1. Supplementary Material

A. Overview

The supplementary material is organized as follows:

- Section B: Implementation Details of the Streamable Vector-Quantized Motion Tokenizer (SVQ) and the Hierarchical Autoregressive Transformer (HAR)
- Section C: Evaluation Metrics
- Section D: SOTA Comparison: Quantitative Results Without Face Module
- Section E: SOTA Comparison: User Study
- Section F: Impact of Codebook Size in Streamable Vector-Quantized Motion Tokenizer
- Section G: Causal Attention Design in xAR and xAR-Fuse
- Section H: Role of Classifier-Free Guidance in *LiveGesture*
- Section I: Role of Compositional vs. Full-Body SVQ Tokenization
- Section J: Ablation on Role of Region Masking in Fusion Training
- Section K: Impact of Noise Injection in Local Region-eXperts
- Section L: Ablation on Attention Depth in xAR and xAR-Fuse

B. Implementation Details

LiveGesture is implemented in PyTorch and trained in two major phases: (i) the per-region *Streamable Vector-Quantized Motion Tokenizer* (SVQ), and (ii) the *Hierarchical Autoregressive Transformer* (HAR).

B.1. Streamable Vector-Quantized Motion Tokenizer

Asymmetric Autoencoder Pretraining.. We train one *Streamable Autoencoder* (SA) per SMPL-X body region, $\mathcal{R} = \{\text{upper body}, \text{lower body}, \text{hands}, \text{face}\}$. The input parameterization for each region is as follows: 13 upper-body joints (78-D Rot6D), 30 hand joints (180-D Rot6D), 100-D FLAME expression parameters plus 3-D jaw rotation for the face, and 9 lower-body joints (54-D Rot6D) augmented with global translation (3-D) and four binary foot-contact indicators. Global translation and contact flags are included only in the lower-body region, as leg motion and foot contacts provide strong supervision for root displacement and help reduce foot-sliding artifacts.

For training, each SA processes motion in fixed sliding windows of length $T_w = 16$ frames, denoted $\{\boldsymbol{\theta}_t^{\text{region}}\}_{t=1}^{T_w}$ for a given region. The bidirectional encoder E begins with a temporal convolutional stem that lifts framewise SMPL-X inputs into a higher-dimensional feature space. It then applies two downsampling stages, each consisting of a stride-2 temporal convolution (halving the frame rate) followed by a `ResNet1D` block composed of dilated temporal residual layers. These residual layers jointly capture local kinematic patterns and longer-range bidirectional dependencies within the window. After two stages, the encoder outputs a latent sequence

$$z^{\text{region}} = \{z_\tau\}_{\tau=1}^{T_w/4},$$

i.e., a compact motion representation at one-quarter of the original frame rate. This asymmetric design allows E to exploit both past and future frames to build a globally coherent latent space, while the decoder remains strictly causal.

The Causal Stream Decoder D_{CS} mirrors the encoder hierarchy but enforces strict causality: all convolutions use left-only padding, so the reconstruction at frame t depends only on latent tokens $\{z_{1:\tau}^{\text{region}}\}$ up to the corresponding downsampled index. Each upsampling stage performs nearest-neighbor temporal upsampling by 2, followed by a causal `ResNet1D` block that refines the feature sequence using only past context. After two stages, the decoder reconstructs the full-resolution regional motion window

$$\hat{\boldsymbol{\theta}}^{\text{region}} = D_{\text{CS}}(z^{\text{region}}).$$

Because D_{CS} is strictly causal and operates at a fixed downsampling factor, the same architecture can be run convolutionally over arbitrarily long sequences at inference time without violating the zero–look-ahead constraint. In the final streaming system, only the causal decoder D_{CS} is used at inference; the bidirectional encoder E is employed solely during offline training to shape the latent space.

Each SA is trained independently with per-region normalization and an L1 reconstruction loss

$$\mathcal{L}_{\text{AE}} = \lambda_{\text{AE}} \left\| \hat{\boldsymbol{\theta}}^{\text{region}} - \boldsymbol{\theta}^{\text{region}} \right\|_1, \quad \lambda_{\text{AE}} = 1.$$

This stage focuses solely on learning a temporally coherent, streamable latent space and does not involve any vector quantization.

Quantization Learning.. In Stage 2, we convert the continuous latents z^{region} into discrete, time-synchronous SVQ motion tokens while preserving the temporal geometry and causal decoding behavior learned in Stage 1. To this end, we freeze both the bidirectional encoder E and the causal decoder D_{CS} and learn a region-specific vector-quantized tokenizer on top of the SA latents.

For each region $r \in \mathcal{R}$, we introduce a codebook

$$C^{\text{region}} = \{c_k\}_{k=1}^{K} \in \mathbb{R}^{K\times 128},$$

with $K = 2048$ entries, updated using EMA with decay 0.99. The encoder latents z_τ^{region} are first linearly projected into the 128-D code space. Each projected latent is then assigned to its nearest codebook vector c_k, producing a discrete token index and the corresponding dequantized embedding sequence

$$\hat{z}^{\text{region}} = \{\hat{z}_\tau\}_{\tau=1}^{T_w/4}.$$

To remain compatible with the frozen decoder latent space, each region employs a lightweight projection head W^{region}, implemented as a small MLP. This head maps dequantized latents back into the latent space expected by D_{CS}:

$$\tilde{z}^{\text{region}} = \{\tilde{z}_\tau\}_{\tau=1}^{T_w/4}, \quad \tilde{z}_\tau = W^{\text{region}}(\hat{z}_\tau).$$

The projected sequence is then decoded causally to reconstruct the regional motion window,

$$\hat{\boldsymbol{\theta}}^{\text{region}} = D_{\text{CS}}(\tilde{z}^{\text{region}}).$$

In this stage, only the codebook C^{region} and projection head W^{region} are trainable; E and D_{CS} remain fixed. The codebook is maintained with EMA and we periodically reset rarely used entries to avoid codebook collapse and encourage effective usage of the discrete space. The Stage 2 objective combines an L1 reconstruction term on SMPL-X region parameters with a standard VQ codebook loss:

$$\mathcal{L}_{\text{stage2}} = \lambda_{\text{rec}} \left\| \hat{\boldsymbol{\theta}}^{\text{region}} - \boldsymbol{\theta}^{\text{region}} \right\|_1 + \lambda_{\text{cb}}\, \mathcal{L}_{\text{cb}}(z^{\text{region}}, e^{\text{region}}),$$

where e^{region} denotes the selected codebook embeddings, $\lambda_{\text{rec}} = 1$, and $\lambda_{\text{cb}} = 0.2$. Gradients pass through the quantizer via a straight-through estimator. This two-stage asymmetric design ensures that (i) the latent space and causal decoder remain stable and well-conditioned for streaming, and (ii) the SVQ tokens are compact, region-specific, and time-synchronous at one-quarter of the original motion frame rate, making them ideal discrete inputs for the downstream region-eXpert autoregressive transformers in HAR.

B.2. Hierarchical Autoregressive Transformer (HAR)

Region-eXpert Autoregressive Transformer (xAR).. Each *region-eXpert* xAR module performs causal autoregressive modeling on the SVQ motion tokens produced at one-quarter of the original motion frame rate. For every region $r \in \mathcal{R} = \{\text{upper body}, \text{lower body}, \text{hands}, \text{face}\}$, we maintain an independent stream of SVQ tokens together with aligned audio tokens from the streamable audio encoder and optional text tokens. The model operates with a maximum history of 32 past motion tokens, each mapped to a 128-D codebook embedding ($K = 2048$ entries) and projected to a 256-D representation through a small MLP; rotary positional embeddings are added for stable temporal alignment during streaming. Each region-eXpert is implemented as a lightweight causal Transformer with $L_{\text{xAR}} = 2$ blocks, where every block contains three *causal audio–motion cross-attention* layers that attend only to past and current audio/text tokens, followed by three *causal temporal self-attention* layers applied over the region's token history; all attention layers use strict lower-triangular masks to ensure zero–look-ahead. The final hidden state at time t is passed through a region-specific linear classifier to produce logits over the 2048-entry codebook vocabulary. To improve robustness, we inject Gaussian noise with standard deviation $\sigma_{\text{noise}} = 0.1$ into the embedded motion history with probability $p_{\text{noise}} = 0.2$, and apply classifier-free dropout to audio/text inputs with probability $p_{\text{cf}} = 0.1$, enabling classifier-free guidance at inference with scale $\gamma = 1.25$. All four region-eXperts share the same streamable audio encoder but maintain independent Transformer and classifier weights, and are trained using Adam (learning rate 1×10^{-4}, batch size 128) under the standard autoregressive objective.

Causal Spatial–Temporal Fusion (xAR-Fuse)..

xAR-Fuse enforces whole-body coordination by operating on top of the frozen hidden states produced by the region-eXpert xAR modules. At each time step t, we gather the region-wise features $\{h_t^r\}_{r\in\mathcal{R}}$ and align them using lightweight per-region PILOR adapters, each implemented as a single linear layer with a residual connection. These adapters add only a small number of parameters per region while reliably mapping independently learned region features into a shared fusion space. The aligned features are then processed by a causal fusion Transformer with $L_{\text{fuse}} = 3$ blocks. Each fusion block contains three components: (i) *inter-region spatial attention* across regions at the current time step, (ii) *causal global temporal attention* with key–value caching for long-horizon streaming, and (iii) *causal audio–motion cross-attention* that conditions fused region tokens on past and current audio/text input. The resulting fused representations are fed into the same region-specific token classifiers used in the local xAR stage, now predicting SVQ tokens from joint multi-region context.

Training uses the hybrid masking strategy described in the main method: uncertainty-guided token masking (UGM) corrupts a subset of the lowest-confidence tokens according to a cosine schedule $\lambda_{\text{UGR}}(s)$, where the effective masking ratio increases from 0 to a maximum of 0.5 over the course of training; for each batch, a masking ratio is sampled uniformly in $[0, \lambda_{\text{UGR}}(s)]$ and applied to the selected tokens. Random region masking (RM) is implemented in an analogous way: we gradually increase a region-drop probability p_{drop} from 0 to 0.5, and for each batch sample a value of p_{drop} within the current range and use it as the probability of dropping an entire region's motion-token sequence, encouraging cross-region reasoning under missing modalities. We also apply classifier-free dropout with probability $p_{\text{cf}} = 0.1$, identical to xAR, to retain compatibility with inference-time classifier-free guidance. During xAR-Fuse training, the SVQ tokenizer, causal audio encoder, and all xAR eXperts are frozen; only the PILOR adapters, fusion Transformer blocks, and token classifiers are updated. We train xAR-Fuse using Adam (learning rate 1×10^{-4}, batch size 128). Because all attention operations are strictly causal, the resulting model runs directly in real-time streaming mode without any architectural changes. In the overall training objective, we weight the local xAR loss and the fusion loss with coefficients $\lambda_{\text{local}} = 0.3$ and $\lambda_{\text{fuse}} = 1.0$, respectively.

B.3. Application: Interactive Human–Avatar Conversation

For deployment, *LiveGesture* connects directly to a streaming speech system such as VITA-Audio. Audio is emitted in small chunks (200 ms hop) and immediately passed to the causal audio encoder, which updates the audio tokens at the SVQ rate. HAR advances synchronously, generating one SVQ token per region at each audio step. The SVQ decoder reconstructs full SMPL-X poses in real time, enabling fully synchronized speech–gesture behavior. Because the entire HAR stack is strictly causal, the same implementation supports real-time human–avatar interaction without look-ahead or buffering.

C. Evaluation Metrics

We evaluate *LiveGesture* using four standard metrics that measure realism, variability, speech–motion synchrony, and facial accuracy, following the official BEAT2 protocol.

Fréchet Gesture Distance (FGD).. FGD [7] measures the distributional similarity between real and generated full-body motion in the feature space of a pretrained gesture encoder. Let feature sets extracted from real and generated gestures be $G = \{\mathbf{g}_i\}$ and $\hat{G} = \{\hat{\mathbf{g}}_i\}$, with means $\boldsymbol{\mu}_G,\ \boldsymbol{\mu}_{\hat{G}}$ and covariances $\Sigma_G,\ \Sigma_{\hat{G}}$. FGD is defined as

$$\text{FGD}(G, \hat{G}) = \|\boldsymbol{\mu}_G - \boldsymbol{\mu}_{\hat{G}}\|_2^2 + \text{Tr}\Big(\Sigma_G + \Sigma_{\hat{G}} - 2(\Sigma_G\Sigma_{\hat{G}})^{1/2}\Big)\,. \tag{1}$$

Lower FGD indicates that generated gestures follow natural human-motion statistics, which is critical for strictly causal, zero–look-ahead generation.

L1 Diversity.. L1 Diversity [1] measures variability across multiple gesture realizations produced for the same audio. Given N generated sequences with corresponding joint positions $\{\mathbf{p}_t^{(i)}\}_{t=1}^T$ for $i = 1, \ldots, N$, the diversity score is

$$\text{Div.} = \frac{1}{2N(N-1)T} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{t=1}^{T} \big\|\mathbf{p}_t^{(i)} - \mathbf{p}_t^{(j)}\big\|_1. \tag{2}$$

Global translation of the SMPL-X body is removed prior to evaluation. Higher values indicate more expressive and varied motion under causal token-by-token prediction.

Beat Constancy (BC).. Beat Constancy [2] evaluates the synchrony between motion beats and prosodic beats in the audio. Motion beats are detected from local minima of upper-body joint velocity, while audio beats correspond to peaks in prosodic intensity. Let g_{mot} and a_{aud} denote the sets of detected motion and audio beat times. BC is computed as

$$\text{BC} = \frac{1}{|g_{\text{mot}}|} \sum_{b_g \in g_{\text{mot}}} \exp\left(-\frac{\min_{b_a \in a_{\text{aud}}} \|b_g - b_a\|^2}{2\sigma^2}\right), \tag{3}$$

where σ is a temporal tolerance parameter. Higher BC indicates stronger speech–gesture alignment. Because *LiveGesture* is fully causal with zero–look-ahead, BC directly reflects its ability to track prosody in real time.

Facial Vertex MSE.. For SMPL-X facial motion accuracy, we compute the mean squared error between ground-truth and predicted mesh vertices following [6]. Let $\mathbf{V} = \{\mathbf{v}_i\}$ and $\hat{\mathbf{V}} = \{\hat{\mathbf{v}}_i\}$ be the sets of ground-truth and predicted facial vertices. The metric is

$$\text{MSE}_{\text{face}} = \frac{1}{|\mathbf{V}|} \sum_{i=1}^{|\mathbf{V}|} \|\mathbf{v}_i - \hat{\mathbf{v}}_i\|_2^2. \tag{4}$$

This complements body-motion metrics by evaluating fine-grained facial deformation fidelity.

D. SOTA Comparison: Quantitative Results Without Face Module

Table 1. Comparison with state-of-the-art methods on BEAT2 without the facial motion module. *LiveGesture* remains the only zero–look-ahead streaming model while achieving competitive or superior performance in BC and Diversity.

Methods	Venue	Streaming	FGD (↓)	BC (→)	Diversity (↑)
Offline Solutions					
GestureLSM [5]	ICCV'25	✗	**4.088**	0.714	<u>13.24</u>
Streaming Solution (Ours)					
LiveGesture (ours)	Ours	✓	<u>4.51</u>	**0.783**	**13.31**

Table 1 reports quantitative results when evaluating only full-body motion without the facial module. GestureLSM, an offline model with full future context, achieves the lowest FGD by leveraging non-causal temporal information. In contrast, *LiveGesture* operates under strict zero–look-ahead constraints, yet obtains the best BC and highest Diversity, indicating superior rhythm alignment and richer motion variability. This behavior reflects the strength of *LiveGesture*'s causal audio conditioning and hierarchical xAR–xAR-Fuse architecture, which jointly enable expressive, beat-synchronous gestures even without access to future frames. Although the absence of facial cues slightly increases the distributional distance (FGD) for our model, the strong improvements in synchrony and diversity demonstrate that *LiveGesture* maintains high perceptual quality while remaining the only fully streamable solution.

E. SOTA Comparison: User Study

Protocol. We recruit fifteen participants and present them with a mixed set of gesture clips generated by CaMN [3], EMAGE [4], GestureLSM [5], and *LiveGesture*. Each clip depicts a speaking subject with full-body SMPL-X motion driven by BEAT2 test utterances. All videos are anonymized and randomly ordered so that participants cannot identify which method produced which sequence, reducing bias toward any specific model.

For every clip, participants provide ratings on a five-point Likert scale (1 = lowest, 5 = highest) along three criteria. *Realness* evaluates the naturalness and plausibility of the produced motion. *Speech–gesture synchrony* measures how well the gestures align with the rhythm and prosodic structure of the audio. *Smoothness* assesses temporal continuity and penalizes jitter, discontinuities, or incoherent regional coordination. We report Mean Opinion Scores (MOS) averaged across all participants and clips for each method.

Results. Table 2 summarizes the MOS results. Both *LiveGesture* and GestureLSM are clearly preferred over CaMN and EMAGE across all criteria, indicating that recent models produce noticeably more convincing co-speech motion. GestureLSM, an offline method with full-sequence access, achieves slightly higher Realness (4.2 vs. 4.1) and Smoothness (4.3

Table 2. Mean Opinion Scores (MOS, 1–5, higher is better) from the user study on BEAT2 test clips. The *Streaming* column indicates whether the method supports zero–look-ahead streaming (✓) or is offline-only (✗). *LiveGesture* is the only strictly streaming model and is preferred on speech–gesture synchrony while remaining competitive in realness and smoothness.

Method	Streaming	Realness ↑	Synchrony ↑	Smoothness ↑
CaMN [3]	✗	3.3	3.2	3.6
EMAGE [4]	✗	3.5	3.5	3.7
GestureLSM [5]	✗	**4.2**	4.1	**4.3**
LiveGesture (ours)	✓	4.1	**4.3**	3.9

vs. 3.9) than *LiveGesture*, reflecting its advantage in long-range temporal refinement when future frames are available. In contrast, *LiveGesture* attains the best speech–gesture synchrony score (4.3), surpassing all offline baselines and aligning with its superior BC metric in the quantitative evaluation. This pattern suggests that our strictly causal xAR–xAR-Fuse architecture, together with the streamable audio encoder, is particularly effective at tracking prosody and timing under zero–look-ahead constraints, while still delivering realism and smoothness comparable to the strongest offline system. Overall, the user study confirms that a fully streaming model can match or nearly match the perceptual quality of state-of-the-art offline methods, while providing better perceived synchrony with speech.

F. Impact of Codebook Size in Streamable Vector-Quantized Motion Tokenizer

Table 3. Effect of codebook size in the SVQ motion tokenizer on full-body gesture generation. Larger codebooks increase representational capacity and yield consistent gains across all metrics.

Codebook size	FGD (↓)	BC (→)	Diversity (↑)
512 entries	6.63	0.734	12.14
1024 entries	5.13	0.774	13.21
2048 entries	4.51	0.794	13.91

Table 3 examines how the size of the region-specific codebook $C^{\text{region}} = \{c_k\}_{k=1}^{K}$ in the SVQ motion tokenizer affects downstream gesture generation. With a small codebook (K=512), many continuous latents z_τ^{region} are mapped to the same discrete code, leading to quantization collisions that restrict the expressiveness of the motion token sequence $\{x_t^r\}$. This loss of granularity degrades realism (higher FGD) and weakens prosodic alignment (lower BC), as subtle variations in timing and amplitude cannot be preserved. Increasing the codebook to K=1024 reduces collisions and provides a richer set of motion primitives, improving both synchrony and diversity. The best performance is obtained with K=2048, where the token inventory is sufficiently large to capture finer-grained spatial and temporal variations while still remaining learnable under causal decoding. The resulting discrete representation enables xAR and xAR-Fuse to model expressive region-level dynamics more faithfully, yielding improved realism (FGD), stronger rhythm alignment (BC), and greater motion variability (Diversity).

Table 4. Comparison of causal self-attention and causal audio–motion cross-attention in the Hierarchical Autoregressive Transformer (HAR). All models maintain strictly aligned tokenization rates between audio tokens $\{a_t\}$ and motion tokens $\{x_t^r\}$ for fair comparison.

Method	FGD (↓)	BC (→)	Diversity (↑)
Causal self-attention	4.63	0.784	12.53
Causal cross-attention	**4.57**	**0.794**	**13.91**

G. Causal Attention Design in xAR and xAR-Fuse

Table 4 compares two causal attention strategies used in the hierarchical autoregressive transformer. In the "causal self-attention" variant, audio and motion embeddings at each time step are combined by elementwise addition,

$$u_t = x_t^r + a_t,$$

and the transformer attends only over the fused sequence $u_{1:t}$. While this preserves causality and keeps audio and motion aligned at the same token rate, the additive fusion collapses modality structure: the model cannot distinguish which features originate from audio and which from motion, and it cannot form directed cross-modal queries. As a result, the network must implicitly disentangle prosodic cues from the blended representation, which weakens rhythm conditioning and reduces expressive variability, leading to higher FGD (4.63), lower BC (0.784), and reduced Diversity (12.53).

In contrast, the "causal cross-attention" variant maintains separate motion queries and audio keys/values, enabling each motion token x_t^r to directly attend to synchronized audio cues $\{a_{1:t}\}$ through an explicit cross-modal pathway. This structured alignment allows the model to selectively extract energy changes, prosodic beats, and local speech dynamics essential for gesture timing and expressiveness. Consequently, causal cross-attention yields systematically better results (FGD = 4.57, BC = 0.794, Div. = 13.91), showing that explicit causal audio–motion conditioning is more effective than additive fusion for real-time gesture generation.

H. Role of Classifier-Free Guidance in LiveGesture

Table 5. Effect of classifier-free guidance scale γ on streaming gesture generation.

γ	FGD↓	BC→	Div.↑
1.00	4.61	0.781	12.90
1.25	4.57	0.794	13.91
1.35	5.23	0.763	12.80
2.00	6.42	0.756	11.74

Table 5 examines how the classifier-free guidance scale γ affects the token prediction distribution in *LiveGesture*. During inference, the guided logits for each region r are computed as

$$\ell^r_{\text{guided}} = \ell^r_{\text{uncond}} + \gamma \left(\ell^r_{\text{cond}} - \ell^r_{\text{uncond}}\right),$$

which amplifies the influence of audio-conditioned predictions while preserving causal decoding. When γ is too small ($\gamma = 1.00$), the model underutilizes modality conditioning, resulting in weaker synchronization and reduced expressiveness. Moderate guidance ($\gamma = 1.25$) provides the best balance, sharpening the conditional distribution enough to strengthen prosodic alignment (highest BC) and support rich gesture variability (highest Diversity) without oversuppressing natural motion variability, leading to the lowest FGD. Increasing γ further ($\gamma \geq 1.35$) over-amplifies conditional logits, causing overly deterministic predictions that reduce Diversity and destabilize temporal dynamics, which ultimately harms realism and synchrony under streaming constraints. These results demonstrate that *LiveGesture* benefits from moderate classifier-free guidance, which reinforces audio–motion coupling while maintaining the flexibility needed for expressive full-body gestures.

I. Role of Compositional vs. Full-Body SVQ Tokenization

Table 6. Comparison between a single full-body SVQ tokenizer and compositional per-region SVQ tokenizers. Both variants use the same total codebook capacity (2048 entries with 128-d embeddings).

Method	FGD↓	BC→	Div.↑
Full-body SVQ (1 tokenizer)	6.84	0.753	11.23
Per-region SVQ (4 tokenizers)	4.57	0.794	13.91

Table 6 compares two approaches for streamable motion tokenization under identical codebook capacity (2048×128). A single full-body SVQ must quantize the entire SMPL-X pose vector into a single latent stream z_τ, forcing one codebook C to represent heterogeneous motion patterns spanning upper body, lower body, hands, and face. This produces severe quantization interference: high-frequency regions (e.g., hands) and low-frequency regions (e.g., torso) compete for the same discrete codes, leading to token collisions and loss of fine-grained structure. As a result, the downstream autoregressive models receive less informative tokens x_t, which degrades realism (higher FGD), weakens prosodic synchronization (lower BC), and suppresses expressive variability (lower Diversity).

In contrast, the compositional design factorizes the motion stream into region-specific latent sequences z_τ^r with their own codebooks C^{region}. This specialization enables each SVQ to capture the appropriate temporal and spatial scale of its region without cross-region interference. The resulting tokens x_t^r preserve fine-grained dynamics and yield much richer conditioning signals for xAR and xAR-Fuse, improving FGD, BC, and Diversity as shown in Table 6.

J. Ablation on Role of Region Masking in Fusion Training

Table 7. Effect of random region masking (RM) on fusion training. $p_{\text{drop}} \sim \mathcal{U}(0, a)$ denotes the range of probabilities used to fully mask a region's token trajectory.

p_{drop} range	FGD↓	BC→	Div.↑
$\mathcal{U}(0, 0.2)$	4.57	0.794	13.91
$\mathcal{U}(0, 0.3)$	4.85	0.753	13.14
$\mathcal{U}(0, 0.5)$	5.30	0.770	12.82

Table 7 analyzes the effect of region-level masking in Stage 2 of fusion training, where an entire region's token history $\{x_{1:t}^r\}$ is removed with probability $p_{\text{drop}} \sim \mathcal{U}(0, a)$. Moderate masking ($a = 0.2$) yields the best performance because it gently exposes the fusion transformer to incomplete cross-region cues while preserving enough valid context to learn stable spatio–temporal dependencies among regions. As a increases, masking removes critical information from multiple regions, forcing xAR-Fuse to infer missing motion solely from $\{a_{1:t}, w_{1:t}\}$ and the remaining regions. This degrades the quality of the fused representations $\tilde{h}_t^r$ and weakens whole-body coordination, leading to reduced synchrony (lower BC), reduced variability (lower Diversity), and larger distribution drift (higher FGD). These results confirm that region masking is beneficial only when corruption remains mild, allowing the fusion model to learn robustness without collapsing inter-region dynamics.

K. Impact of Noise Injection in Local Region-eXperts

Table 8. Effect of noise injection in local region-eXperts during Stage 1 training. $p_{\text{noise}} \sim \mathcal{U}(0, a)$ determines the probability of adding Gaussian noise to embedded history tokens.

p_{noise} range	FGD↓	BC→	Div.↑
$\mathcal{U}(0, 0.2)$	4.57	0.794	13.91
$\mathcal{U}(0, 0.3)$	4.67	0.773	13.01
$\mathcal{U}(0, 0.5)$	5.30	0.760	12.63

Table 8 evaluates the effect of noise injection into the embedded history tokens of each region-eXpert, where noise is applied with probability $p_{\text{noise}} \sim \mathcal{U}(0, a)$. Light noise ($a = 0.2$) improves robustness by preventing overreliance on perfectly clean token histories, which rarely occur during causal autoregressive inference. This helps each local eXpert learn stable conditional distributions

$$p_\phi^r\big(x_t^r \mid x_{1:t-1}^r, a_{1:t}, w_{1:t}\big),$$

resulting in better synchrony and variance during downstream fusion. However, larger a introduces excessive corruption early in training, degrading the temporal structure in the latent sequences and disrupting the mapping between region tokens and audio cues. This harms both BC and Diversity, and increases FGD due to over-regularization. These results show that mild stochastic perturbation is sufficient to improve streaming robustness, whereas heavy noise erodes the fine-grained temporal patterns essential for expressive gesture generation.

L. Ablation on Attention Depth in xAR and xAR-Fuse

Tables 9 and 10 show the effect of increasing the depth of causal attention in both the local Region-eXpert transformers (xAR) and the global fusion transformer (xAR-Fuse).

In xAR, raising the number of causal audio–motion cross-attention layers and causal temporal self-attention layers from two to three consistently improves FGD, BC, and Diversity. Under strict zero–look-ahead constraints, deeper causal modeling allows each expert to more effectively capture fine-grained dependencies in the joint sequence $(x_{1:t-1}^r, a_{1:t})$, improving rhythm sensitivity and region-specific temporal expressiveness.

Table 9. Effect of causal audio–motion cross-attention depth and causal temporal self-attention depth in the Region-eXpert Autoregressive Transformer (xAR).

Cross-attn layers	Temporal-attn layers	FGD↓	BC→	Div.↑
2	2	4.60	0.791	13.42
3	3	**4.57**	**0.794**	**13.91**

Table 10. Effect of causal audio–motion cross-attention depth and causal spatial–temporal attention depth in the fusion transformer (xAR-Fuse).

Cross-attn layers	Spatio–temporal layers	FGD↓	BC→	Div.↑
2	2	4.65	0.784	13.35
3	3	**4.57**	**0.794**	**13.91**

A similar trend is observed for xAR-Fuse. Increasing the number of causal cross-attention layers and spatio–temporal fusion layers from two to three enhances FGD, BC, and Diversity. The deeper configuration performs more rounds of causal spatial reasoning over region embeddings $\{h_t^r\}$ and more extensive temporal reasoning over global motion history, resulting in stronger whole-body coordination and improved audio–motion synchrony.

Overall, the deeper (3+3)-layer configuration yields the best performance in both xAR and xAR-Fuse. Under strictly causal conditions, both local and global modules benefit from increased attention depth, compensating for the absence of future context and producing coherent, expressive, rhythm-aligned motion in real time.

References

[1] 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. 3

[2] 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. 4

[3] 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. *arXiv preprint arXiv:2203.05297*, 2022. 4, 5

[4] Haiyang Liu, Zihao Zhu, Giorgio Becherini, Yichen Peng, Mingyang Su, You Zhou, Naoya Iwamoto, Bo Zheng, and Michael J Black. EMAGE: Towards Unified Holistic Co-Speech Gesture Generation via Masked Audio Gesture Modeling. *arXiv preprint arXiv:2401.00374*, 2023. 4, 5

[5] Pinxin Liu, Luchuan Song, Junhua Huang, and Chenliang Xu. Gesturelsm: Latent shortcut based co-speech gesture generation with spatial-temporal modeling. In *IEEE/CVF International Conference on Computer Vision*, 2025. 4, 5

[6] Jinbo Xing, Menghan Xia, Yuechen Zhang, Xiaodong Cun, Jue Wang, and Tien-Tsin Wong. Codetalker: Speech-driven 3d facial animation with discrete motion prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12780–12790, 2023. 4

[7] 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 TOG*, 39(6), 2020. 3