Past as State, Present as Attention: Persistent-State Blockwise Flow Matching for Long-Horizon Co-Speech Motion Generation

Yi Liu[1,3] **Xiangyue Zhang**[2] **Jianfang Li**[3]* **Jia Ma**[1] **Yisheng He**[3] **Jianqiang Ren**[3]
[1] Tongji University [2] The University of Tokyo [3] Alibaba Group
{liuyi61,2432018}@tongji.edu.cn x-zhang@mi.t.u-tokyo.ac.jp
{wuhui.ljf,heyisheng.hys,jianqiang.rjq}@alibaba-inc.com

Project website: https://ly403.github.io/PASPA/

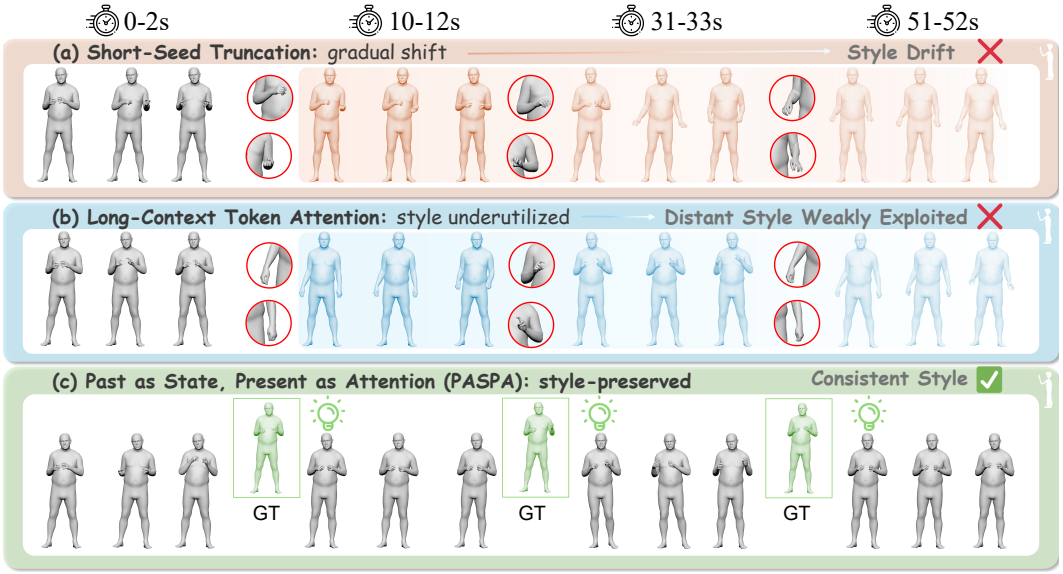

Figure 1: Long-horizon co-speech motion generation (52 s) from the same speech and speaker, with the GT style keeping both hands raised near the chest. (a) Short-Seed Truncation matches the style initially but monotonically drifts: hands gradually drop. (b) Long-Context Token Attention retains full history but struggles to maintain a consistent style, matching or deviating regardless of temporal distance. (c) PASPA (ours) preserves the raised-hands style throughout, closely matching the GT.

Abstract

Co-speech motion generation remains challenging over long horizons due to the need for temporal coherence and consistent style. Existing methods typically generate long motions segment by segment, either conditioning on a short seed of recent frames, which gradually drifts, or retaining ever-longer explicit histories, which are costly to maintain and hard to learn from limited data. We trace these limitations to a key asymmetry between past and present: past motion carries slowly varying style that should be compactly summarized, whereas the current segment exhibits fast-changing local dynamics that require fine-grained modeling. This asymmetry motivates a simple ***P**ast-**A**s-**S**tate, **P**resent-as-**A**ttention* principle for long-horizon

*Corresponding author.

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

co-speech motion generation. Following this principle, we propose **PASPA**, which summarizes past history with an inter-block persistent state and models the current segment with intra-block bidirectional self-attention. We instantiate PASPA within a blockwise flow matching framework, with multi-block supervision for effective training over variable histories, prefill-decode state caching for efficient rollout, and hybrid classifier-free guidance for enhanced history and speech conditioning. Extensive public-benchmark experiments show that PASPA achieves state-of-the-art FGD performance and preserves motion style across long-sequence generation.

1 Introduction

Humans naturally accompany speech with expressive body movements that convey emphasis, emotion, and communicative intent beyond words alone [53]. Co-speech motion generation aims to synthesize realistic full-body motions that are temporally aligned and semantically consistent with speech [56]. It is crucial to applications like digital humans [99], virtual assistants [87], and animated characters [24].

Early co-speech motion generation often relies on direct speech-to-motion mapping [86], but the inherent one-to-many nature of speech-to-motion correspondence has made generative modeling increasingly important [38]. Accordingly, diffusion models [27] and flow matching [42, 50], the prevailing generative paradigms, have been widely adopted in this field.[2] While they improve short-horizon realism and controllability [48, 92], long-horizon co-speech motion generation remains challenging, as it requires preserving temporal coherence [75] and speaker style [1] (see Fig. 1).

For long-horizon generation, most methods proceed segment by segment, making inter-block history propagation the central challenge. A common strategy is to use only a seed (*i.e.*, the previous segment's last few frames; Fig. 2a) [45, 89, 92] as condition. With such limited history, temporal coherence and style gradually drift over time (see Fig. 1a) [75]. Recent streaming methods exploit longer histories [8, 75, 77], but retain contexts like frames or tokens (see Fig. 2b). Maintaining such explicit histories incurs growing memory and compute costs as horizons extend, especially with attention-based backbones [74]. More critically, learning to exploit ever longer explicit histories is difficult given the limited data in co-speech motion generation [45], often leading to unstable style preservation (see Fig. 1b). These observations raise a fundamental question: *is there a more effective way to represent and propagate the past than explicitly attending to raw historical sequences?*

We answer this affirmatively: the past and present play distinct roles in motion generation and should be represented accordingly. Past history primarily carries slowly varying stylistic traits (*e.g.*, gesture amplitude and spatial preference) [1, 3], which should be compactly summarized rather than stored verbatim. By contrast, local motion dynamics are fast-changing, making full-resolution modeling of the current block more suitable for capturing fine-grained motion details. This asymmetry motivates a simple ***P**ast-**A**s-**S**tate, **P**resent-as-**A**ttention* principle: the past should be compressed as a persistent state that is carried across segments, while the present should be modeled with full attention for expressive generation (see Fig. 2c). Guided by this principle, we propose **PASPA**, which achieves temporally coherent and style-consistent long-horizon co-speech motion generation (see Fig. 1c).

In PASPA, past history and the current block are handled by two complementary mechanisms. For past history, we design a block-level recurrent state mechanism built upon Gated DeltaNet [83, 84], which progressively summarizes preceding blocks into a persistent state and provides a selective-retention, temporal-decay inductive bias for slowly evolving motion style. For the current block, we use blockwise self-attention [60, 74] to enable full bidirectional interaction for fine-grained modeling. We integrate them into a hybrid PASPA module, which forms the core component of our model.

We build PASPA on a semi-autoregressive blockwise flow matching (BFM) framework inspired by recent block diffusion paradigms [6, 95], and develop tailored training and inference protocols. For training, we apply distinct noise levels to multiple blocks and supervise their denoising in parallel, yielding dense multi-block supervision and exposing each block to states summarizing variable-length histories. For inference, PASPA separates state prefilling from iterative decoding and reuses cached historical states for efficient rollout. To improve generation quality, we introduce a hybrid classifier-free guidance [26] that augments audio-conditioned guidance with history-drop guidance [66], enabling explicit control over historical dependence while preserving speech alignment.

[2] We use diffusion and flow matching interchangeably in this paper due to their mathematical equivalence [4, 43].

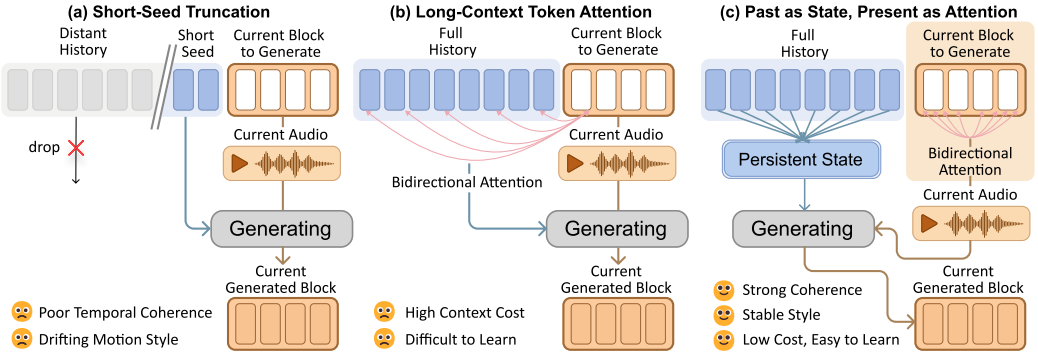

Figure 2: History propagation in co-speech motion generation. (a) Short-Seed Truncation keeps only recent frames. (b) Long-Context Token Attention retains history as tokens for global attention. (c) PASPA compresses history into a persistent state and applies full attention within the current block.

Our main technical contributions are: (1) We identify a key asymmetry in the roles of past and present in co-speech motion generation, and formulate a simple *past-as-state, present-as-attention* principle for long-horizon modeling. (2) Following this principle, we propose **PASPA**, a persistent-state method built on blockwise flow matching, whose core module couples inter-block persistent state modeling with intra-block attention. (3) We develop tailored training and inference schemes, including multi-block supervision, prefill-decode decoupling with state caching, and hybrid classifier-free guidance.

2 Related Work

Co-Speech Motion Generation. The task synthesizes speech-aligned body movements. Its one-to-many nature motivates generative paradigms such as diffusion models (DMs) [27, 67] and VQ-VAE-based generation [73] for realism and diversity [2, 3, 24, 93, 98]. Further enhancements for semantic controllability include hierarchical designs and language-based guidance [9, 41, 49, 59, 91]. VQ-VAE-based methods [45, 51, 85, 89, 90] generate motions through learned codebooks, but often suffer from limited diversity due to discrete code constraints. DM-based methods [5, 11, 13, 54, 92] better support diverse motion synthesis and have become a dominant paradigm in this field.

Diffusion Models and Flow Matching. DMs [27] and Flow Matching (FM) [42, 50] have achieved strong results in image generation [14, 17]. Advances include novel architectures [20, 58, 78], conditional guidance [26, 32, 37], and accelerated sampling [50, 65]. They also extend to sequential domains (*e.g.*, video) via autoregressive (AR) factorization for variable-length rollout [6, 10, 62]. Representative approaches include Block Diffusion [6, 95], which combines intra-block denoising with inter-block autoregression, and Diffusion Forcing [10], which assigns independent per-token noise levels to bridge next-token prediction [61] and full-sequence diffusion. Related advances mitigate exposure bias [29], enable real-time generation [47], and support flexible history conditioning [66]. This AR diffusion paradigm has also been adapted to motion generation [8, 25, 77, 96].

Long-Context Sequence Modeling. Attention [74] dominates long-context modeling but scales quadratically with sequence length [70] and lacks inductive biases for data-limited regimes [18]. Alternatives like state space models [16, 22] and linear attention [34, 82, 83] replace pairwise attention with linear recurrent state updates, improving efficiency and introducing stronger temporal inductive bias. Specifically, Gated DeltaNet (GDN) [84] is highly relevant to us, as its gating and delta-rule updates yield a temporal-decay bias that we leverage to model continuous motion dynamics. Such long-context designs have also been adopted in motion generation (*e.g.*, Motion Mamba [94]).

3 Methodology

3.1 Preliminary

Co-Speech Motion Generation. This task synthesizes motion $\boldsymbol{x} \in \mathbb{R}^{T \times C}$ and facial expression $\boldsymbol{e} \in \mathbb{R}^{T \times C_e}$ from a speech waveform $\boldsymbol{a} \in \mathbb{R}^{T_a}$. Following the SMPL-X [52, 57] parameterization, the

motion sequence $\boldsymbol{x}$ consists of $J = 55$ joint rotations and a 3D global translation. The joint rotations are converted from axis-angle to the 6D representation (rot6d) [97], yielding $C = 6J + 3 = 333$. Following FLAME [40], the expression $\boldsymbol{e}$ is represented by $C_e = 100$ FLAME coefficients. For long-horizon generation, co-speech motion is commonly generated segment by segment [45, 89, 92]. Accordingly, the motion and expression sequences are partitioned into N blocks of length K, denoted by $\boldsymbol{x} = [\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N]$ and $\boldsymbol{e} = [\boldsymbol{e}^1, \ldots, \boldsymbol{e}^N]$, where $\boldsymbol{x}^n \in \mathbb{R}^{K \times C}$ and $\boldsymbol{e}^n \in \mathbb{R}^{K \times C_e}$ denote the n-th motion and expression blocks, respectively. The waveform is segmented analogously into N blocks of length K_a (*i.e.*, $\boldsymbol{a} = [\boldsymbol{a}^1, \ldots, \boldsymbol{a}^N]$), where $\boldsymbol{a}^n \in \mathbb{R}^{K_a}$ denotes the n-th speech segment.

Flow Matching. Our method builds on FM [42, 50], a prevailing generative paradigm. It constructs a probability path by linearly interpolating between clean motion $\boldsymbol{x}_0$ and noise $\boldsymbol{z}$, governed by time t:

$$\boldsymbol{x}_t = (1 - t)\boldsymbol{x}_0 + t\boldsymbol{z}, \quad \boldsymbol{z} \sim \mathcal{N}(0, \boldsymbol{I}), \quad t \in [0, 1], \tag{1}$$

where $\mathcal{N}$ denotes the Gaussian distribution. Typically, FM learns the path velocity and samples by solving the corresponding probability flow ordinary differential equation [42, 50] of this path.

3.2 Past-as-State, Present-as-Attention Module

In segment-by-segment motion generation [45, 92], past history and the current block serve different roles. Empirically, style cues remain correlated over long timescales, whereas local motion dynamics decay rapidly (see Fig. 3). Motivated by this asymmetry, PASPA (see Fig. 4) decouples their processing: an inter-block persistent state compactly summarizes past history, while an intra-block bidirectional attention models the current block via dense interaction.

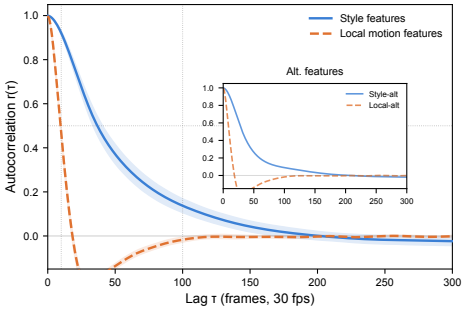

Figure 3: Autocorrelation $r(\tau)$ on BEAT2, averaged across speakers with 95% confidence intervals (shaded). Style features remain correlated beyond 100 frames, whereas local features decay rapidly. Inset: raw windowed pose statistics. See Sec. A for more details.

3.2.1 Inter-Block Persistent State.

Why Persistent State for the Past? Past history carries stable style and long-range tendencies. Such information is abstract and sparse: its effective content lies in a few persistent components, and token-level attention would waste capacity on transient fluctuations. Thus, the past should be selectively condensed into a persistent state, not indiscriminately preserved. We achieve this via Gated DeltaNet (GDN) [84], whose gating [82] and delta-rule [63, 76] updates suppress fluctuations and accumulate informative trends, yielding a strong inductive bias for history propagation in our data-limited setting.

Blockwise Gated DeltaNet (BGDN). GDN is token-recurrent, but our generation proceeds block by block. We thus propose a blockwise GDN (BGDN, Fig. 4a), driven by a tailored Block-Causal GDN kernel. Inside the BGDN module (Fig. 4 (a.1)), we take the noisy current block as input and the encoded clean history as the context. Following GDN, we derive queries $\boldsymbol{q}_i \in \mathbb{R}^{d_k}$ from the input, with keys $\boldsymbol{k}_i \in \mathbb{R}^{d_k}$, values $\boldsymbol{v}_i \in \mathbb{R}^{d_v}$, and gates $\alpha_i, \beta_i \in (0, 1)$ from the context, omitting GDN's temporal mixing to prevent cross-block leakage. Within the kernel, these variables update a persistent state $\boldsymbol{S}_i \in \mathbb{R}^{d_v \times d_k}$ and yield outputs $\boldsymbol{o} \in \mathbb{R}^{d_v}$. $\boldsymbol{o}$ are then projected to produce final outputs.

Inside the kernel (Fig. 4 (a.2)), we retain GDN's standard state update to accumulate histories:

$$\boldsymbol{S}_i = \boldsymbol{S}_{i-1} \left(\alpha_i \left(\boldsymbol{I} - \beta_i \boldsymbol{k}_i \boldsymbol{k}_i^\intercal\right)\right) + \beta_i \boldsymbol{v}_i \boldsymbol{k}_i^\intercal, \tag{2}$$

where α_i adaptively decays irrelevant fluctuations, and β_i governs the delta-rule update for selective information injection. However, applying GDN's dynamic read mechanism to blockwise generation would cause intra-block future leakage. We therefore restrict reads to the frozen block-initial state:

$$\text{GDN: } \boldsymbol{o}_i = \boldsymbol{S}_i \boldsymbol{q}_i, \qquad \text{BGDN: } \boldsymbol{o}_i = \boldsymbol{S}^0_{[n]} \boldsymbol{q}_i \quad \forall (n-1)K < i \leq nK, \tag{3}$$

where $\boldsymbol{S}^r_{[n]}$ $(0 \leq r \leq K)$ denotes the state after processing the r-th token of block n. Instead of reading the continuously updated state $\boldsymbol{S}_i$, all queries within block n exclusively read from the block-start state $\boldsymbol{S}^0_{[n]}$ (also the previous block's final state $\boldsymbol{S}^K_{[n-1]}$). This formulation mathematically guarantees that each block only conditions on strict past history, ensuring the blockwise causality.

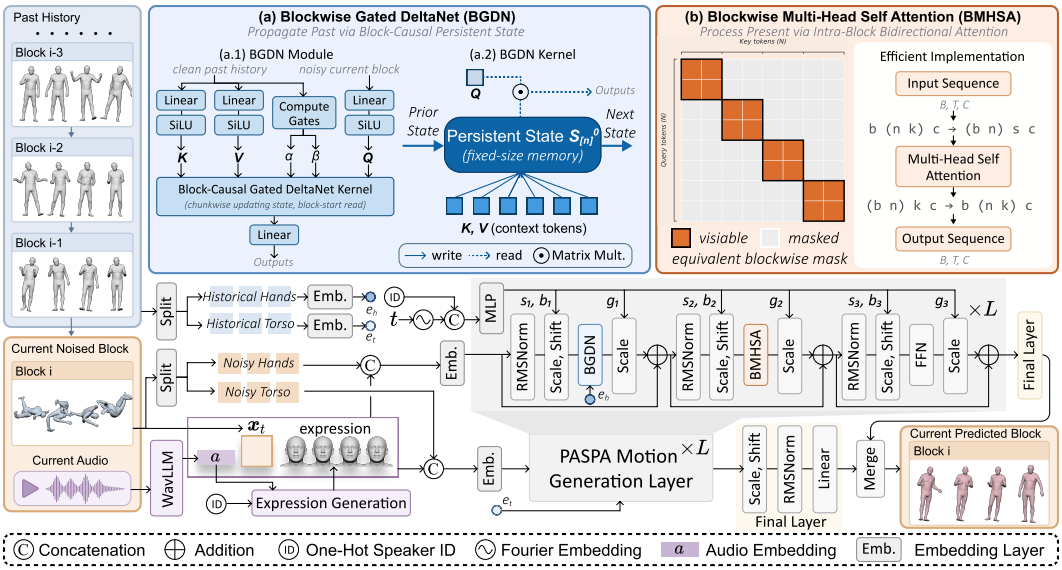

Figure 4: Network architecture of PASPA. (a) **BGDN** propagates history through a persistent state $\boldsymbol{S}_{[n]}^0$ via a block-causal GDN kernel. (b) **BMHSA** captures local dynamics via efficient block-diagonal bidirectional attention. The overall network (bottom) sequentially applies BGDN and BMHSA.

Efficient Implementation. GDN admits an efficient chunkwise parallel form [80, 84] (detailed in Sec. B). Since we leave the state update of GDN unchanged, its parallel computation requires no modification. The read, however, must be adapted to enforce block-causality. Specifically, the computational chunk size is fixed to the generation block size K, and all tokens in block n exclusively share the block-start state $\boldsymbol{S}_{[n]}^0$ (*i.e.*, the previous block's final state $\boldsymbol{S}_{[n-1]}^K$). This yields:

$$\boldsymbol{O}_{[n]}^{\text{BGDN}} = \boldsymbol{Q}_{[n]} \left(\boldsymbol{S}_{[n]}^0\right)^{\intercal} \in \mathbb{R}^{K \times d_v}, \qquad \boldsymbol{Q}_{[n]} \in \mathbb{R}^{K \times d_k}, \quad \boldsymbol{S}_{[n]}^0 = \boldsymbol{S}_{[n-1]}^K \in \mathbb{R}^{d_v \times d_k}. \tag{4}$$

This reduction to a dense matrix multiplication makes the read highly amenable to Tensor Core acceleration, for which we implement a custom Triton [72] kernel based on FLA [80].

3.2.2 Intra-Block Bidirectional Attention.

Why Bidirectional Attention for the Present? The current block carries fast-changing dynamics, such as velocities and fine-grained articulations, which demand dense modeling of local motion details. Correspondingly, BGDN shares a common block-start state, leaving token-to-token interactions within the current block unmodeled. We assign this role to a dedicated bidirectional self-attention module [74] rather than a causal one: although frames within a block follow a micro-level temporal order, each token still benefits from both preceding and subsequent local context within the segment.

Blockwise Multi-Head Self-Attention (BMHSA). Applying bidirectional self-attention [74] across the entire sequence naively would break the block-level semi-autoregression. We therefore propose BMHSA, which permits bidirectional interaction within each block while strictly forbidding any cross-block interaction. Given an input sequence $\boldsymbol{X} \in \mathbb{R}^{NK \times d}$, the output is computed as follows:

$$\boldsymbol{O} = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}} + \boldsymbol{B}^{\text{blk}}\right)\boldsymbol{V}\boldsymbol{W}_O, \quad \boldsymbol{B}_{ij}^{\text{blk}} = \begin{cases} 0 & \lceil i/K \rceil = \lceil j/K \rceil \\ -\infty & \text{otherwise} \end{cases}, \tag{5}$$

where $i, j \in \{1, \ldots, NK\}$ are token indexes, $\boldsymbol{B}^{\text{blk}} \in \{0, -\infty\}^{NK \times NK}$ is the block-diagonal mask (see Fig. 4b). We compute the query, key, and value matrices $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{NK \times d}$ as $\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q$, $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K$, and $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V$, respectively, with rotary position embedding (RoPE) [68] applied to $\boldsymbol{Q}$ and $\boldsymbol{K}$. Here, $\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V, \boldsymbol{W}_O \in \mathbb{R}^{d \times d}$ are projection matrices, with $\boldsymbol{W}_O$ used for output projection. The formulation above is single-head, with the multi-head extension following [74].

Efficient Implementation. Rather than materializing $\boldsymbol{B}$, we reshape $\boldsymbol{X}$ from $\mathbb{R}^{NK\times d}$ to $\mathbb{R}^{N\times K\times d}$ and apply standard bidirectional attention, treating the block index as batch (see Fig. 4b). This yields the same output as Eq. (5) while reducing the complexity from $\mathcal{O}((NK)^2 d)$ to $\mathcal{O}(NK^2 d)$.

3.2.3 Network Architecture.

We instantiate our PASPA layer following the DiT design [58] (see Fig. 4) with two modifications. Structurally, we replace the self-attention sublayer with two sequential sublayers: BGDN and BMHSA. For adaptive modulation, we replace AdaLN with AdaRMSNorm [15], where the modulation parameters are predicted from the Fourier-embedded timestep t [69] and the speaker ID embedding. Our architecture follows GlobalDiff [92], but uses our PASPA layer. We first use a frozen audio-to-expression network to obtain expressions from WavLM features [12] and speaker ID. Motions are represented in global rotation space and split into torso (with translation) and hands branches. For each branch, the noisy stream embeds body-part motion with full noisy motion, audio features, and expressions as tokens for the stacked PASPA layers, while the history stream encodes clean past motion into each BGDN module ($\boldsymbol{e}_h, \boldsymbol{e}_t$ in Fig. 4). The branch outputs are merged to the full motion.

3.3 Blockwise Flow Matching

Inspired by recent block diffusion paradigms [6, 95], we formalize segment-by-segment motion generation as semi-autoregressive process with block-causal factorization $p(\boldsymbol{x} \mid \boldsymbol{a}, \boldsymbol{e}) = \prod_{n=1}^{N} p(\boldsymbol{x}^n \mid \boldsymbol{x}^{<n}, \boldsymbol{a}^n, \boldsymbol{e}^n)$. Here, $\boldsymbol{x}^{<n} = [\boldsymbol{x}^1, \ldots, \boldsymbol{x}^{n-1}]$ provides clean history, $\boldsymbol{x}^n$ is the current denoising target, and $\boldsymbol{a}^n, \boldsymbol{e}^n$ are the aligned speech and expression condition of block n. We precompute $\boldsymbol{e}^n$ from $\boldsymbol{a}^n$ via an audio-to-expression network (Sec. 3.2.3). This process applies inter-block autoregression for long-horizon causality and intra-block bidirectional modeling for dense local dynamics.

Multi-Block Supervision. To realize the semi-autoregressive factorization, we instantiate the FM interpolation in Eq. (1) for each block, allowing every block to carry its own timestep $t^n \in [0, 1]$:

$$\boldsymbol{x}_{t^n}^n = (1 - t^n)\boldsymbol{x}_0^n + t^n \boldsymbol{z}^n, \qquad \boldsymbol{z}^n \sim \mathcal{N}(0, \boldsymbol{I}), \qquad \forall n = 1, \ldots, N, \tag{6}$$

where $\boldsymbol{x}_0^n$ is the n-th clean block.Following common practice in motion generation [71, 92], we train the network $\mathcal{F}_\theta$ to directly predict each clean block $\boldsymbol{x}_0^n$, rather than the velocity in FM [42]. The block-causal $\mathcal{F}_\theta$ performs dense parallel denoising over the entire sequence in a single forward pass:

$$\hat{\boldsymbol{x}}_0 = [\hat{\boldsymbol{x}}_0^1, \hat{\boldsymbol{x}}_0^2, \ldots, \hat{\boldsymbol{x}}_0^N]^\intercal = \mathcal{F}_\theta([\boldsymbol{x}_{t^1}^1, \boldsymbol{x}_{t^2}^2, \ldots, \boldsymbol{x}_{t^N}^N]^\intercal, \boldsymbol{x}_0, \boldsymbol{t}, \boldsymbol{a}), \quad \boldsymbol{t} = [t^1, t^2, \ldots, t^N]^\intercal, \tag{7}$$

where $\boldsymbol{a}$ is the aligned speech and $\boldsymbol{x}_0 = [\boldsymbol{x}_0^1, \ldots, \boldsymbol{x}_0^N]^\intercal$ is encoded as ground-truth history following teacher forcing [55, 64]. BGDN routes only $\boldsymbol{x}_0^{<n}$ into block n's prediction to preserve block-causality (Sec. 3.2.1). Unlike per-block supervision [45, 89, 92], each iteration accumulates loss over all blocks, yielding $N\times$ denser supervision. More critically, heterogeneous block timesteps expose the model to diverse current-block noise levels at each history length, improving rollout robustness. Notably, we enforce $t^1 \le t^2 \le \cdots \le t^N$ to mirror the inference-time noise profile, and replace the typical uniform timestep distribution with a logit-normal one [20]. Both are supported by ablations in Tab. 4.

Loss Function. We supervise the predicted $\hat{\boldsymbol{x}}_0$ in Eq. (7) with the following overall objective[3]:

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{a}, \boldsymbol{z}, \boldsymbol{t}} \left[\frac{1}{N} \sum_{n=1}^{N} \left[e^{-u(t^n)} \mathcal{L}_{\text{rot}}^n + \mathsf{sg}\left(e^{-u(t^n)} \right) \left(\mathcal{L}_{\text{joint}}^n + \mathcal{L}_{\text{struct}}^n \right) + u(t^n) \right] + \mathcal{L}_{\text{bd}} \right], \tag{8}$$

$$\mathcal{L}_{\text{bd}} = \frac{1}{N-1} \sum_{n=1}^{N-1} \left[\mathcal{L}_{\text{rot}}\left(\hat{\boldsymbol{b}}_n, \boldsymbol{b}_n \right) + \mathcal{L}_{\text{vel}}\left(\hat{\boldsymbol{b}}_n, \boldsymbol{b}_n \right) + \mathcal{L}_{\text{acc}}\left(\hat{\boldsymbol{b}}_n, \boldsymbol{b}_n \right) \right]. \tag{9}$$

Here, $\mathcal{L}_{\text{rot}}^n$ is the MSE loss on rot6d [97] and translations, while $\mathcal{L}_{\text{joint}}^n$ and $\mathcal{L}_{\text{struct}}^n$ are forward kinematics (FK)-based losses on joint positions and skeletal structures adopted from GlobalDiff [92]. To adaptively mitigate optimization conflicts across blocks at varying noise levels, we train a small MLP $u(\cdot)$ to predict per-timestep weights [33] rather than manual tuning. We apply a stop-gradient (sg) to prevent the fluctuations of FK-based losses from destabilizing $u(\cdot)$. The boundary loss $\mathcal{L}_{\text{bd}}$ regularizes cross-block continuity via velocity and acceleration matching losses (*i.e.*, $\mathcal{L}_{\text{vel}}, \mathcal{L}_{\text{acc}}$) on the transition windows $\hat{\boldsymbol{b}}_n$ and $\boldsymbol{b}_n$ of $\hat{\boldsymbol{x}}_0^n$ and $\boldsymbol{x}_0^n$, and is excluded from the adaptive weighting as each window spans two blocks at different timesteps. See Sec. C for definitions of $\mathcal{L}_{\text{rot}}, \mathcal{L}_{\text{joint}}, \mathcal{L}_{\text{struct}}, \mathcal{L}_{\text{vel}}, \mathcal{L}_{\text{acc}}, \hat{\boldsymbol{b}}_n, \boldsymbol{b}_n$.

[3]With a slight abuse of notation, we use $\mathcal{L}$ to denote both the scalar loss term and the corresponding loss function.

Algorithm 1 Inference with hybrid CFG, where $\boldsymbol{c}$ denotes the joint condition of speech and expression.

Require: Condition $\boldsymbol{c}^{1:N}$; Timesteps $\{t_k\}_{k=1}^{M}$; CFG scales $w_{\text{cond}}(t), w_{\text{hist}}(t)$; seed $(\boldsymbol{x}_{\text{seed}}, \boldsymbol{c}_{\text{seed}})$

1: $\boldsymbol{S} \leftarrow \text{Prefill}(\boldsymbol{S},\ \boldsymbol{x}_{\text{seed}},\ \boldsymbol{c}_{\text{seed}})$ **if** $(\boldsymbol{x}_{\text{seed}}, \boldsymbol{c}_{\text{seed}})$ is not $\emptyset$ **else 0** ▷ initialize GDN state

2: **for** $n = 1, 2, \ldots, N$ **do** ▷ iterate for N blocks

3: $\quad \boldsymbol{x}_{t_1}^n \sim \mathcal{N}(0, \boldsymbol{I})$ ▷ initial noise from Gaussian distribution

4: $\quad$ **for** $k = 1, 2, \ldots, M-1$ **do** ▷ iterate for $M-1$ denoising steps

5: $\quad\quad \hat{\boldsymbol{x}}_0^{n,\emptyset} \leftarrow \mathcal{F}_\theta(\boldsymbol{x}_{t_k}^n, t_k, \boldsymbol{S}_\emptyset, \emptyset)$, $\hat{\boldsymbol{x}}_0^{n,\text{c}} \leftarrow \mathcal{F}_\theta(\boldsymbol{x}_{t_k}^n, t_k, \boldsymbol{S}_\emptyset, \boldsymbol{c}^n)$ ▷ unconditional and condition-only

6: $\quad\quad \hat{\boldsymbol{x}}_0^{n,\text{h}} \leftarrow \mathcal{F}_\theta(\boldsymbol{x}_{t_k}^n, t_k, \boldsymbol{S}, \emptyset)$, $\hat{\boldsymbol{x}}_0^{n} \leftarrow \mathcal{F}_\theta(\boldsymbol{x}_{t_k}^n, t_k, \boldsymbol{S}, \boldsymbol{c}^n)$ ▷ history-only and full

7: $\quad\quad \hat{\boldsymbol{x}}_0^n \leftarrow \hat{\boldsymbol{x}}_0^{n,\emptyset} + w_{\text{cond}}(t)(\hat{\boldsymbol{x}}_0^{n,c} - \hat{\boldsymbol{x}}_0^{n,\emptyset}) + w_{\text{hist}}(t)(\hat{\boldsymbol{x}}_0^{n,h} - \hat{\boldsymbol{x}}_0^{n,\emptyset}) + w_{\text{cond}}(t)w_{\text{hist}}(t)(\hat{\boldsymbol{x}}_0^{n} - \hat{\boldsymbol{x}}_0^{n,c} - \hat{\boldsymbol{x}}_0^{n,h} + \hat{\boldsymbol{x}}_0^{n,\emptyset})$ ▷ hybrid CFG

8: $\quad\quad \boldsymbol{x}_{t_{k+1}}^n \leftarrow \text{SamplingStep}(\boldsymbol{x}_{t_k}^n,\ \hat{\boldsymbol{x}}_0^n,\ t_k,\ t_{k+1})$ ▷ take a sampling step for denoising

9: $\quad \boldsymbol{S} \leftarrow \text{Prefill}(\boldsymbol{S},\ \hat{\boldsymbol{x}}_0^n,\ \boldsymbol{c}^n)$ ▷ write block into $\boldsymbol{S}$

10: **return** $\hat{\boldsymbol{x}}_0^{1:N}$ ▷ concatenate all generated motion segments into the final result

3.4 Inference for Long-Horizon Rollout

Prefill-Decode Rollout with State Caching. For long-horizon inference, PASPA employs a prefill-decode rollout (see Algorithm 1). *Prefill* only encodes the clean segment to update the BGDN state, bypassing the denoising sublayers. Invoked after each block iteration (line 8), it integrates the freshly denoised block into the state. *Decode* (lines 4-7) treats $\boldsymbol{S}$ strictly as a read-only condition. It iteratively denoises the current block (line 7), evaluating $\mathcal{F}_\theta$ under varying conditions for hybrid CFG (detailed below). Throughout the rollout, the fixed-size $\boldsymbol{S}$ is the sole persistent cache, initialized by prefilling seed $\boldsymbol{x}_{\text{seed}}$ and $\boldsymbol{a}_{\text{seed}}$ (or zeros) and reused across all N blocks. Unlike linearly growing token caches [36, 74], $\boldsymbol{S}$ guarantees an $\mathcal{O}(1)$ query complexity per step, regardless of history length.

Hybrid Classifier-Free Guidance. Standard CFG [26] modulates only the condition $\boldsymbol{c}^n$, leaving historical dependence uncontrolled. Motivated by [66], we introduce a learnable null state $\boldsymbol{S}_\emptyset$ for hybrid guidance via four predictions (Algorithm 1, lines 5-6): unconditional $(\boldsymbol{S}_\emptyset, \emptyset)$, condition-only $(\boldsymbol{S}_\emptyset, \boldsymbol{c}^n)$, history-only $(\boldsymbol{S}, \emptyset)$, and full $(\boldsymbol{S}, \boldsymbol{c}^n)$. They are combined via a multilinear composition (Algorithm 1, line 7) that incorporates marginal and second-order joint terms, empirically outperforming simpler alternatives (see Tab. 13). To prevent distortions at noise extremes [37], scales $w_{\text{cond}}(t)$ and $w_{\text{hist}}(t)$ are time-dependent, activating only during mid-noise intervals $[0.1, 0.6]$ (otherwise 1). To train Hybrid CFG, we independently drop $\boldsymbol{c}^n$ with probability 10% and sample $\boldsymbol{S}$ as unchanged, null $\boldsymbol{S}_\emptyset$, or interpolated $\alpha\boldsymbol{S} + (1-\alpha)\boldsymbol{S}_\emptyset$ with probabilities $70\%/10\%/20\%$, where α is uniformly sampled from $[0, 1]$. This interpolation enables continuous inference-time control of history strength.

4 Experiments

Setup. We evaluate on BEAT2 [45] using its official splits, and on SHOW [85] following the evaluation protocol of DiffSHEG [11]. Following [54], BEAT2 comparisons use both the "Scott" subset (**1 Speaker**) and the full test set (**All Speakers**), while ablations use the **1 Speaker** setting. PASPA is optimized via Muon [31, 46] with auxiliary AdamW [35] (training details in Sec. E.1). We report FGD [87] (the primary quality indicator) and Diversity (Div) [38] on both datasets, with BC [39] and face MSE [81] on BEAT2, and Beat Alignment (BA) on SHOW. Here, ↑/↓ indicate that higher/lower values are better. For readability, BEAT2 FGD and MSE are multiplied by 10 and 10^8, respectively; SHOW metrics are reported without rescaling. In all tables, gray rows mark our final setting; **bold** and underlined denote the best and second-best results. See Secs. D to F for details.

4.1 Main Results

Results on BEAT2. Table 1 compares PASPA with prior methods on BEAT2. In the 1 Speaker setting, PASPA achieves the best FGD (the primary quality metric) and MSE, indicating superior generation quality. It also obtains the second-best Div and competitive BC, reflecting favorable diversity and speech-motion alignment. In the All Speakers setting, PASPA achieves the best FGD and MSE, showing its stronger multi-speaker generalization. Compared with the FM-based GlobalDiff [92], PASPA improves all metrics, suggesting the advantage of persistent-state history

Table 1: Comparison with state-of-the-art (SOTA) methods on the BEAT2 dataset.

Method	1 Speaker				All Speakers			
	FGD↓	BC↑	Div↑	MSE↓	FGD↓	BC↑	Div↑	MSE↓
CaMN [44]	6.64	0.677	10.86	–	5.12	0.200	5.58	–
DSG [81]	8.81	0.724	11.49	–	11.74	0.734	11.12	–
EMAGE [45]	5.51	0.772	13.06	7.680	6.92	0.284	6.06	6.908
DiffSHEG [11]	8.99	0.714	11.91	7.665	–	–	–	–
MambaTalk [79]	5.37	0.781	13.05	6.289	–	–	–	–
HoloGest [13]	5.34	**0.795**	**14.15**	–	6.46	**0.803**	**13.53**	–
RAG-GESTURE [54]	8.08	0.734	11.97	–	4.87	0.514	9.94	–
SemTalk [89]	4.28	0.777	12.91	6.153	–	–	–	–
GestureLSM [48]	4.09	0.714	13.24	–	–	–	–	–
GlobalDiff [92]	4.78	0.705	13.13	6.330	2.63	0.404	8.24	4.144
Ours	**3.71**	0.746	13.38	**5.768**	**2.33**	0.645	9.42	**3.793**

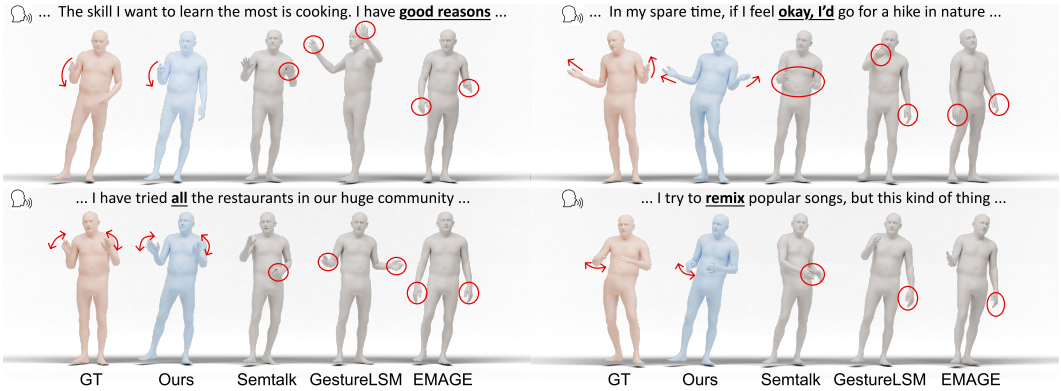

Figure 5: Qualitative comparison. PASPA consistently matches the GT in gesture direction (red arrows), amplitude, and speech alignment. Baselines exhibit various failure modes (red circles): unnatural hand positions, hesitant or mismatched gestures, and missing semantic response.

modeling for FM-based co-speech motion generation. Figure 5 also qualitatively shows that PASPA produces more natural gestures with better semantic alignment to speech than the baselines.

We further conduct a blind ranking study on BEAT2 with 18 participants, yielding 167 valid ranking groups. Participants rank four anonymized methods on 20-second clips across four criteria, with randomized method order and a ground-truth reference. The four criteria are motion realism, semantic consistency with speech, motion-speech synchronization, and motion diversity. PASPA achieves the highest top-1 win rate across all four criteria (see Fig. 6), with 39.8% overall, further supporting these findings. See Secs. E.3 and F.3 for study details and the supplementary material for video demos.

Results on SHOW. To complement our primary evaluation, we evaluate PASPA on SHOW using the same training recipe as on BEAT2 without dataset-specific hyperparameter tuning. As shown in Tab. 2, PASPA achieves the lowest FGD, our primary generation-quality metric, at 0.00111, a 43.9% reduction from our FM-based baseline GlobalDiff, which achieves an FGD of 0.00198. PASPA yields slightly higher BA than GlobalDiff, while its Div remains second best. These results provide complementary evidence of PASPA's effectiveness beyond BEAT2 in co-speech motion generation.

4.2 Ablations and Analysis

As outlined in Sec. 3, PASPA's architecture realizes the past–present asymmetry through inter-block persistent-state modeling and intra-block bidirectional attention. Weakly monotonic timesteps and adaptive weighting stabilize multi-block learning, while Hybrid CFG refines generation quality at inference time and prefill-decode separation accelerates generation. The following ablations and analyses examine these complementary roles and their effects on generation quality and efficiency.

Table 2: Comparison on the SHOW dataset.

Method	FGD↓	BA↑	Div↑
GT	–	0.867	0.834
LS3DCG [24]	0.00478	**0.947**	0.645
TalkSHOW [85]	0.00328	0.872	0.711
DiffSHEG [11]	0.00271	0.902	0.810
GlobalDiff [92]	0.00198	0.870	**1.180**
PASPA	**0.00111**	0.876	1.043

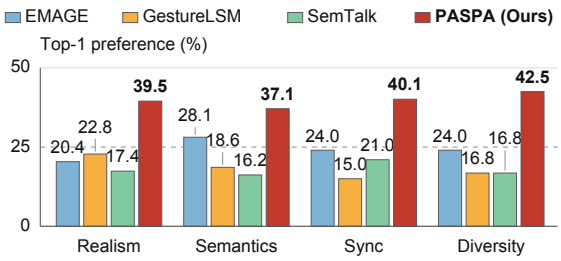

Figure 6: User-study top-1 win rates (%; dashed: 25%).

Table 3: Ablation on past-present modeling. **Past/Curr.**: past/current block modeling. Attn_K (Attn_{all}): attention over K-frame (full) history.

Past	Curr.	FGD↓	BC↑	Div↑	MSE↓
–	Attn	7.065	0.743	13.16	5.727
Attn_8	Attn	6.051	0.721	12.14	5.765
Attn_{all}	Attn	5.710	0.744	11.14	5.696
State	State	7.652	0.785	11.19	6.264
State	Attn	5.216	0.734	14.26	5.761

Table 4: Ablation on training build-up. Each "+" row inherits all components above.

Variant	FGD↓	BC↑	Div↑	MSE↓
Single	7.410	0.744	11.95	6.341
Multi	8.456	**0.747**	12.58	6.090
+ Mono	6.346	0.738	13.78	6.385
+ AdaW	5.614	0.736	**14.93**	**5.736**
+ SG	5.216	0.734	14.26	5.761
+ LogitN	**4.450**	0.739	12.36	5.763

Asymmetric Past-Present Modeling. We validate the PASPA principle by varying the past and current modeling under a fixed training recipe in Tab. 3. Removing history (history branch output set to zero) substantially degrades FGD, confirming the necessity of long-range context. Increasing token-attention history from Attn_8 to Attn_{all} improves FGD, but still underperforms PASPA, showing that retaining more raw history does not guarantee effective history utilization. Moreover, replacing current-block attention with state-based recurrent modeling (*i.e.*, State+State) yields worse FGD than even the no-history variant. This shows that the current block benefits from dense bidirectional interaction and that state-based current-block modeling hurts generation quality. These comparisons validate PASPA's *past-as-state, present-as-attention* division, which yields the best FGD.

Blockwise Training Formulation. We show the build-up ablation of our training recipe in Tab. 4. Naively switching from single-block (*Single*) to multi-block (*Multi*) supervision degrades FGD despite denser supervision, due to unordered block timesteps and imbalanced losses across blocks and noise levels. Weakly monotonic timesteps (Mono) and adaptive loss weighting (AdaW; see Eq. (8)) address these two issues, significantly reducing FGD to 5.614. Beyond the two critical stabilizers, SG (*i.e.*, $\text{sg}(\cdot)$ in Eq. (8)) reduces FGD to 5.216, and LogitN (*i.e.*, logit-normal time sampling [20]) brings it to 4.450. We use the optimal LogitN parameters in Tab. 4; see Tab. 9 for the full sweep.

Classifier-Free Guidance. Table 5 evaluates the CFG build-up, including training dropout and inference guidance. Condition dropout, acting as data augmentation, already reduces FGD from 4.450 to 3.937, and condition guidance further improves it to 3.883. However, history dropout without guidance slightly degrades FGD, emphasizing that it offers no inherent quality gain alone. Further adding history guidance without the second-order joint term also remains worse than condition CFG. Only the full Hybrid CFG substantially reduces FGD to 3.532, validating the explicit interaction between condition and history guidance. The activation interval is fixed to $t \in [0.1, 0.6]$. Detailed CFG scale, composition, and activation-interval ablations are in Tabs. 11a, 11b, 13 and 14. An end-to-end comparison with and without CFG under the main-experiment setting is in Tab. 15.

Long-Rollout Analysis. We quantify long-rollout quality preservation in Tab. 6 (implementation in Sec. E.2). We construct controlled baselines: *Short Seed* uses the last 8 frames, while *Token Attention* attends to all past tokens. They represent finite-window and full-history conditioning, respectively, with other settings matched to PASPA. We use these controlled variants rather than complete methods to isolate history propagation from method-specific differences (*e.g.*, architecture, losses). Although all methods start with reasonable FGD, Short Seed degrades most severely, Token Attention degrades less by retaining explicit history, and PASPA shows the weakest degradation, proving persistent states better preserve long-range tendencies. PASPA also achieves the highest diversity (Div) throughout

Table 5: CFG ablation. Cond (Hist): condition (history) dropout and guidance. $(w_{\text{cond}}, w_{\text{hist}})$ simplified to $(w_{\text{c}}, w_{\text{h}})$.

Dropout	Guidance $(w_{\text{c}}, w_{\text{h}})$		FGD↓	BC↑	Div↑	MSE↓
–	–	(–, –)	4.450	0.739	12.36	**5.763**
Cond	–	(1.0, –)	3.937	0.741	13.46	5.814
Cond	Cond	(1.5, –)	3.883	0.740	13.39	5.823
Cond+Hist	Cond	(1.5, 1.0)	3.989	0.743	13.17	5.782
Cond+Hist	Hybrid†	(1.5, 1.5)	3.967	0.742	**13.76**	5.800
Cond+Hist	Hybrid	(1.5, 1.5)	**3.532**	**0.745**	13.30	5.779

† We remove the second-order joint term in line 7 of Algorithm 1.

Table 6: Long-horizon generation. FGD and Div evaluated on the terminal 10 s window across $T \in \{10, 30, 60\}$ s rollouts.

Method	[0, 10]s		[20, 30]s		[50, 60]s	
	FGD↓	Div↑	FGD↓	Div↑	FGD↓	Div↑
Short Seed	4.731	6.581	5.886	6.033	6.721	6.081
Token Attention	**4.448**	6.289	5.388	6.510	5.771	6.427
PASPA	**4.448**	**7.217**	**5.119**	**7.133**	**5.745**	**7.138**

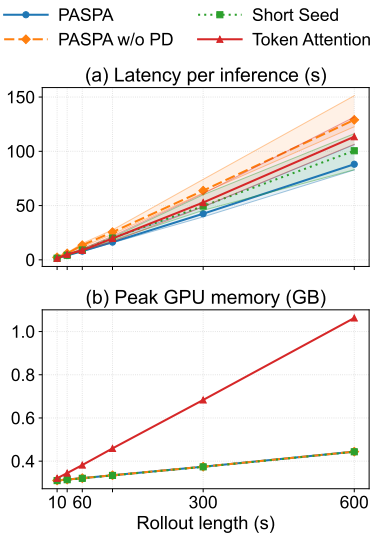

Figure 7: Efficiency analysis. Lines: median latency and peak GPU memory. Shaded bands: min-max latency.

Table 7: Inference overhead of Hybrid CFG under the final PASPA configuration.

Guidance	120 s (3,600 frames)		300 s (9,000 frames)		600 s (18,000 frames)	
	Lat. (ms)	Mem. (GB)	Lat. (ms)	Mem. (GB)	Lat. (ms)	Mem. (GB)
No CFG	11,411	0.36	28,562	0.40	55,978	0.47
Hybrid CFG	12,439	0.37	29,966	0.41	61,093	0.48

rollout. Figure 1 shows consistent qualitative trends, supporting PASPA's long-rollout advantage. See Fig. 9 for complementary long-rollout trends, additional metrics, and uncertainty estimates.

We further test efficiency without CFG on synthetic ultra-long rollouts in Fig. 7, where PASPA w/o PD denotes PASPA without prefill-decode separation. PASPA matches Short Seed's peak memory and stays far below Token Attention's, as history is accumulated into a fixed-size state instead of growing explicit tokens. While latency increases with length for all methods, prefill-decode separation is decisive: it accelerates PASPA from the slowest to the fastest. See Sec. E.2 for details.

Hybrid CFG Inference Overhead. Since Fig. 7 benchmarks all methods without CFG, we measure the additional cost of enabling Hybrid CFG in Tab. 7. Enabling Hybrid CFG increases latency (*Lat.*) by only 4.9–9.1% and peak memory (*Mem.*) by 0.01 GB across the evaluated lengths. We attribute this modest latency increase to batched guidance exploiting spare GPU capacity, as the small motion-generation workload does not saturate the GPU. Together with Tab. 15, these results show that Hybrid CFG improves quality with modest latency overhead and nearly unchanged memory scaling.

5 Conclusion

We present PASPA for long-horizon co-speech motion generation based on the observation that past history and the current segment demand different treatments. PASPA realizes this via a hybrid design, summarizing past history into a persistent state while modeling the current segment with bidirectional attention. Built on blockwise flow matching, PASPA incorporates multi-block supervision for effective training and a prefill-decode protocol with hybrid classifier-free guidance for stable, expressive inference. Extensive public-benchmark evaluations show that PASPA achieves state-of-the-art FGD performance, preserving style and preventing temporal drift during extended motion generation.

Acknowledgments

This work was supported by the Alibaba Research Intern Program.

References

[1] Chaitanya Ahuja, Dong Won Lee, Yukiko I Nakano, and Louis-Philippe Morency. Style transfer for co-speech gesture animation: A multi-speaker conditional-mixture approach. In *European Conference on Computer Vision*, pages 248–265. Springer, 2020.

[2] Chaitanya Ahuja, Dong Won Lee, and Louis-Philippe Morency. Low-resource adaptation for personalized co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20566–20576, 2022.

[3] Chaitanya Ahuja, Pratik Joshi, Ryo Ishii, and Louis-Philippe Morency. Continual learning for personalized co-speech gesture generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 20893–20903, 2023.

[4] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

[5] Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *ACM Transactions on Graphics (TOG)*, 42(4):1–18, 2023.

[6] Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[7] Christian Bischof and Charles Van Loan. The wy representation for products of householder matrices. *SIAM Journal on Scientific and Statistical Computing*, 8(1):s2–s13, 1987.

[8] Yiyi Cai, Yuhan Wu, Kunhang Li, You Zhou, Bo Zheng, and Haiyang Liu. Flooddiffusion: Tailored diffusion forcing for streaming motion generation. *arXiv preprint arXiv:2512.03520*, 2025.

[9] Bohong Chen, Yumeng Li, Yao-Xiang Ding, Tianjia Shao, and Kun Zhou. Enabling synergistic full-body control in prompt-based co-speech motion generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6774–6783, 2024.

[10] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: next-token prediction meets full-sequence diffusion. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 24081–24125, 2024.

[11] Junming Chen, Yunfei Liu, Jianan Wang, Ailing Zeng, Yu Li, and Qifeng Chen. Diffsheg: 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*, pages 7352–7361, 2024.

[12] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[13] Yongkang Cheng and Shaoli Huang. Hologest: Decoupled diffusion and motion priors for generating holisticly expressive co-speech gestures. *arXiv preprint arXiv:2503.13229*, 2025.

[14] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10850–10869, 2023.

[15] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Forty-first International Conference on Machine Learning*, 2024.

[16] Tri Dao and Albert Gu. Transformers are ssms: generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, pages 10041–10071, 2024.

[17] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pages 8780–8794, 2021.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[19] Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pages 569–593. Springer, 1992.

[20] 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.

[21] Ronald A Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521, 1915.

[22] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[23] Han Guo, Songlin Yang, Tarushii Goel, Eric P. Xing, Tri Dao, and Yoon Kim. Log-linear attention. In *The Fourteenth International Conference on Learning Representations*, 2026.

[24] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Lingjie Liu, Hans-Peter Seidel, Gerard Pons-Moll, Mohamed Elgharib, and Christian Theobalt. Learning speech-driven 3d conversational gestures from video. In *Proceedings of the 21st ACM International Conference on Intelligent Virtual Agents*, pages 101–108, 2021.

[25] Bo Han, Hao Peng, Minjing Dong, Yi Ren, Yixuan Shen, and Chang Xu. Amd: Autoregressive motion diffusion. In *38th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI-24)*, pages 2022–2030. AAAI Press, 2024.

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

[27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 6840–6851, 2020.

[28] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[29] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[30] Thierry Joffrain, Tze Meng Low, Enrique S Quintana-Ortí, Robert van de Geijn, and Field G Van Zee. Accumulating householder transformations, revisited. *ACM Transactions on Mathematical Software (TOMS)*, 32(2):169–179, 2006.

[31] Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024.

[32] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 52996–53021, 2024.

[33] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24174–24184, 2024.

[34] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, pages 5156–5165, 2020.

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[36] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

[37] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 122458–122483, 2024.

[38] Jing Li, Di Kang, Wenjie Pei, Xuefei Zhe, Ying Zhang, Zhenyu He, and Linchao Bao. Audio2gestures: Generating diverse gestures from speech audio with conditional variational autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11293–11302, 2021.

[39] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13401–13412, 2021.

[40] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6), 2017.

[41] Yuanzhi Liang, Qianyu Feng, Linchao Zhu, Li Hu, Pan Pan, and Yi Yang. Seeg: Semantic energized co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10473–10482, 2022.

[42] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

[43] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

[44] Haiyang Liu, Zihao Zhu, Naoya Iwamoto, Yichen Peng, Zhengqing Li, You Zhou, Elif Bozkurt, and Bo Zheng. Beat: A large-scale semantic and emotional multi-modal dataset for conversational gestures synthesis. In *European conference on computer vision*, pages 612–630. Springer, 2022.

[45] Haiyang Liu, Zihao Zhu, Giorgio Becherini, Yichen Peng, Mingyang Su, You Zhou, Xuefei Zhe, Naoya Iwamoto, Bo Zheng, and Michael J Black. Emage: Towards unified holistic co-speech gesture generation via expressive masked audio gesture modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1144–1154, 2024.

[46] Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for llm training. *arXiv preprint arXiv:2502.16982*, 2025.

[47] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025.

[48] Pinxin Liu, Luchuan Song, Junhua Huang, Haiyang Liu, and Chenliang Xu. Gesturelsm: Latent shortcut based co-speech gesture generation with spatial-temporal modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10929–10939, 2025.

[49] Xian Liu, Qianyi Wu, Hang Zhou, Yinghao Xu, Rui Qian, Xinyi Lin, Xiaowei Zhou, Wayne Wu, Bo Dai, and Bolei Zhou. Learning hierarchical cross-modal association for co-speech gesture generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10462–10472, 2022.

[50] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[51] Yifei Liu, Qiong Cao, Yandong Wen, Huaiguang Jiang, and Changxing Ding. Towards variable and coordinated holistic co-speech motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1566–1576, 2024.

[52] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), 2015.

[53] David McNeill. *Hand and mind: What gestures reveal about thought*. University of Chicago press, 1992.

[54] M Hamza Mughal, Rishabh Dabral, Merel CJ Scholman, Vera Demberg, and Christian Theobalt. Retrieving semantics from the deep: an rag solution for gesture synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16578–16588, 2025.

[55] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[56] Simbarashe Nyatsanga, Taras Kucherenko, Chaitanya Ahuja, Gustav Eje Henter, and Michael Neff. A comprehensive review of data-driven co-speech gesture generation. *Comput. Graph. Forum*, 42(2):569–596, 2023.

[57] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019.

[58] 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.

[59] Xingqun Qi, Chen Liu, Lincheng Li, Jie Hou, Haoran Xin, and Xin Yu. Emotiongesture: Audio-driven diverse emotional co-speech 3d gesture generation. *IEEE Transactions on Multimedia*, 26:10420–10430, 2024.

[60] Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. Blockwise self-attention for long document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2555–2565, 2020.

[61] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training.

[62] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *International Conference on Machine Learning*, pages 42818–42835. PMLR, 2024.

[63] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pages 9355–9366. PMLR, 2021.

[64] Florian Schmidt. Generalization in generation: A closer look at exposure bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 157–167, 2019.

[65] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[66] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *International Conference on Machine Learning*, pages 56242–56280. PMLR, 2025.

[67] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 11918–11930, 2019.

[68] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[69] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 7537–7547, 2020.

[70] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.

[71] 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.

[72] Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 10–19, 2019.

[73] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6309–6318, 2017.

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[75] Evgeniia Vu, Andrei Boiarov, and Dmitry Vetrov. Streaming generation of co-speech gestures via accelerated rolling diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 26054–26061, 2026.

[76] Bernard Widrow and Marcian E Hoff. Adaptive switching circuits. In *Neurocomputing: foundations of research*, pages 123–134. 1988.

[77] Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. Motionstreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10086–10096, 2025.

[78] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: Efficient High-Resolution Image Synthesis with Linear Diffusion Transformer, 2024.

[79] Zunnan Xu, Yukang Lin, Haonan Han, Sicheng Yang, Ronghui Li, Yachao Zhang, and Xiu Li. Mambatalk: efficient holistic gesture synthesis with selective state space models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 20055–20080, 2024.

[80] Songlin Yang and Yu Zhang. Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism, 2024.

[81] Sicheng Yang, Zhiyong Wu, Minglei Li, Zhensong Zhang, Lei Hao, Weihong Bao, Ming Cheng, and Long Xiao. Diffusestylegesture: Stylized audio-driven co-speech gesture generation with diffusion models. *arXiv e-prints*, pages arXiv–2305, 2023.

[82] Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning*, pages 56501–56523. PMLR, 2024.

[83] Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 115491–115522, 2024.

[84] Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *The Thirteenth International Conference on Learning Representations*, 2025.

[85] Hongwei Yi, Hualin Liang, Yifei Liu, Qiong Cao, Yandong Wen, Timo Bolkart, Dacheng Tao, and Michael J Black. Generating holistic 3d human motion from speech. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 469–480, 2023.

[86] Youngwoo Yoon, Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4303–4309. IEEE, 2019.

[87] Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Speech gesture generation from the trimodal context of text, audio, and speaker identity. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.

[88] Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 12381–12392, 2019.

[89] Xiangyue Zhang, Jianfang Li, Jiaxu Zhang, Ziqiang Dang, Jianqiang Ren, Liefeng Bo, and Zhigang Tu. Semtalk: Holistic co-speech motion generation with frame-level semantic emphasis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13761–13771, 2025.

[90] Xiangyue Zhang, Jianfang Li, Jiaxu Zhang, Jianqiang Ren, Liefeng Bo, and Zhigang Tu. Echomask: Speech-queried attention-based mask modeling for holistic co-speech motion generation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 10827–10836, 2025.

[91] Xiangyue Zhang, Yiyi Cai, Kunhang Li, Kaixing Yang, You Zhou, Zhengqing Li, Xuangeng Chu, Jiaxu Zhang, and Haiyang Liu. Personagesture: Single-reference co-speech gesture personalization for unseen speakers. *arXiv preprint arXiv:2605.06064*, 2026.

[92] Xiangyue Zhang, Jianfang Li, Jianqiang Ren, and Jiaxu Zhang. Mitigating error accumulation in co-speech motion generation via global rotation diffusion and multi-level constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12834–12842, 2026.

[93] Xiangyue Zhang, Jianfang Li, Jiaxu Zhang, Kaixing Yang, and Steven Hoi. Streamtalk: Streaming co-speech gesture generation with key-pose anchoring. *arXiv preprint arXiv:2608.01643*, 2026.

[94] Zeyu Zhang, Akide Liu, Ian Reid, Richard Hartley, Bohan Zhuang, and Hao Tang. Motion mamba: Efficient and long sequence motion generation. In *European Conference on Computer Vision*, pages 265–282. Springer, 2024.

[95] Zeyu Zhang, Shuning Chang, Yuanyu He, Yizeng Han, Jiasheng Tang, Fan Wang, and Bohan Zhuang. Blockvid: Block diffusion for high-quality and consistent minute-long video generation. *arXiv preprint arXiv:2511.22973*, 2025.

[96] Kaifeng Zhao, Gen Li, and Siyu Tang. Dartcontrol: A diffusion-based autoregressive motion model for real-time text-driven motion control. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

[98] Lingting Zhu, Xian Liu, Xuanyu Liu, Rui Qian, Ziwei Liu, and Lequan Yu. Taming diffusion models for audio-driven co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10544–10553, 2023.

[99] Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. Human motion generation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4):2430–2449, 2023.

A Empirical Motivation for the Asymmetry Between Past and Present

Diagnostic Setup. As illustrated in Fig. 3, we employ autocorrelation analysis to investigate whether stylistic features and frame-level local dynamics exhibit distinct temporal persistence in co-speech motion generation. The analysis is conducted on the training subset of BEAT2 [45].

For each sequence, we convert SMPL-X rotations [52, 57] to 3D joint positions by forward kinematics. To factor out orientation and scale variations, we respectively remove pelvic global yaw and normalize seven root-relative upper-body joints (neck, shoulders, elbows, wrists) by shoulder width. We use

$$\boldsymbol{P}_t = \left[\boldsymbol{p}_t^{\text{neck}}, \boldsymbol{p}_t^{\text{left-shoulder}}, \boldsymbol{p}_t^{\text{right-shoulder}}, \boldsymbol{p}_t^{\text{left-elbow}}, \boldsymbol{p}_t^{\text{right-elbow}}, \boldsymbol{p}_t^{\text{left-wrist}}, \boldsymbol{p}_t^{\text{right-wrist}}\right] \in \mathbb{R}^{7\times 3}$$

to denote the normalized upper-body pose at frame t, where each $\boldsymbol{p}_t^{(\cdot)} \in \mathbb{R}^3$ is a root-local 3D joint position. Both the style and local feature families are computed over the same trailing window $\mathcal{W}_t = \{t - W + 1, \ldots, t\}$ with $W = 30$ frames (*i.e.*, 1 s).

The style feature $\boldsymbol{s}_t \in \mathbb{R}^5$ consists of five sub-features, defined as follows:

$$\boldsymbol{s}_t = \left[s_t^{\text{ext}}, s_t^{\text{span}}, s_t^{\text{height}}, s_t^{\text{lat}}, s_t^{\text{occ}}\right], \tag{10}$$

whose five components are computed as follows:

- *Extension* measures the average distance from the elbows and wrists to the neck:

$$s_t^{\text{ext}} = \frac{1}{4W} \sum_{r\in\mathcal{W}_t} \sum_{a\in\mathcal{J}_{\text{ext}}} \left\|\boldsymbol{p}_r^a - \boldsymbol{p}_r^{\text{neck}}\right\|_2, \tag{11}$$

  with $\mathcal{J}_{\text{ext}} = \{\text{left-elbow}, \text{right-elbow}, \text{left-wrist}, \text{right-wrist}\}$.
- *Span* measures the average distance between the two wrists:

$$s_t^{\text{span}} = \frac{1}{W} \sum_{r\in\mathcal{W}_t} \left\|\boldsymbol{p}_r^{\text{left-wrist}} - \boldsymbol{p}_r^{\text{right-wrist}}\right\|_2. \tag{12}$$

- *Height* measures the average wrist height relative to the neck along the vertical axis:

$$s_t^{\text{height}} = \frac{1}{2W} \sum_{r\in\mathcal{W}_t} \left[\left(\boldsymbol{p}_r^{\text{left-wrist}} - \boldsymbol{p}_r^{\text{neck}}\right)_y + \left(\boldsymbol{p}_r^{\text{right-wrist}} - \boldsymbol{p}_r^{\text{neck}}\right)_y\right]. \tag{13}$$

- *Laterality* measures the left-right asymmetry of hand extension:

$$s_t^{\text{lat}} = \frac{1}{W} \sum_{r\in\mathcal{W}_t} \left(\left\|\boldsymbol{p}_r^{\text{right-wrist}} - \boldsymbol{p}_r^{\text{neck}}\right\|_2 - \left\|\boldsymbol{p}_r^{\text{left-wrist}} - \boldsymbol{p}_r^{\text{neck}}\right\|_2\right). \tag{14}$$

- *Occupancy* measures the spatial extent explored by the wrists within the window:

$$s_t^{\text{occ}} = \log\left(\text{tr}\left[\text{Cov}\left(\left\{\boldsymbol{p}_r^{\text{left-wrist}}, \boldsymbol{p}_r^{\text{right-wrist}}\right\}_{r\in\mathcal{W}_t}\right)\right] + \epsilon\right). \tag{15}$$

These components provide an interpretable diagnostic of stylistic speaker characteristics, whose temporal persistence can be compared with local motion dynamics.

The local motion feature $\boldsymbol{l}_t \in \mathbb{R}^{21}$ is defined as a window-level high-pass pose residual:

$$\boldsymbol{l}_t = \text{vec}\left(\boldsymbol{P}_t - \frac{1}{W} \sum_{r\in\mathcal{W}_t} \boldsymbol{P}_r\right) \in \mathbb{R}^{21}. \tag{16}$$

Here, $\text{vec}(\cdot)$ flattens the 7×3 joint-position matrix into a 21D vector. This residual removes the local window mean and emphasizes fast-changing pose variations around the recent motion tendency. By construction, $\boldsymbol{l}_t$ and $\boldsymbol{s}_t$ forms complementary frequency components over the same window $\mathcal{W}_t$, attributing any difference to the data rather than the operator.

For the alternative features shown in the inset of Fig. 3, with the local feature unchanged, we replace the hand-designed style feature $\boldsymbol{s}_t$ with the windowed raw pose mean:

$$\boldsymbol{s}_t^{\text{alt}} = \text{vec}\left(\frac{1}{W} \sum_{r\in\mathcal{W}_t} \boldsymbol{P}_r\right) \in \mathbb{R}^{21}, \tag{17}$$

which is used to examine robustness against the choice of our hand-crafted style descriptors.

Feature Preprocessing. Before computing autocorrelation, we apply two preprocessing steps.

First, to prevent speech energy from inflating the autocorrelation of features, we partial out its linear contribution from each feature dimension. Let $\boldsymbol{y}_t \in \mathbb{R}^D$ denote any of $\boldsymbol{s}_t$, $\boldsymbol{l}_t$, or $\boldsymbol{s}_t^{\text{alt}}$, and let ρ_t denote the frame-aligned log-RMS speech energy (*i.e.*, the natural logarithm of the per-frame root mean square of the waveform). For each frame t, we build a lagged audio-energy vector:

$$\boldsymbol{u}_t^{\text{rms}} = [\rho_{t-L}, \ldots, \rho_t, \ldots, \rho_{t+L}]^\intercal, \qquad L = 15, \tag{18}$$

which covers a temporal range of ± 0.5 seconds at 30 FPS. For each feature dimension d, we fit a ridge regressor [28] from the lagged audio energy to the feature value:

$$\hat{\boldsymbol{\beta}}_d = \arg\min_{\boldsymbol{\beta}} \sum_{t \in \mathcal{I}} \left(\boldsymbol{y}_{t,d} - (\boldsymbol{u}_t^{\text{rms}})^\intercal \boldsymbol{\beta} \right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2, \tag{19}$$

where $\mathcal{I}$ denotes the valid frame indices with complete audio lags and we use $\lambda = 1$. We replace each feature dimension with its residual to obtain features without linear speech-energy contribution:

$$\tilde{\boldsymbol{y}}_{t,d} = \boldsymbol{y}_{t,d} - (\boldsymbol{u}_t^{\text{rms}})^\intercal \hat{\boldsymbol{\beta}}_d. \tag{20}$$

Second, each feature dimension is standardized to zero mean and unit variance using per-speaker statistics, so that no single dimension dominates the autocorrelation. For notational simplicity, we use $\boldsymbol{y}_t$ to denote the preprocessed feature sequence in the following.

Trace Autocorrelation. Given a preprocessed multivariate feature sequence $\boldsymbol{y}_t \in \mathbb{R}^D$, where $\boldsymbol{y}_t$ can be either the style feature $\boldsymbol{s}_t$ or the local feature $\boldsymbol{l}_t$, we measure its temporal persistence using trace autocorrelation (ACF). Let $\bar{\boldsymbol{y}}$ denote the temporal mean. For a temporal lag τ, the trace ACF is:

$$r_{\boldsymbol{y}}(\tau) = \frac{\operatorname{tr}\left(\frac{1}{T-\tau} \sum_{t=1}^{T-\tau} (\boldsymbol{y}_t - \bar{\boldsymbol{y}}) (\boldsymbol{y}_{t+\tau} - \bar{\boldsymbol{y}})^\intercal \right)}{\operatorname{tr}\left(\frac{1}{T} \sum_{t=1}^{T} (\boldsymbol{y}_t - \bar{\boldsymbol{y}}) (\boldsymbol{y}_t - \bar{\boldsymbol{y}})^\intercal \right)}. \tag{21}$$

This yields a variance-weighted ACF over feature dimensions, without selecting a coordinate.

After computing sequence-level ACFs, we aggregate these ACFs $r_{\boldsymbol{y}}(\tau)$ in two stages:

- *Within each speaker*, we average the per-sequence ACFs weighted by $\max(T_i - \tau, 0)$, where T_i is the length of sequence i, so that short sequences contribute less at large lags.
- *Across speakers*, we aggregate via the Fisher-z transform [21]:

$$\bar{r}_{\boldsymbol{y}}(\tau) = \tanh\left(\frac{1}{S} \sum_{s=1}^{S} \operatorname{arctanh}\left(r_{\boldsymbol{y}}^s(\tau) \right) \right), \quad \tau > 0, \tag{22}$$

  where $r_{\boldsymbol{y}}^s(\tau)$ is the within-speaker ACF of speaker s, and S is the number of speakers. Specifically, we set $\bar{r}_{\boldsymbol{y}}(0) = 1$ by definition.

Finally, we estimate 95% confidence intervals by bootstrap resampling [19] over speakers (2000 resamples), which corresponds to the shaded regions shown in Fig. 3. The same procedure is applied to both main and alternative features.

Observations and Implications. Figure 3 reveals a clear temporal-persistence asymmetry between the two feature types. The style ACF $\bar{r}_{\boldsymbol{s}}(\tau)$ remains substantially positive beyond 100 frames. Conversely, the local ACF $\bar{r}_{\boldsymbol{l}}(\tau)$ decays to near zero within a few tens of frames, indicating that frame-level pose dynamics are inherently short-term. The pattern persists when replacing hand-crafted style descriptors with the windowed raw pose mean (inset), confirming the asymmetry is data-intrinsic rather than descriptor-induced. This asymmetry motivates PASPA's core design: past history mainly carries slowly varying stylistic tendencies that should be summarized into a persistent state, whereas the current block contains fast-changing local dynamics that benefit from dense intra-block attention.

B Background: Chunkwise Parallel Form of Gated DeltaNet

We denote the query, key, and value matrices as $\boldsymbol{Q}, \boldsymbol{K} \in \mathbb{R}^{T\times d_k}$ and $\boldsymbol{V} \in \mathbb{R}^{T\times d_v}$, where T is the sequence length and d_k, d_v denote the per-head query/key and value dimensions, respectively.

The chunkwise parallel form partitions $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ into $N = T/K$ non-overlapping chunks of size K (assuming T is divisible by K). The n-th chunk ($n \in \{1, 2, \ldots, N\}$) of $\boldsymbol{Q}$ is written as $\boldsymbol{Q}_{[n]} := \boldsymbol{q}_{(n-1)K+1:nK+1}$ and analogously for $\boldsymbol{K}_{[n]}, \boldsymbol{V}_{[n]}$. We further denote the r-th token within chunk n as $\boldsymbol{q}^r_{[n]} := \boldsymbol{q}_{(n-1)K+r} \in \mathbb{R}^{d_k}$, with $\boldsymbol{k}^r_{[n]}, \boldsymbol{v}^r_{[n]}$ defined analogously. The scalar gating $\alpha^r_{[n]}$ and write-strength $\beta^r_{[n]}$ follow the same within-chunk indexing. Finally, we define the cumulative gating product $\gamma^r_{[n]} := \prod_{j=1}^{r} \alpha^j_{[n]}$ for the first r tokens of chunk n, with $\gamma^0_{[n]} := 1$ by convention.

The chunkwise parallel form computes the state update and output read for an entire chunk via:

$$\boldsymbol{S}_{[n+1]} = \overrightarrow{\boldsymbol{S}_{[n]}} + \Big(\widetilde{\boldsymbol{U}}_{[n]} - \overleftarrow{\boldsymbol{W}_{[n]}}\boldsymbol{S}^{\intercal}_{[n]}\Big)^{\intercal}\overrightarrow{\boldsymbol{K}_{[n]}}, \tag{23}$$

$$\boldsymbol{O}_{[n]} = \underbrace{\overleftarrow{\boldsymbol{Q}_{[n]}}\boldsymbol{S}^{\intercal}_{[n]}}_{\text{inter-chunk term}} + \underbrace{\big(\boldsymbol{Q}_{[n]}\boldsymbol{K}^{\intercal}_{[n]} \odot \boldsymbol{M}\big)\Big(\widetilde{\boldsymbol{U}}_{[n]} - \overleftarrow{\boldsymbol{W}_{[n]}}\boldsymbol{S}^{\intercal}_{[n]}\Big)}_{\text{intra-chunk term}}, \tag{24}$$

where $\boldsymbol{M} \in \mathbb{R}^{K\times K}$ is the intra-chunk causal mask with $\boldsymbol{M}_{ij} = 1$ if $i \geq j$ and 0 otherwise. The output $\boldsymbol{O}_{[n]} \in \mathbb{R}^{K\times d_v}$ decomposes into two terms: an *inter-chunk term* that reads from the block-start state $\boldsymbol{S}_{[n]}$ carrying information from previous chunks, and an *intra-chunk term* capturing token-level interactions within the current chunk. We detailed the core notations of Eqs. (23) and (24) below.

Arrow Notation. The left and right arrows denote cumulative gating coefficients within a chunk:

$$\overleftarrow{\boldsymbol{q}^r_{[n]}} = \gamma^r_{[n]}\boldsymbol{q}^r_{[n]}, \quad \overleftarrow{\boldsymbol{w}^r_{[n]}} = \gamma^r_{[n]}\boldsymbol{w}^r_{[n]}, \quad \overrightarrow{\boldsymbol{k}^r_{[n]}} = \frac{\gamma^K_{[n]}}{\gamma^r_{[n]}}\boldsymbol{k}^r_{[n]}, \quad \overrightarrow{\boldsymbol{S}_{[n]}} = \gamma^K_{[n]}\boldsymbol{S}_{[n]}, \tag{25}$$

with the first three equations extending to stacked matrix forms $\overleftarrow{\boldsymbol{Q}_{[n]}}, \overleftarrow{\boldsymbol{W}_{[n]}}, \overrightarrow{\boldsymbol{K}_{[n]}} \in \mathbb{R}^{K\times d_k}$, whose r-th rows are $\overleftarrow{\boldsymbol{q}^r_{[n]}}, \overleftarrow{\boldsymbol{w}^r_{[n]}}, \overrightarrow{\boldsymbol{k}^r_{[n]}}$, respectively. Intuitively, left arrows rescale each token by the cumulative gating from the start, while right arrows apply the remaining gating toward final position K.

WY coefficients and the UT transform. The matrices $\widetilde{\boldsymbol{U}}_{[n]} \in \mathbb{R}^{K\times d_v}$ and $\boldsymbol{W}_{[n]} \in \mathbb{R}^{K\times d_k}$ are the WY representation [7] coefficients for the cumulative generalized Householder product $\prod_i(\boldsymbol{I} - \beta^i_{[n]}\boldsymbol{k}^i_{[n]}\boldsymbol{k}^{i\intercal}_{[n]})$ that arises when unrolling Eq. (2) across a chunk. The matrices $\widetilde{\boldsymbol{U}}_{[n]}$ and $\boldsymbol{W}_{[n]}$ admit closed-form expressions via the UT transform [30], as follows:

$$\boldsymbol{T}_{[n]} = \Big[\boldsymbol{I} + \text{strictLower}\big(\text{diag}(\beta_{[n]})(\boldsymbol{\Gamma}_{[n]} \odot \boldsymbol{K}_{[n]}\boldsymbol{K}^{\intercal}_{[n]})\big)\Big]^{-1}\text{diag}(\beta_{[n]}), \tag{26}$$

$$\boldsymbol{W}_{[n]} = \boldsymbol{T}_{[n]}\boldsymbol{K}_{[n]}, \qquad \widetilde{\boldsymbol{U}}_{[n]} = \boldsymbol{T}_{[n]}\boldsymbol{V}_{[n]}. \tag{27}$$

Here $\boldsymbol{\Gamma}_{[n]} \in \mathbb{R}^{K\times K}$ is the decay-aware mask with $(\boldsymbol{\Gamma}_{[n]})_{ij} = \frac{\gamma^i_{[n]}}{\gamma^j_{[n]}}$ for $i \geq j$ and 0 otherwise, and strictLower$(\cdot)$ extracts the strictly lower-triangular part of a matrix, zeroing out other parts.

Remarks. Equations (23) and (24)) consist of dense matrix multiplications and element-wise operations, making the chunkwise form well-suited for tensor-core acceleration. Flash Linear Attention (FLA) [80] further provides Triton-accelerated implementations of such chunkwise kernels.

C Detailed Formulations of Loss Functions

In this section, we provide the exact mathematical definitions of the individual loss terms.

C.1 Details of Sequence-Level Losses

The rotation loss $\mathcal{L}_{\text{rot}}^n$ is the per-frame Mean Squared Error (MSE) between predicted and ground-truth motion representations within block n:

$$\mathcal{L}_{\text{rot}}^n = \frac{1}{K}\sum_{i=1}^{K}\left\|\hat{\boldsymbol{x}}_0^{n,i} - \boldsymbol{x}_0^{n,i}\right\|_2^2, \tag{28}$$

where $\hat{\boldsymbol{x}}_0^{n,i}$ and $\boldsymbol{x}_0^{n,i}$ denote the predicted and ground-truth 6D motion representations with the global translation at the i-th frame of block n, respectively.

Following GlobalDiff [92], we apply forward kinematics (FK) to map the rotations [97] into 3D joint positions in the global space. We denote the predicted and ground truth FK joint positions at the i-th frame of block n as $\hat{\boldsymbol{\mathcal{J}}}_0^{n,i}$ and $\boldsymbol{\mathcal{J}}_0^{n,i}$, respectively, and define two FK-based losses on top of them.

The joint loss $\mathcal{L}_{\text{joint}}^n$ directly supervises absolute joint positions:

$$\mathcal{L}_{\text{joint}}^n = \frac{1}{K}\sum_{i=1}^{K}\left\|\hat{\boldsymbol{\mathcal{J}}}_0^{n,i} - \boldsymbol{\mathcal{J}}_0^{n,i}\right\|_2^2. \tag{29}$$

The structural loss $\mathcal{L}_{\text{struct}}^n$ instead supervises bone-level relations, enforcing consistency of pairwise bone-angle similarities to discourage unnatural bone stretching. Let $\mathcal{E}$ denote the set of bones in the kinematic tree, where each bone $e = (p(j), j)$ connects a child joint j to its parent $p(j)$. The normalized predicted bone direction $\hat{\boldsymbol{d}}_e^{n,i}$ and ground truth bone direction $\boldsymbol{d}_e^{n,i}$ are

$$\hat{\boldsymbol{d}}_e^{n,i} = \frac{\hat{\boldsymbol{\mathcal{J}}}_j^{n,i} - \hat{\boldsymbol{\mathcal{J}}}_{p(j)}^{n,i}}{\left\|\hat{\boldsymbol{\mathcal{J}}}_j^{n,i} - \hat{\boldsymbol{\mathcal{J}}}_{p(j)}^{n,i}\right\|_2 + \epsilon}, \qquad \boldsymbol{d}_e^{n,i} = \frac{\boldsymbol{\mathcal{J}}_j^{n,i} - \boldsymbol{\mathcal{J}}_{p(j)}^{n,i}}{\left\|\boldsymbol{\mathcal{J}}_j^{n,i} - \boldsymbol{\mathcal{J}}_{p(j)}^{n,i}\right\|_2 + \epsilon}. \tag{30}$$

The predicted and ground truth pairwise bone-angle similarity matrix $\hat{\boldsymbol{\mathcal{A}}}^{n,i}$ and $\boldsymbol{\mathcal{A}}^{n,i}$ are

$$\hat{\boldsymbol{\mathcal{A}}}_{e,e'}^{n,i} = \left(\hat{\boldsymbol{d}}_e^{n,i}\right)^\top \hat{\boldsymbol{d}}_{e'}^{n,i}, \qquad \boldsymbol{\mathcal{A}}_{e,e'}^{n,i} = \left(\boldsymbol{d}_e^{n,i}\right)^\top \boldsymbol{d}_{e'}^{n,i}. \tag{31}$$

The structural loss is then defined as:

$$\mathcal{L}_{\text{struct}}^n = \frac{1}{K|\mathcal{E}|^2}\sum_{i=1}^{K}\sum_{e,e'\in\mathcal{E}}\rho\Big(\hat{\boldsymbol{\mathcal{A}}}_{e,e'}^{n,i} - \boldsymbol{\mathcal{A}}_{e,e'}^{n,i}\Big), \tag{32}$$

where we use $\rho(\Delta) = |\Delta|$ (the original GlobalDiff paper uses the squared form $\rho(\Delta) = \Delta^2$).

C.2 Details of Boundary Losses

Boundary Windows. The transition window $\hat{\boldsymbol{b}}_n$ at the boundary of the n-th predicted block $\hat{\boldsymbol{x}}_0^n$ concatenates the last W predicted frames of block n with the first W predicted frames of block $n+1$:

$$\hat{\boldsymbol{b}}_n := \big(\hat{\boldsymbol{b}}_n^{(1)}, \ldots, \hat{\boldsymbol{b}}_n^{(2W)}\big), \quad \text{where} \quad \hat{\boldsymbol{b}}_n^{(j)} = \begin{cases} \hat{\boldsymbol{x}}_0^{n,\,K-W+j} & j = 1, \ldots, W, \\ \hat{\boldsymbol{x}}_0^{n+1,\,j-W} & j = W+1, \ldots, 2W. \end{cases} \tag{33}$$

and K is the block size. The ground truth window $\boldsymbol{b}_n$ is constructed analogously from the clean n-th block $\boldsymbol{x}_0^n$. In all of our experiments, we set $W = 4$.

Boundary Loss Formulations. Since adjacent blocks are denoised without temporal overlap during the training process, kinematic discontinuities may appear at the transition points. To address this, we impose the boundary loss on the concatenated $2W$-frame boundary window $(\hat{\boldsymbol{b}}_n, \boldsymbol{b}_n)$ in Eq. (33).

Specifically, the boundary rotation loss $\mathcal{L}_{\text{rot}}(\hat{\boldsymbol{b}}_n, \boldsymbol{b}_n)$ applies the MSE in Eq. (28) to the $2W$ frames of the transition window. To explicitly guarantee temporal smoothness, we compute the first-order

(velocity) and second-order (acceleration) finite differences on the FK-derived joint positions of the boundary window. The velocity matching loss $\mathcal{L}_{\text{vel}}$ and acceleration matching loss $\mathcal{L}_{\text{acc}}$ are:

$$\mathcal{L}_{\text{vel}}(\hat{\boldsymbol{b}}_n, \boldsymbol{b}_n) = \frac{1}{2W-1} \sum_{i=1}^{2W-1} \left\| \left(\hat{\boldsymbol{\mathcal{J}}}^{i+1}_{\boldsymbol{b}_n} - \hat{\boldsymbol{\mathcal{J}}}^{i}_{\boldsymbol{b}_n} \right) - \left(\boldsymbol{\mathcal{J}}^{i+1}_{\boldsymbol{b}_n} - \boldsymbol{\mathcal{J}}^{i}_{\boldsymbol{b}_n} \right) \right\|_2^2, \tag{34}$$

$$\mathcal{L}_{\text{acc}}(\hat{\boldsymbol{b}}_n, \boldsymbol{b}_n) = \frac{1}{2W-2} \sum_{i=1}^{2W-2} \left\| \left(\hat{\boldsymbol{\mathcal{J}}}^{i+2}_{\boldsymbol{b}_n} - 2\hat{\boldsymbol{\mathcal{J}}}^{i+1}_{\boldsymbol{b}_n} + \hat{\boldsymbol{\mathcal{J}}}^{i}_{\boldsymbol{b}_n} \right) - \left(\boldsymbol{\mathcal{J}}^{i+2}_{\boldsymbol{b}_n} - 2\boldsymbol{\mathcal{J}}^{i+1}_{\boldsymbol{b}_n} + \boldsymbol{\mathcal{J}}^{i}_{\boldsymbol{b}_n} \right) \right\|_2^2, \tag{35}$$

where $\hat{\boldsymbol{\mathcal{J}}}^{i}_{\boldsymbol{b}_n}$ and $\boldsymbol{\mathcal{J}}^{i}_{\boldsymbol{b}_n}$ represents the FK joint position at the i-th frame within the boundary slice $\hat{\boldsymbol{b}}_n$ and $\boldsymbol{b}_n$, respectively. These kinematic terms collectively ensure that the generated motion remains physically plausible and temporally continuous across block transitions.

D Dataset Information

BEAT2. BEAT2, proposed by EMAGE [45], is a public mesh-level, motion-captured holistic co-speech motion benchmark derived from the earlier BEAT dataset [44], providing synchronized 3D body and facial motion paired with speech audio. It contains about 60 hours of 3D motion data paired with speech from 25 speakers (12 female, 13 male), comprising 1,762 dialogue sequences that each last around 65 seconds on average. Data representation includes SMPL-X [57] body parameters (55 axis-angle joints, 3D translation, 4D foot-contact) and FLAME [40] facial expressions (100D coefficients). Motion and speech are released at 30 FPS and 16 kHz respectively. Following prior work [45, 54, 89, 92], we adopt the same data partition and report results under two protocols: (1) The **1 Speaker** protocol trains and evaluates on the "Scott" subset, isolating style modeling within a single individual and enabling direct comparison with prior works that report single-speaker performance; (2) The **All Speakers** protocol uses the full speaker set jointly, probing cross-identity generalization.

SHOW. SHOW [85] is used as an additional benchmark to complement our primary evaluation on BEAT2. It contains about 26.9 hours of in-the-wild talking videos from four speakers with diverse speaking styles. Unlike the motion-captured annotations in BEAT2, SHOW provides pseudo-ground-truth motion reconstructed from monocular videos, represented as SMPL-X parameters covering body poses, hand gestures, and facial expressions. Motion is provided at 30 FPS with synchronized speech audio. We follow the gesture-evaluation protocol of DiffSHEG [11] for the SHOW experiments.

E Implementation Details

E.1 Implementation Details of Main Results

Table 8 summarizes training and inference configurations on BEAT2, our primary benchmark. We apply these settings to SHOW for supplementary evaluation of PASPA's effectiveness beyond BEAT2.

Dataset and Motion Representation. We train and evaluate on BEAT2 [45], comprising 25 speakers with synchronized speech, body motion, and expressions. Each motion frame consists of 55 skeleton joints (25 torso, 30 hands), a global root translation, and a 100-dimensional FLAME expression vector [40]. Following common practice, native axis-angle rotations are converted to rot6d [97]. The rot6d rotations [97] and global root translation form a 333-dimensional motion vector. For SHOW [85], we expand the hand PCA coefficients into axis-angle rotations and align the poses with BEAT2's 55-joint ordering, followed by conversion to global rot6d. Translation channels are set to zero, while the 100-dimensional expression coefficients are retained. For both datasets, speech is processed at 16 kHz, yielding 1024-dim WavLM features [12].

Architecture and Hardware. We adopt a 6-layer backbone shown in Fig. 4 with hidden dimension 384 and 8 attention heads. It primarily consists of a PASPA module (Sec. 3.2), with BGDN for history propagation and BMHSA for current-block modeling. We use RMSNorm [88] for normalization and RoPE [68] for positional encoding within BMHSA (see Sec. 3.2.2). The block size is set to $K = 64$ by default, the empirically optimal value in Tab. 10. Speaker IDs are one-hot encoded and mapped to separate learned embeddings in the expression predictor and motion backbone. In the latter, the speaker and timestep embeddings are concatenated into a 768-dimensional conditioning vector.

Table 8: Implementation details of our best training and inference configurations on BEAT2.

Category	Hyperparameter	Setting / Value
Dataset & Setup	Skeleton	55 joints (30 hands, 25 torso)
	Translation	3D global root translation
	Expression	100-dim FLAME blendshapes
	Rotation Representation	axis-angle (raw) $\rightarrow$ rot6d [97]
	Motion Dimension	$55 \times 6 + 3 = 333$
	Audio Features	WavLM [12], 16 kHz, 1024-dim, pre-extracted
	# Speakers	25 identities
Hardware	Devices & Precision	4$\times$ NVIDIA A800 (40 GB) GPUs, Float32
	Parallelism	Data parallel (DDP)
Architecture	Parameters / MACs	65.158 M, 16.633 G
	Hidden Dim / Depth / Heads	384 / 6 / 8
	Time (and ID) Embedding	768 Channels
	History Module	Blockwise Gated DeltaNet
	Self-Attention Module	Blockwise Multi-Head Self-Attention
	Norm / Positional Encoding	RMSNorm [88] / RoPE [68]
	Block Size (K)	64 frames
Training	Training Length	1,000 epochs (316,000 steps)
	Training Time	$\sim$30 hours on 4 GPUs ($\sim$120 GPU-hours)
	Checkpoint Selection	Every 10 epochs; best-FGD checkpoint reported
	Batch Size	72 per GPU (global: 288)
	Training Sequence	256 frames (*i.e.*, 4 contiguous blocks)
	CFG Dropout	cond 0.1; hist: keep 0.7 / null 0.1 / interp. 0.2
	Gradient Clipping	max-norm $= 1.0$
	Adaptive Loss Weighting	✔ (*Enabled*)
	Time Sampling	Weakly Monotonic Logit-normal [20]
	Logit-Normal	mean $\mu = -1.0$, std $\sigma = 1.0$
Optimization	Optimizer / EMA	Muon [31, 46] + AdamW [35] / ✗ (*Disabled*)
	Learning Rate	$10^{-3} \rightarrow 10^{-4}$ at step 5,000
	Muon Settings	momentum $= 0.95$, RMS-adjusted LR
	AdamW Settings	$\beta_1 = 0.9$, $\beta_2 = 0.999$
	Weight Decay	0.01
Inference	Sampling Steps / Seed Length	10 / 8
	CFG Mode / CFG Scales	Hybrid (*Interaction*) / $w_{\text{cond}} = 1.5, w_{\text{hist}} = 1.5$
	Activation Interval	$t \in [0.1, 0.6]$

For both datasets, a speaker-conditioned network predicts 100-dimensional expression coefficients from WavLM features. On BEAT2, we reuse GlobalDiff's pretrained audio-to-expression network [92]. On SHOW, we train a TalkSHOW-based face predictor [85] using WavLM features to predict the expression. WavLM and expression predictors remain frozen during PASPA training.

The BEAT2 configuration has 65.158 M parameters and 16.633 G multiply-accumulate operations (MACs) per forward pass. The MACs are measured via `ptflops` at batch size 1 and sequence length 256, with all Triton kernels replaced by their PyTorch fallbacks. Training is distributed via data parallelism on 4 NVIDIA A800 GPUs using Float32 precision.

Training. We train for 1,000 epochs on both datasets (316,000 steps on BEAT2 and 82,000 on SHOW). On BEAT2, main experiments and 500-epoch ablations and analyses take approximately 30 and 15 hours, respectively, on 4 GPUs. Following the standard reporting protocol used by prior work [89, 92], we save checkpoints every 10 epochs and report results from the checkpoint which achieves the best FGD on each dataset. Each GPU processes 72 sequences of 256 frames (*i.e.*, 4 blocks), yielding a global batch size of 288. As detailed in Sec. 3.3, we sample timesteps from a logit-normal distribution [20] ($\mu = -1.0$, $\sigma = 1.0$) and enforce a weakly monotonic order across

blocks, with both designs empirically validated in Tabs. 4 and 9. The training loss uses adaptive loss weighting (see Eq. (8)) to balance contributions across timesteps and blocks, which is proven crucial in Tab. 4. The condition (`cond`) is dropped with 0.1 probability, while the history state (`hist`) is assigned three modes: kept (`keep`, 0.7), nulled (`null`, 0.1), or linearly interpolated to null (`interp`, 0.2). To avoid extreme gradient updates, we rescale gradients to unit L2 norm.

Optimization. We optimize PASPA via a mixed Muon-AdamW optimizer [31, 35, 46] for fast convergence, omitting weight EMA (exponential moving average). Following standard practice, Muon [31, 46] optimizes the hidden matrix-shaped weights, while AdamW [35] handles the remaining parameters, including the input embeddings, the final output projection, and the 1D gains and biases. We further place BGDN under AdamW, as we find Muon ineffective on it. Muon uses momentum 0.95 with RMS-matched learning-rate adjustment to keep the Muon update scale comparable to AdamW. AdamW uses $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Both optimizers apply weight decay 0.01. The shared learning rate is 10^{-3}, dropping to 10^{-4} after $5{,}000$ steps to accelerate early training.

Inference. We use 10 denoising steps. Following common practice [45, 89, 92], we prefill an 8-frame seed motion (see line 1 of Algorithm 1) into the state to initialize the rollout. We adopt Hybrid CFG under the *Interaction* mode (*i.e.*, keeping second-order joint term in line 7 of Algorithm 1; concrete form in Tab. 12). We use guidance scales $w_{\text{cond}} = 1.5$ and $w_{\text{hist}} = 1.5$ for the condition and the history, validated in Tab. 11. Guidance is restricted to $t \in [0.1, 0.6]$, tuned via Tab. 14.

E.2 Implementation Details of Other Experiments

Long-rollout endpoint analysis. We conduct the long-rollout style preservation analysis in Tab. 6. This analysis requires sequences up to 60 s, and restricting the evaluation to a single speaker leaves too few such clips for stable statistics. We therefore evaluate on a subset of the full test set comprising all sequences exceeding 60 s. For each rollout horizon $T \in \{10, 30, 60\}$ s, metrics are computed on the terminal 10 s window, corresponding to $[0, 10]$ s, $[20, 30]$ s, and $[50, 60]$ s respectively. We use two baselines: *Short Seed* (corresponding to Short-Seed Truncation in Fig. 1, which keeps only the last 8 frames as history) and *Token Attention* (corresponding to Long-Context Token Attention in Fig. 1, which retains all past frames as tokens for attention). Figure 9 further reports rollout trends with additional metrics, using 95% confidence intervals from $1{,}000$ clip-level bootstrap resamples.

Inference Efficiency Benchmark. We evaluate scalability on a single NVIDIA 4090 GPU under Float32 precision in Fig. 7, using the same configuration in Tab. 8, with CFG disabled for all methods. *PASPA* refers to our default setting (using prefill-decode rollout and state caching), while *PASPA w/o PD* disables prefill-decode separation and re-prefills the full history at every denoising step. We also include two baselines from Fig. 1: *Token Attention* and *Short Seed*. We sweep rollout length over $\{10, 30, 60, 120, 300, 600\}$ s (300 to $18{,}000$ frames at 30 fps). For each configuration, after 10 warm-ups, we record 50 timed runs, measuring per-call latency via `time.perf_counter` (with `torch.cuda.synchronize`) and peak memory via `torch.cuda.max_memory_allocated`. For the CFG-overhead analysis in Tab. 7, we benchmark PASPA with and without Hybrid CFG on one NVIDIA H100 using an input batch size of 1, following the same timing and memory protocol.

E.3 Implementation Details of the User Study

We conduct a blind ranking study through a custom web interface (see Fig. 8) to assess perceptual quality. A total of 18 participants take part in the study, yielding 167 valid ranking groups after filtering incomplete annotations. For each clip, participants are shown the ground-truth motion together with four anonymized generated motions produced by SemTalk [89], GestureLSM [48], EMAGE [45], and our PASPA, with method order randomized per clip. Participants rank all four methods from best (1) to worst (4) along four criteria: (1) realism, (2) semantic consistency with speech, (3) motion-speech synchronization, and (4) motion diversity, with unique ranks enforced per criterion. We report the mean rank in Sec. F.3 (lower is better), where PASPA achieves the best rank across all four criteria.

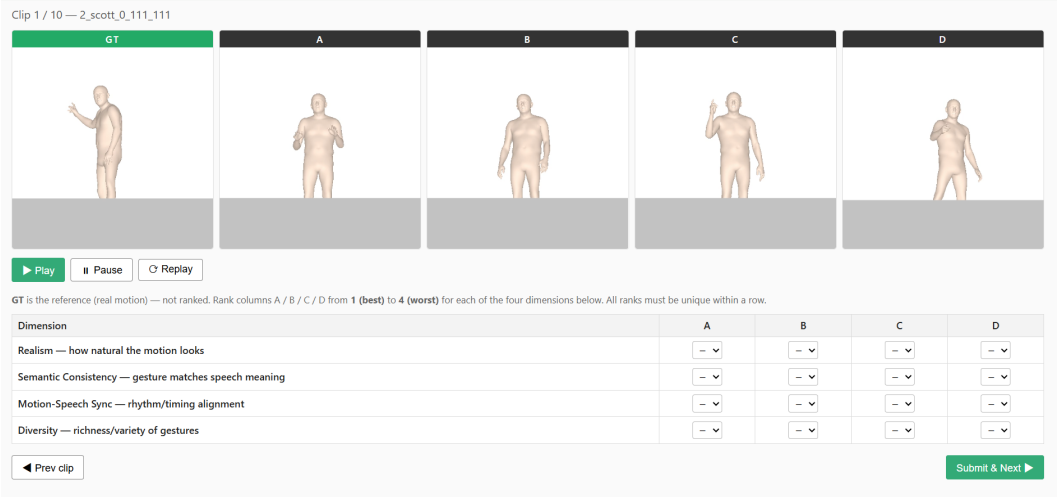

Figure 8: User study interface. Participants rank four anonymized methods (A/B/C/D, randomized per clip) produced by SemTalk, GestureLSM, EMAGE and PASPA, against a non-rankable ground-truth reference (GT), along four criteria: realism, semantic consistency, motion-speech sync, and diversity.

Table 9: Ablation on training time distribution, including uniform and logit-normal sampling.

Time Sampling	**FGD** ↓	**BC** ↑	**Div** ↑	**MSE** ↓
Uniform	5.216	0.734	14.26	5.761
Logit-Normal sweep on mean μ with std $\sigma = 1.0$				
$\mu = 0.0$	4.952	0.736	12.56	5.913
$\mu = -0.2$	5.552	**0.742**	13.00	5.881
$\mu = -0.4$	5.320	0.738	13.74	5.843
$\mu = -0.6$	4.908	0.734	13.28	5.773
$\mu = -0.8$	4.548	0.730	12.24	5.795
$\mu = -1.0$	**4.450**	0.739	12.36	5.763
$\mu = -1.2$	5.128	0.738	**14.35**	**5.660**

Table 10: Ablation on block size (K) and training window length (T) of our blockwise FM.

Variant	**FGD** ↓	**BC** ↑	**Div** ↑	**MSE** ↓
Block size sweep with fixed $T = 256$				
$K = 16$	5.030	**0.744**	13.58	9.158
$K = 32$	5.188	0.730	**13.89**	13.941
$K = 64$	**4.450**	0.739	12.36	**5.763**
Training window sweep with fixed $K = 64$				
$T = 128$	4.955	0.732	**14.70**	**5.660**
$T = 256$	**4.450**	**0.739**	12.36	5.763
$T = 512$	5.811	0.731	14.42	5.807
$T = 768$	4.706	0.735	14.20	5.768

F Additional Experimental Results

F.1 Additional Ablation Studies

F.1.1 Time Distribution

Table 9 provides the parameter study supporting our LogitN choice (*i.e.*, logit-normal time sampling [20]) in the main ablation (see Tab. 4). All rows share the same training recipe (*i.e.*, Mono + AdaW + SG in Tab. 4) and differ only in the time sampling distribution mentioned in Sec. 3.3. The Uniform row corresponds to the +SG variant in Tab. 4, while the LogitN rows fix the standard deviation $\sigma = 1.0$ and vary the mean μ. Choosing a moderately negative LogitN mean improves FGD over uniform sampling, with $\mu = -1.0$ achieving the best FGD. However, an overly negative mean (*e.g.*, $\mu = -1.2$) worsens FGD, suggesting that time sampling should not be biased too aggressively. Although other μ values are favored by individual secondary metrics (*e.g.*, BC, Div and MSE), we use FGD as the primary selection criterion and therefore adopt $\mu = -1.0$.

Table 11: CFG scale ablations for Condition CFG and Hybrid CFG. Cond CFG denotes condition CFG. Both panels use the same activation interval $t \in [0.1, 0.6]$.

(a) Condition CFG

w_{cond}	FGD↓	BC↑	Div↑	MSE↓
–	4.450	0.739	12.36	**5.763**
Condition CFG guidance				
0.5	4.151	0.739	13.28	5.811
1.0	3.937	0.741	13.46	5.814
1.5	**3.883**	0.740	13.39	5.823
2.0	4.060	0.741	13.19	5.803
2.5	4.235	0.742	13.14	5.825
3.0	4.080	0.748	13.27	5.811
3.5	4.208	0.746	13.50	5.827
4.5	4.640	**0.753**	**13.66**	5.810

(b) Hybrid CFG ablation

$(w_{\text{cond}}, w_{\text{hist}})$	FGD↓	BC↑	Div↑	MSE↓
$(1.5, -)$	3.883	0.740	13.39	5.823
Hist scale sweep ($w_{\text{cond}} = 1.5$)				
$(1.5, 0.5)$	4.163	0.741	13.17	5.788
$(1.5, 1.0)$	3.989	0.743	13.17	5.782
$(\mathbf{1.5}, \mathbf{1.5})$	**3.532**	0.745	13.30	**5.779**
$(1.5, 2.0)$	4.156	0.744	**13.77**	5.797
$(1.5, 2.5)$	5.603	**0.751**	13.76	5.779
Cond scale sweep ($w_{\text{hist}} = 1.5$)				
$(0.5, 1.5)$	3.884	0.739	13.28	5.803
$(1.0, 1.5)$	3.611	0.740	13.20	5.781
$(2.0, 1.5)$	3.657	0.745	13.57	5.784
$(2.5, 1.5)$	3.769	0.748	13.74	5.790

F.1.2 Block Size and Training Sequence Length

Table 10 studies the sensitivity to the block size K and the training window length T. For the block-size sweep, we fix $T = 256$ and vary K, while for the window-length sweep, we fix $K = 64$ and vary T. In the block-size sweep, $K = 64$ achieves the best FGD, whereas smaller blocks degrade FGD and substantially increase MSE. This suggests that overly short blocks fragment the local motion segment and limit the dense bidirectional interactions needed for modeling fast-changing intra-block dynamics. In the window-length sweep, $T = 256$ gives the best FGD. A shorter window ($T = 128$) remains competitive on some secondary metrics, but its worse FGD indicates that limited histories hurt the primary motion quality. Conversely, increasing the window length (*e.g.*, $T = 512$) does not further improve FGD, which we attribute to the larger number of blocks in each training window, introducing more inter-block dependencies and making multi-block denoising harder to learn. We therefore use $K = 64$ and $T = 256$ as the default setting.

F.1.3 Classifier-Free Guidance

Conditional Classifier-Free Guidance. Table 11a ablates Condition CFG. The first row is trained without CFG-aware dropout and evaluated without inference-time guidance, while $w_{\text{cond}} = 1.0$ is trained with CFG-aware dropout but evaluated without inference-time guidance. This comparison shows that condition dropout alone substantially improves FGD from 4.450 to 3.937. We attribute this gain to condition dropout acting as a training-time data augmentation, which improves robustness to imperfect or weakened conditioning. Increasing the scale to $w_{\text{cond}} = 1.5$ further improves FGD to 3.883, showing that moderate inference-time guidance provides an additional gain. However, larger scales degrade FGD (motion quality), despite improving BC and Div, suggesting over-guidance.

Hybrid Classifier-Free Guidance. Table 11b ablates Hybrid CFG with the two CFG scales w_{cond} and w_{hist}. The first row $(1.5, -)$ is the Condition CFG optimum from Tab. 11a, which uses the checkpoint trained with condition (*i.e.*, speech and expression) dropout only and serves as the baseline. All subsequent rows use the checkpoint trained with both condition and history dropout mentioned in Sec. 3.4. We first sweep the history scale with $w_{\text{cond}} = 1.5$ fixed. Adding history dropout in training alone, without history guidance (*i.e.*, $(1.5, 1.0)$), slightly degrades FGD from 3.883 to 3.989, indicating that the history dropout does not directly improve quality. However, applying history guidance unlocks a substantial improvement, with $(1.5, 1.5)$ achieving the best FGD (*i.e.*, 3.532). Over-guidance is more severe along the history axis than in Condition CFG: $w_{\text{hist}} = 2.5$ substantially degrades FGD to 5.603. We then sweep w_{cond} with $w_{\text{hist}} = 1.5$ fixed:

Table 12: Formulations of Hybrid CFG decomposition modes. $\hat{\boldsymbol{x}}_0$, $\hat{\boldsymbol{x}}_0^c$, $\hat{\boldsymbol{x}}_0^h$, and $\hat{\boldsymbol{x}}_0^{c,h}$ denote predicted clean motions under neither condition nor history, condition only, history only, and both condition and history, respectively.

Mode	Guided prediction
State First	$\hat{\boldsymbol{x}}_0 + w_{\text{hist}}(\hat{\boldsymbol{x}}_0^h - \hat{\boldsymbol{x}}_0) + w_{\text{cond}}(\hat{\boldsymbol{x}}_0^{c,h} - \hat{\boldsymbol{x}}_0^h)$
Cond First	$\hat{\boldsymbol{x}}_0 + w_{\text{cond}}(\hat{\boldsymbol{x}}_0^c - \hat{\boldsymbol{x}}_0) + w_{\text{hist}}(\hat{\boldsymbol{x}}_0^{c,h} - \hat{\boldsymbol{x}}_0^c)$
Independent	$\hat{\boldsymbol{x}}_0 + w_{\text{cond}}(\hat{\boldsymbol{x}}_0^c - \hat{\boldsymbol{x}}_0) + w_{\text{hist}}(\hat{\boldsymbol{x}}_0^h - \hat{\boldsymbol{x}}_0)$
Interaction	$\hat{\boldsymbol{x}}_0 + w_{\text{cond}}(\hat{\boldsymbol{x}}_0^c - \hat{\boldsymbol{x}}_0) + w_{\text{hist}}(\hat{\boldsymbol{x}}_0^h - \hat{\boldsymbol{x}}_0) + w_{\text{cond}}w_{\text{hist}}(\hat{\boldsymbol{x}}_0^{c,h} - \hat{\boldsymbol{x}}_0^c - \hat{\boldsymbol{x}}_0^h + \hat{\boldsymbol{x}}_0)$

Table 13: Ablation on Hybrid CFG decomposition. All variants use Hybrid CFG with $(w_{\text{cond}}, w_{\text{hist}}) = (1.5, 1.5)$ and the activation interval $t \in [0.1, 0.6]$.

Mode	FGD↓	BC↑	Div↑	MSE↓
Independent	3.967	0.742	**13.76**	5.800
Cond First	3.704	0.743	13.35	5.793
State First	3.724	**0.747**	13.30	5.788
Interaction	**3.532**	0.745	13.30	**5.779**

Table 14: Ablation on the CFG activation interval. All variants use Hybrid CFG with *Interaction* composition, setting $(w_{\text{cond}}, w_{\text{hist}}) = (1.5, 1.5)$ inside the interval and $(1, 1)$ outside.

Interval	FGD↓	BC↑	Div↑	MSE↓
$[0.0, 0.6]$	4.515	**0.751**	13.59	5.782
$[\mathbf{0.1}, \mathbf{0.6}]$	**3.532**	0.745	13.30	5.779
$[0.1, 0.7]$	3.706	0.746	13.60	5.786
$[0.1, 1.0]$	4.089	0.743	**13.75**	**5.773**

$(1.5, 1.5)$ remains the FGD optimum, while other values of w_{cond} yield slightly higher FGD but can improve secondary metrics (*e.g.*, Div). We adopt $(w_{\text{cond}}, w_{\text{hist}}) = (1.5, 1.5)$ as the default.

Table 12 summarizes four ways to compose the condition- and history-guidance directions in Hybrid CFG, and Tab. 13 reports their performance. Table 12 adopts a simplified notation: $\hat{\boldsymbol{x}}_0$, $\hat{\boldsymbol{x}}_0^c$, $\hat{\boldsymbol{x}}_0^h$, and $\hat{\boldsymbol{x}}_0^{c,h}$ denote the unconditional, condition-only, history-only, and full predictions, respectively. *State First* and *Cond First* apply the two guidance directions sequentially in opposite orders: *State First* first moves from $\hat{\boldsymbol{x}}_0$ to $\hat{\boldsymbol{x}}_0^h$ and then toward $\hat{\boldsymbol{x}}_0^{c,h}$, while *Cond First* reverses the order. *Independent* treats both guidance directions as parallel marginal contributions relative to $\hat{\boldsymbol{x}}_0$, while *Interaction* additionally adds a second-order term capturing their joint effect. As shown in Tab. 13, *Interaction* achieves the best FGD, supporting our choice in Algorithm 1. We attribute this to the explicit modeling of the joint effect between condition and history beyond their marginal contributions.

Classifier-Free Guidance Activation Interval. Following the observation that classifier-free guidance is mainly beneficial in a mid-noise interval [37], we restrict CFG to this range. Concretely, this is implemented by setting the timestep-dependent scales (*e.g.*, $w_{\text{cond}}(t)$) to non-trivial values only within the chosen interval, and to 1 otherwise (*i.e.*, no guidance). Table 14 validates this design by varying the activation interval. Among the tested cases, $[0.1, 0.6]$ achieves the best FGD. Extending the interval toward either the high-noise side (*e.g.*, $[0.1, 0.7]$) or the low-noise side (*e.g.*, $[0.0, 0.6]$) degrades FGD. We therefore use $[0.1, 0.6]$ as the default activation interval.

Effect of CFG on Main Results. Table 15 compares PASPA with and without CFG under the $1{,}000$-epoch main-experiment configuration, complementing the 500-epoch ablation in Tab. 5. Compared with Hybrid CFG, PASPA without CFG yields worse FGD scores: 3.76 vs. 3.71 in the 1 Speaker setting and 2.59 vs. 2.33 in the All Speakers setting. Hybrid CFG improves FGD, BC, and MSE in both settings, with slightly lower Div. Since FGD is our primary quality metric, we use Hybrid CFG.

F.2 Additional Analysis of Our Method

Additional Long-Rollout Curves. Figure 9 extends the long-rollout results in Tab. 6. It plots endpoint FGD/Div across rollout horizons and additionally reports BC and MSE. Consistent with Tab. 6, PASPA shows the weakest FGD degradation and maintains the highest Div throughout the

Table 15: Effect of inference-time CFG on the main BEAT2 results.

Guidance	1 Speaker				All Speakers			
	FGD↓	BC↑	Div↑	MSE↓	FGD↓	BC↑	Div↑	MSE↓
w/o CFG	3.76	0.744	**13.80**	5.788	2.59	0.639	**9.48**	3.802
Hybrid CFG	**3.71**	**0.746**	13.38	**5.768**	**2.33**	**0.645**	9.42	**3.793**

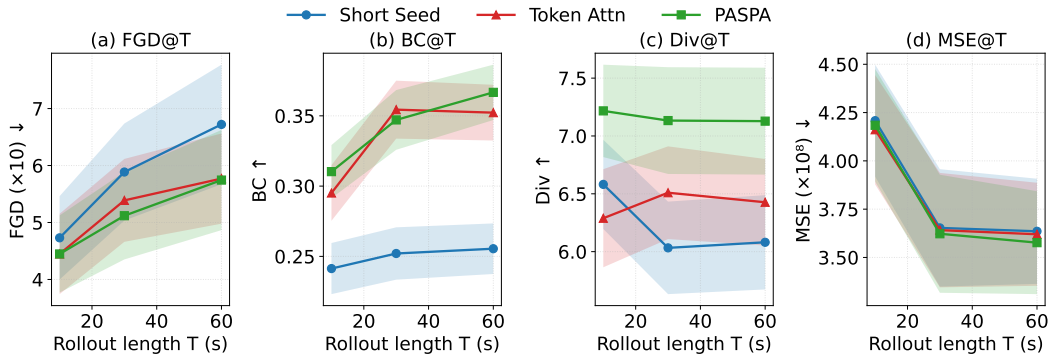

Figure 9: Long-rollout endpoint metrics across $T \in \{10, 30, 60\}$ s, evaluated on the terminal 10 s. Shaded regions denote 95% Wald confidence intervals estimated from clip-level bootstrap resampling.

rollout, supporting its advantage in long-horizon generation. PASPA also achieves competitive BC and slightly lower MSE at later horizons, further supporting its advantage in long-horizon generation.

Generation-Seed Robustness. Table 16 summarizes the 50-seed All Speakers evaluation under the final configuration. Notably, the main-table result in Tab. 1 comes from the original evaluation, whereas the seed sweep re-evaluates the same checkpoint with freshly initialized generation seeds. PASPA achieves a mean FGD of 2.237 with a standard deviation of 0.074, indicating limited sensitivity to generation randomness under this fixed-checkpoint setting. This consistency supports the reliability of the main FGD result beyond a single favorable sampling run.

F.3 User Study

We conduct a blind ranking study to assess perceptual quality beyond automatic metrics (setup details in Sec. E.3). Table 17 reports the mean rank, where PASPA achieves the best on all four criteria, with an overall rank of 2.14 clearly ahead of the second-best EMAGE (2.51). These results are consistent with the top-1 win rates reported in the main paper (see Fig. 6). The user study, together with the metrics in Tab. 1 and qualitative comparisons in Fig. 5, as well as the video demos in the supplementary material, jointly supports PASPA's advantage over current SOTA methods.

G Limitations, Future Work, Failure Cases and Broader Impacts

G.1 Limitations and Future Work

While our method shows promising results in co-speech motion generation, it still has limitations.

Generalization to Other Motion Domains. The proposed *past-as-state, present-as-attention* principle is derived from observations on co-speech motion generation, and we have not verified whether the same asymmetry holds in other long-horizon motion domains such as music-to-dance [39] or text-to-motion [71]. Even if this asymmetry generalizes, the specific PASPA instantiation may require adaptation, and its effectiveness in these broader settings remains to be validated. We leave the empirical verification of PASPA on other motion generation domains to future work.

Table 16: Generation-seed variability in the All Speakers setting. Only the generation seed varies across 50 runs; CIs are 95% Student-t intervals for the mean.

Metric	Main result	50 generation seeds		
		Mean	Sample std.	95% CI of mean
FGD ↓	2.33	2.237	0.074	[2.216, 2.258]
BC ↑	0.645	0.6407	0.0015	[0.6402, 0.6411]
Div ↑	9.42	9.273	0.079	[9.251, 9.296]
MSE ↓	3.793	3.761	0.003	[3.760, 3.762]

Table 17: User study results. Each cell reports the mean rank $\pm$ standard deviation across all participants and clips (lower is better; 1 = best, 4 = worst).

Method	Realism	Semantic	Sync	Diversity	Overall
EMAGE	2.63 ± 1.09	2.41 ± 1.13	2.49 ± 1.08	2.50 ± 1.09	2.51 ± 1.10
GestureLSM	2.60 ± 1.15	2.80 ± 1.12	2.82 ± 1.09	2.83 ± 1.12	2.76 ± 1.12
SemTalk	2.63 ± 1.04	2.59 ± 1.01	2.58 ± 1.08	2.57 ± 0.99	2.59 ± 1.03
PASPA	$\mathbf{2.14 \pm 1.12}$	$\mathbf{2.20 \pm 1.12}$	$\mathbf{2.11 \pm 1.11}$	$\mathbf{2.10 \pm 1.14}$	$\mathbf{2.14 \pm 1.12}$

Bounded State Capacity. PASPA propagates past history through a persistent state of fixed dimensionality $\boldsymbol{S} \in \mathbb{R}^{d_v \times d_k}$, regardless of how much history precedes the current block. This bounded memory imposes an intrinsic compression limit: for extremely long sequences, older context may be gradually overwritten by new updates. The state is designed to retain slowly varying style statistics (*e.g.*, gesture amplitude and spatial preference), but may lose details of distant gestures or long-range correspondences needed to reproduce earlier motions. This trade-off is fundamental to the broader family of linear recurrent models [16, 22, 82, 84]. A promising direction is the recently proposed log-linear attention [23], which replaces the fixed-size state with a logarithmically growing set of states and directly applies to Gated DeltaNet. We leave this for future work.

Train-Inference Gap. Like most autoregressive generation methods, PASPA is trained with teacher forcing method (each block is generated conditioned on ground-truth history) [55, 64], while inference must rely on the previously generated blocks, leading to exposure bias. This bias may lead to error accumulation in long rollouts, potentially manifesting as gesture repetition, posture collapse, identity drift, or global-translation artifacts. We do not explore recent techniques that explicitly bridge this gap, such as Self Forcing [29], which performs autoregressive rollout during training to align the training and inference distributions. Integrating such strategies is a natural direction for future work.

Evaluation Limitations. Our quantitative evaluation uses automatic metrics, yet no universally accepted metric fully captures long-term stylistic consistency and perceptual quality. We therefore complement these metrics with a user study to assess PASPA's effectiveness. Developing new style or perceptual metrics is beyond our scope and is left for future work. Moreover, the generalizability of these findings should be further validated across broader real-world deployment scenarios.

G.2 Failure Cases

Figure 10 shows a representative failure mode of our method. In the ground truth, the speaker performs a large in-place circling motion, leading to strong and highly non-linear changes in global translation. Although PASPA generates plausible local gestures, it fails to recover this circling trajectory and produces motion that remains mostly around a fixed spatial region. We attribute this limitation to the current motion generation pipeline [92], where global translation is learned together with joint rotations (*i.e.*, rot6d [97]) in the same motion stream. These two components have different geometric structures: translation lies in the Euclidean space $\mathbb{R}^3$, whereas the 6D rotation representation (*i.e.*, rot6d [97]) parameterizes rotations through constrained orthonormal frames on the Stiefel manifold, corresponding to the structure of $\mathrm{SO}(3)$. Learning them jointly, especially within

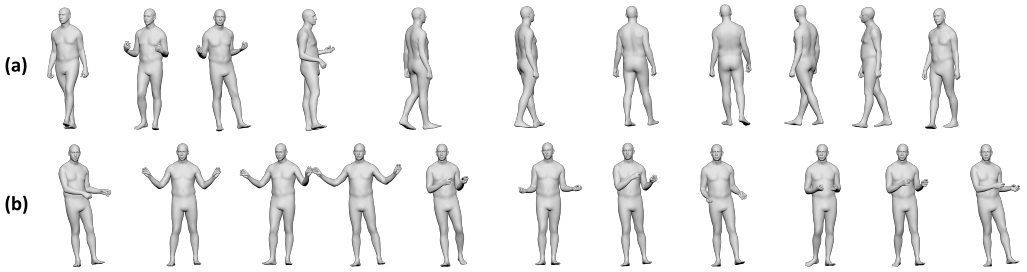

Figure 10: Failure case on sequence `16_jorge_1_3_3`. Time progresses from left to right. (a) Ground-truth. (b) Our generated sequence.

the torso branch, may underemphasize the distinct dynamics of large-amplitude global translation. Following GlobalDiff's co-speech motion generation architecture [92], we keep it for consistency, but more accurate modeling of global translation remains an important direction for future work.

G.3 Broader Impacts

Positive Impacts. PASPA aims to improve long-horizon co-speech motion generation, which can benefit digital humans [99], virtual assistants [87], animated characters [24], education, and entertainment by enabling more coherent and expressive nonverbal behavior. By reducing the manual effort required for full-body animation, it may also lower the production cost of educational content, virtual avatars, and interactive media.

Negative Impacts. Like other generative models for human motion and appearance [45, 71, 85], PASPA could be misused to create misleading speech-driven avatars, contributing to impersonation, disinformation, or non-consensual digital likenesses. Improved long-horizon coherence and style preservation may make synthetic embodied behavior harder to distinguish from real recordings. Although our method generates motion rather than audio or appearance, this risk remains relevant when motion is combined with separately synthesized voice, face, or identity generation systems.

Mitigation. Our experiments are conducted in a research setting using public co-speech motion datasets adopted in prior work [45], and our contribution focuses on improving motion generation within this established benchmark setting, rather than identity recognition, surveillance, or high-stakes decision making. We do not collect new identifiable personal data or deploy the model in user-facing applications. When releasing code or checkpoints, we will use research-oriented license terms and encourage users to avoid deceptive, non-consensual, or identity-misrepresenting applications.

H Licenses of Used Assets

Datasets. We summarize the datasets used in this work and their licenses below.

- **BEAT2** [45]: This dataset is released under the Apache 2.0 license.
- **SHOW** [85]: Provided under the project's Data & Software Copyright License for non-commercial scientific research purposes.

Code and Models. We build upon the following public codebases and pretrained models.

- **SANA** [78]: Released under the Apache 2.0 license. We adopt its codebase as the backbone for building our flow-matching framework.
- **GlobalDiff** [92]: Released under the Apache 2.0 license. We use its pretrained audio-to-expression weights for BEAT2, as well as parts of its data loading pipeline, global-rotation forward-kinematics logic, loss functions, and network architecture.
- **Flash Linear Attention (FLA)** [80]: Released under the MIT license. We adopt its Triton-based Gated DeltaNet (GDN) [84] kernel and PyTorch-based GDN layer implementation, and extend the Triton kernel to support our block-causal variant.

- **Self-Forcing** [29]: Released under the Apache 2.0 license. We adapt its autoregressive-diffusion inference pipeline to our setting.
- **SMPL-X** [57]: Released under the SMPL-X non-commercial scientific research license. We use the SMPL-X body model and its Blender add-on for constructing SMPL-X-compatible digital human representations.
- **Emage Evaltools** [45]: Released under the Apache 2.0 license. We use it for computing evaluation metrics.
- **TalkSHOW** [85]: We adapt its face-prediction implementation to WavLM features and train the predictor on SHOW. No repository-level license file is provided; the project website lists non-commercial research terms.
- **DiffSHEG** [11]: Its code is released under the BSD 3-Clause license. We use its public evaluation components and pretrained feature extractors for SHOW evaluation.
- **WavLM** [12]: Its official implementation is released under the MIT license. We use a frozen WavLM-Large encoder to extract speech features for BEAT2 and SHOW.

All assets are properly cited in the main text and used in compliance with their respective licenses for academic research.

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: The abstract and introduction in Sec. 1 clearly state our scope and contributions for long-horizon co-speech motion generation: (1) identifying the past-present asymmetry and formulating the *past-as-state, present-as-attention* principle; (2) proposing the **PASPA** module with inter-block state modeling and intra-block attention; (3) developing a blockwise flow matching framework with tailored training and inference. These claims are developed in Sec. 3, supported by main results in Sec. 4.1, and validated by ablations in Sec. 4.2.

   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 discuss limitations in Sec. G, including the unverified generalization of the past-as-state, present-as-attention principle to other motion domains, the bounded capacity of the fixed-size state, and the teacher-forcing train-inference gap that may cause exposure bias in long rollouts. We also provide future works and failure case analysis in Secs. G.1 and G.2, respectively.

   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: [N/A]

   Justification: The paper does not present formal theoretical results such as theorems or lemmas requiring proof. The mathematical expressions are used to define the method and analysis, including the FM path in Eq. (1), the BGDN update/read rules in Eqs. (2) and (3), the BMHSA block mask in Eq. (5), the blockwise FM formulation in Eqs. (6) and (7), the training objectives in Eqs. (8) and (9).

   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: We describe the core method in sufficient detail for reproduction: the architecture is specified in Fig. 4 and Sec. 3.2.3; the blockwise FM formulation is given in Sec. 3.3; the loss objectives are defined in Eqs. (8) and (9) with full details in Sec. C; and the long-horizon sampling procedure is specified by Algorithm 1 and Sec. 3.4. We also report the experimental setup, including metrics, in Sec. 4, the datasets in Sec. D, and the full training/inference configuration in Sec. E. Table 8 summarizes the BEAT2 configurations, with SHOW-specific adaptations described in Sec. E.1.

   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: We will release the code with instructions for environment setup, data preparation, training, inference, and evaluation upon publication. The experiments use BEAT2 and SHOW, obtained through their respective official releases and access procedures. Dataset descriptions and representation conversions are provided in Secs. D and E.1.

   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: The core paper specifies the main experimental setting in Sec. 4, including the train/test splits, evaluation subsets, optimizers and evaluation metrics. Implementation details for both datasets, including shared settings and dataset-specific adaptations, are provided in Sec. E. Table 8 summarizes the BEAT2 configurations. Upon code release, we will provide the configuration files and scripts used for the reported experiments, including YAML and shell-script based training and evaluation settings.

   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 report uncertainty or variability for key analyses. Figure 3 uses 95% bootstrap confidence intervals for autocorrelation estimates, Fig. 9 uses 95% confidence intervals from clip-level bootstrap resampling for long-rollout metrics, and Fig. 7 reports median values with min-max latency bands over repeated runs. For the user study, Tab. 17 reports mean rank with standard deviation across valid ranking groups. Table 16 reports means, standard deviations, and 95% Student-t confidence intervals across 50 generation seeds. Details are provided in the corresponding captions and Secs. E.2 and E.3.

   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 report the compute resources in Tab. 8 and Sec. E. The BEAT2 model is trained on 4 NVIDIA A800 GPUs (40GB) using Float32 precision for 1000 epochs, taking approximately 30 hours (*i.e.*, 120 GPU-hours). The SHOW motion model is trained for 1000 epochs (82,000 steps), using the same batch-size and precision settings. We also report the model size, MACs, batch size, and other optimization settings in Sec. E. Models for the controlled ablations are trained for 500 epochs using the same hardware and configuration. The generation-seed evaluation in Tab. 16 reuses the final model without additional training. For the inference-efficiency benchmark, we specify the single-GPU setup and experimental protocol in Sec. E.2.

   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 reviewed the NeurIPS Code of Ethics and believe that the research conforms to it. Our experiments use public co-speech motion benchmarks under their intended research setting, and the work focuses on motion generation methodology rather than identity recognition, surveillance, or high-stakes decision making.

   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: [Yes]

    Justification: We discuss broader impacts in Sec. G.3, including positive applications in domains such as digital humans, virtual assistants, and animated characters, as well as reduced animation production cost. We also discuss potential negative impacts such as misleading speech-driven avatars, impersonation, disinformation, and non-consensual digital likenesses, and clarify that our experiments are conducted in a research setting using public datasets without collecting new identifiable personal data or deploying user-facing systems.

    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 does not release a new scraped dataset or a high-risk generative model such as a language, image, voice, or identity generator. Our method focuses on motion generation for research purposes, and does not generate language, image, video, audio, photorealistic appearance, or identity information. Broader misuse considerations are discussed in Sec. G.3.

    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: [Yes]

    Justification: We use publicly available datasets and codebases, and properly cite their original papers in the main text. The complete list of used assets, including their sources and licenses (*e.g.*, Apache 2.0, MIT, BSD 3-Clause, and the SMPL-X non-commercial research license), is summarized in Sec. H, and all assets are used in compliance with their respective licenses for academic research.

    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 release the PASPA code, trained model checkpoints, and generated motion results, with accompanying documentation covering environment setup, data preparation, training, inference, and evaluation, as well as licensing information and limitations.

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: [Yes]

Justification: The paper includes a user study for perceptual evaluation in Sec. F.3. We provide full participant instructions in Sec. F.3 and a screenshot of the evaluation interface in Fig. 8. Participants were volunteers recruited within our institution, without monetary compensation. The user study follows the NeurIPS Code of Ethics.

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: [Yes]

Justification: The user study involves only voluntary ranking of short generated motion clips, with no personal data collected and no risk to participants beyond ordinary computer use. Participants were volunteers who were informed of the study purpose and voluntary nature of participation. Under our institution's review process, this type of low-risk perceptual study with informed consent does not require formal IRB approval. The study follows the NeurIPS Code of Ethics.

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: The core method development and experiments in this paper do not use LLMs as important, original, or non-standard components. Any use of LLMs, if any, was limited to writing, editing, or formatting assistance and did not affect the methodology, experiments, scientific claims, or originality of the work.

    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.