}	175.12 _{±1.8}	3831.35 _{±91.2}	8.08 _{±0.7}
DSG RDLA 2	222.25 _{±1.2}	173.76 _{±1.0}	5772.40 _{±63.9}	13.65 _{±0.1}
DSG RDLA 4	157.77 _{±0.5}	93.22 _{±0.6}	16791.03 _{±59.4}	42.19 _{±1.5}
Taming orig.	154.70 _{±1.2}	80.70 _{±0.8}	10784.86 _{±54.7}	418.85 _{±12.7}
Taming roll.	190.09 _{±1.2}	124.42 _{±1.0}	9064.00 _{±102.7}	353.62 _{±23.5}
PersGestor orig.	230.11 _{±6.0}	165.17 _{±1.9}	4060.36 _{±68.2}	11.12 _{±1.6}
PersGestor roll.	242.14 _{±7.8}	189.31 _{±2.7}	3936.75 _{±56.2}	9.14 _{±0.5}

Table 1: Results of quantitative analysis on ZEGGS dataset.

state-of-the-art performance among diffusion-based methods on these benchmarks.

A straightforward method for accelerating the sampling process is reducing the number of denoising steps per frame. As described in section 2, in our framework, each frame undergoes a predefined number of denoising steps s before the rolling window shifts forward in time. In RDLA we reducing this number from s to 1 leading to a total inference step count of $T_r = N$, where N is the total number of frames in sliding window. As a result, we report RDLA results with a reduced number of denoising steps ($T_r = 100$) using DDPM (Ho, Jain, and Abbeel 2020) sampling strategy, while additional experiments with varying T_r values and alternative sampling strategies are provided in the Supplementary Material.

Beyond reducing the number of sampling steps, we leveraged RDLA to accelerate inference in the temporal dimension. By constructing a denoising ladder with step size l , we simultaneously denoised l frames at each iteration, achieving an l -fold acceleration. Our experiments applied RDLA to DSG on ZEGGS and BEAT datasets with ladder step sizes $l \in \{2, 4\}$. For RDLA fine-tuning we used model’s weights obtained in section 4 for $l = 1$ as initial and continue training for 3000 epochs with learning rate = $1e - 7$, dropout = 0.1, wd = 0.01, and all remaining parameters followed the original DSG setup. During our experiments, we observed that RDLA performs best when the context length increases proportionally with the ladder step size. Accordingly, we report results using a context length of $n^{\text{cont}} = 28$ for ZEGGS and $n^{\text{cont}} = 50$ for BEAT. Minor tremor that we observed for larger l is mitigated via the temporal-smoothness loss (eq. (6)) and an On-the-Fly Smoothing strategy, though the latter was not used in reported results. Further details are provided in the Supplementary Material.

Method	Div _g ↑	Div _k ↑	FD _g ↓	FD _k ↓
GT	279.52	116.17	–	–
DSG orig.	201.83 _{±8.4}	63.78 _{±5.7}	37062.19 _{±459.9}	77.28 _{±1.9}
DSG roll.	241.50 _{±9.8}	76.09 _{±6.1}	21441.91 _{±434.0}	69.23 _{±5.7}
DSG RDLA 2	190.41 _{±9.9}	58.72 _{±8.4}	17309.63 _{±327.1}	56.24 _{±5.5}
DSG RDLA 4	207.47 _{±9.1}	83.08 _{±8.7}	18739.77 _{±467.2}	59.02 _{±4.9}
Taming orig.	139.64 _{±0.6}	57.40 _{±0.5}	11632.64 _{±158.7}	67.94 _{±3.7}
Taming roll.	169.59 _{±1.6}	73.41 _{±1.4}	9835.27 _{±190.7}	74.15 _{±9.2}
PersGestor orig.	173.05 _{±3.7}	51.54 _{±1.6}	11936.82 _{±324.5}	143.37 _{±5.1}
PersGestor roll.	181.12 _{±3.2}	62.50 _{±1.5}	10815.76 _{±350.4}	130.14 _{±5.2}
DiffSHEG orig.	247.47 _{±3.0}	108.7 _{±2.0}	13547.4 _{±835}	4517.2 _{±172}
DiffSHEG roll.	294.77 _{±4.2}	110.8 _{±2.1}	10201.9 _{±1586}	1696.6 _{±307}

Table 2: Results of quantitative analysis on BEAT dataset.

Results

The quantitative evaluation on ZEGGS and BEAT datasets is summarized in table 1 and table 2 (values are reported as mean_{±std}, where the mean and standard deviation are computed over five independent runs). Across both datasets, rolling variants generally outperform their original counterparts, with statistically significant gains in Fréchet distance and diversity (paired t -test, $p < 0.05$), except FD_k for Taming. DSG rolling shows the most notable improvement, while other rolling versions also exhibits enhanced performance. Since our method surpasses baselines that utilize direct conditioning on seed frames or outpainting strategy, it clearly demonstrates the superiority of our approach for generating coherent and high-quality motion. These results confirm that our rolling diffusion framework not only improves generation quality but does so in a model-agnostic manner. By applying our method across multiple state-of-the-art baselines without modifying their core architectures, we show that the framework generalizes well and can serve as a plug-and-play solution for streaming gesture synthesis.

Results in table 1 show that a 2× acceleration ($l = 2$) yields only a modest performance drop on ZEGGS, with metrics close to the baseline. On BEAT (table 2), RDLA improves Fréchet distances for both $l = 2$ and $l = 4$, and further boosts kinetic diversity at 4× acceleration. Thus, RDLA provides substantial speedups while preserving or even improving perceptual and motion quality. The differing trends stem from dataset characteristics: ZEGGS is more expressive, whereas BEAT is calmer; increasing l introduces smoothing that is more disruptive for ZEGGS.

Method	l	Steps	FPS	Latency (sec)
DSG orig. (baseline)	–	1000	10	8
DSG roll. (ours)	1	1000	10	0.06
DSG roll. (ours)	1	100	70	0.006
DSG RDLA (ours)	2	100	120	0.003
DSG RDLA (ours)	4	100	200	0.002

Table 3: Generation speed comparison.

Acceleration findings. On a NVIDIA A40 (48 GB), our rolling DSG and RDLA variants greatly reduce latency without sacrificing throughput, as summarized in table 3.

Qualitative comparisons. Figure 3 presents representative keyframes comparing our method with baseline versions of DSG, Taming, and PersonaGestor. As illustrated, our approach produces more expressive and semantically aligned gestures, such as clear withdrawal motions in response to negations. Full video visualizations are provided in the Supplementary Material for a comprehensive qualitative comparison.

User Study

To assess the quality of our generated co-speech gestures, we conducted a user study using pairwise comparisons between our model and a baseline. We selected the ZEGGS dataset for its clear and expressive gestures, which allow for a precise evaluation of movement quality, stylistic consistency, and synchronization with speech. We used the DSG model as a baseline for comparison.

Participants were shown pairs of 15-second videos, each synchronized with the same audio but generated using different models. Both videos were displayed simultaneously, with their positions randomized in each trial to eliminate possible bias towards one side. The participants were asked to compare the two animations and rate them based on style consistency, naturalness and fluidity of animations, audio-animation synchronization, presence of technical issues (such as the gluing effect). The rating could take values $\{-2, -1, 0, 1, 2\}$ where -2 indicates a strong preference for the baseline, -1 – slight baseline preference, 0 means no noticeable differences, 1 indicates a slight preference for our model and 2 indicates a strong preference for our model. To ensure the reliability of our evaluators, we included some videos multiple times and verified that the assessors provided consistent ratings before including their responses in the final analysis. The study was carried out on 60 pairs of video (10 per style in six different styles). Twenty-two professional assessors trained to work with video data were recruited to participate.

The distribution of the user study results is shown in fig. 4 (Left). Our rolling modification of DSG significantly outperforms the original DSG, aligning well with the quantitative evaluation results; the preference is statistically significant (two-sided Wilcoxon signed-rank, $p < 0.05$). To compare RDLA with our original method, we conducted a user study between DSG rolling and RDLA with $l = 2$. The distribution of the RDLA user study results is shown in fig. 4

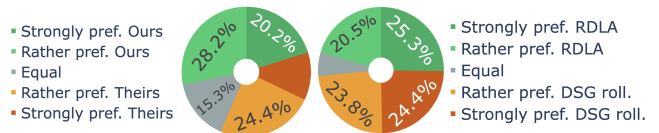


Figure 4: User study results. **Left:** “Ours” means DSG rolling modification, “Theirs” means original DSG. In total 48.4% of participants preferred our model while 36.3% preferred original DSG. **Right:** RDLA user study results. In total 48.2% of participants preferred DSG rolling model while 45.7% preferred RDLA.

(Right). The RDLA approach is only slightly inferior to our original method, which is consistent with the quantitative findings.

5 Ablation Study

In our ablation study (see details in the Supplementary Material), we analyzed key components influencing the Accelerated Rolling Diffusion framework’s performance. First, we evaluated the necessity of minimal noise in context frames. Results showed that incorporating minimal noise ($\sigma_1^2 = 0.00004$) into context frames significantly improved model robustness and prevented overfitting, thus enhancing generalization. Next, we assessed loss weighting strategies, comparing clamped-SNR (Barrault et al. 2024) weighting versus uniform weighting. Uniform weighting ($a(t) = 1$ in eq. (5)) provided a simpler, more stable training process without compromising performance, confirming its suitability for efficient sequential generation. Additionally, we conducted ablations on the components of RDLA: inertial loss regularization (eq. (6)), progressive fine-tuning, on-the-fly smoothing (OFS, not used in the reported results). Removing $L_{RDLA}(\theta)$ increased motion jitter, underscoring its importance for coherent motion. Progressive fine-tuning was crucial, as models without it exhibited significant quality and diversity loss. Omitting OFS only slightly degraded gesture smoothness. This strategy enabled effective adaptation to increased ladder step sizes, preserving performance.

6 Conclusion

We introduced Accelerated Rolling Diffusion, a framework for real-time, high-quality co-speech gesture generation. Experiments on ZEGGS and BEAT, plus user studies, demonstrate strong generalizability and consistent gains across diffusion-based gesture models. Our method produces continuous, temporally coherent motion without post-processing. To boost efficiency, we proposed Rolling Diffusion Ladder Acceleration (RDLA), which uses structured noise scheduling to reduce sampling steps and achieve up to 4× speedup while maintaining realism. RDLA is a promising direction for efficient diffusion acceleration, with potential extensions to other temporal modalities such as text and video.

References

- Barrault, L.; Duquenne, P.-A.; Elbayad, M.; Kozhevnikov, A.; Alastruey, B.; Andrews, P.; Coria, M.; Couairon, G.; Costa-jussà, M. R.; Dale, D.; et al. 2024. Large Concept Models: Language Modeling in a Sentence Representation Space. *arXiv e-prints*.
- Chen, J.; Liu, Y.; Wang, J.; Zeng, A.; Li, Y.; and Chen, Q. 2024. Diffshg: A diffusion-based approach for real-time speech-driven holistic 3d expression and gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7352–7361.
- Chen, S.; Wang, C.; Chen, Z.; Wu, Y.; Liu, S.; Chen, Z.; Li, J.; Kanda, N.; Yoshioka, T.; Xiao, X.; et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6): 1505–1518.
- Ghorbani, S.; Ferstl, Y.; Holden, D.; Troje, N. F.; and Carbonneau, M.-A. 2023. ZeroEGGS: Zero-shot Example-based Gesture Generation from Speech. In *Computer Graphics Forum*, volume 42, 206–216. Wiley Online Library.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.
- Lin, S.; Liu, B.; Li, J.; and Yang, X. 2024. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 5404–5411.
- Liu, H.; Zhu, Z.; Iwamoto, N.; Peng, Y.; Li, Z.; Zhou, Y.; Bozkurt, E.; and Zheng, B. 2022. Beat: A large-scale semantic and emotional multi-modal dataset for conversational gestures synthesis. In *European conference on computer vision*, 612–630. Springer.
- Ng, E.; Romero, J.; Bagautdinov, T.; Bai, S.; Darrell, T.; Kanazawa, A.; and Richard, A. 2024. From audio to photoreal embodiment: Synthesizing humans in conversations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1001–1010.
- Nyatsanga, S.; Kucherenko, T.; Ahuja, C.; Henter, G. E.; and Neff, M. 2023. A comprehensive review of data-driven co-speech gesture generation. In *Computer Graphics Forum*, volume 42, 569–596. Wiley Online Library.
- Ruhe, D.; Heek, J.; Salimans, T.; and Hoogeboom, E. 2024. Rolling Diffusion Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 42818–42835. PMLR.
- Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.
- Vu, E.; Boiarov, A.; and Vetrov, D. 2025. Streaming Generation of Co-Speech Gestures via Accelerated Rolling Diffusion. *arXiv preprint arXiv:2503.10488*.
- Yang, S.; Wu, Z.; Li, M.; Zhang, Z.; Hao, L.; Bao, W.; Cheng, M.; and Xiao, L. 2023. DiffuseStyleGesture: Stylized Audio-Driven Co-Speech Gesture Generation with Diffusion Models. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 5860–5868.
- Zhang, F.; Wang, Z.; Lyu, X.; Zhao, S.; Li, M.; Geng, W.; Ji, N.; Du, H.; Gao, F.; Wu, H.; et al. 2024. Speech-driven personalized gesture synthetics: Harnessing automatic fuzzy feature inference. *IEEE Transactions on Visualization and Computer Graphics*.
- Zhu, L.; Liu, X.; Liu, X.; Qian, R.; Liu, Z.; and Yu, L. 2023. Taming diffusion models for audio-driven co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10544–10553.