MSc Artificial Intelligence
Master Thesis

Generating Dyadic Context-Aware Gestures for Virtual Avatars via Causal Transformers

by
David Werkhoven
13909495

2026/01/05 – 2026/07/17
36 ECTS

Supervisor: Dr. Esam Ghaleb
University Supervisor: Prof. dr. R. Fernández

Examiner: Dr. J.P. Trujillo

Universiteit van Amsterdam
Instituut voor Informatica

Generating Dyadic Context-Aware Gestures for Virtual Avatars via Causal Transformers

Abstract

Traditional co-speech gesture generation relies on non-causal models such as diffusion or bidirectional transformers, which require future audio and thus prevent real-time deployment. Existing models are also mostly optimized for single-speaker settings, failing to capture the dynamic feedback loops of dyadic interactions. This work introduces a context-aware, style-conditioned auto-regressive framework that generates, upper-body gestures frame-by-frame on-demand.

The approach uses a decoupled, two-stage pipeline trained on the large-scale Seamless Interaction dataset. In Stage 1, an upper-body RVQ-VAE compresses SMPL-H joint rotations, into a compact, hierarchical vocabulary of motion tokens. Training on a filtered, high-motion 13.1 hour subset avoids codebook collapse and provides a natural denoising prior. In Stage 2, a lightweight decoder-only causal transformer, predicts these motion tokens step-by-step. To avoid the quadratic self-attention costs of flattened sequences, a decoupled prediction protocol is implemented: the transformer auto-regressively predicts the broad posture (Layer 1), while a cross-attention residual predictor resolves detailed hand and finger movements (Layers 2–4).

To include contextual features, the transformer uses FiLM layers to integrate personality vectors and audio volume, while cross-attention over speaker and partner audio streams keeps timing in sync. An open-source, headless web-based rendering pipeline is also introduced to speed up visual evaluation. Benchmarks and ablation studies show that the proposed baseline model, generates co-speech movements matching conversational rhythms in a strictly causal, streaming manner.
To view the code and visual results, please visit:
https://pirateera.github.io/thesis-project-page/

Contents

List of Figures

List of Tables

List of Abbreviations

SOTA	State of the art
VQ-VAE	Vector-Quantized Variational Autoencoder
RVQ-VAE	Residual Vector-Quantized Variational Autoencoder
IPC	Interpersonal Circumplex
SMPL	Skinned Multi-Person linear
FiLM	Feature-wise Linear Modulation
LMDB	Lightning Memory-Mapped Database
DPP	Distributed Data Parallel
LLM	Large Language Model
EGL	Embedded-System Graphics Library
GT	Ground Truth
BC	Beat Consistency
FD	Fréchet Distance

Acknowledgements

This thesis marks the final result of an incredibly rewarding journey, and it would not have been possible without the guidance, support, and collaboration of many individuals. Beyond building technical skills, I had the privilege of working with brilliant researchers who became close colleagues and friends.

First and foremost, I would like to express my gratitude to my university supervisor at the University of Amsterdam (UvA), Prof. Raquel Fernández. Thank you for granting me the opportunity to collaborate with Dr. Esam Ghaleb and for giving me the academic freedom to pursue such a novel, experimental, and challenging project. Your trust and support allowed me to explore this research path with confidence.

I am incredibly grateful to my daily supervisor, Dr. Esam Ghaleb, for his constant guidance, care, and encouragement throughout every stage of this thesis. Esam, your mentorship has been invaluable, and your dedication to helping me navigate the complexities of this research kept me motivated even during the most demanding phases of the project.

I am also very appreciative of how I was treated as an MSc student under the wing of the Max Planck Institute (MPI). I am grateful for the trust the institute placed in me and, in particular, for the generous allocation of computational resources on the MPI computing cluster. Having access to this high-performance infrastructure was crucial, allowing me to develop the valuable skill of training large-scale, computationally intensive models that require long-term optimization pipelines.

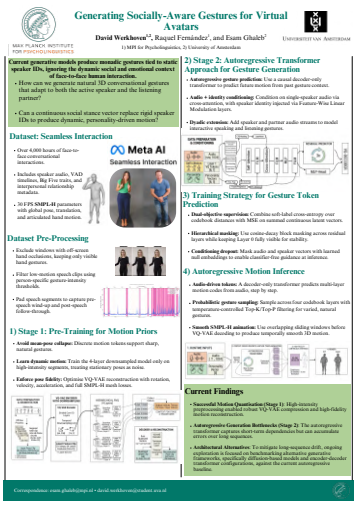
During my time at the MPI, I was fortunate to engage in many stimulating discussions with my colleagues. I want to thank Ferdinand Paar for the countless hours we spent brainstorming the theoretical foundations of co-speech gesture generation and sharing insights on sequence modeling. I also extend my sincere thanks to Lanmiou Liu, whose professional insights and expertise was key to helping me overcome technical issues when I was stuck training the Stage 1 reconstruction model.

Additionally, presenting this work as a poster presentation at the Netherlands Conference on Computer Vision (NCCV 2026) alongside Dr. Esam Ghaleb was a highlight of this journey. This experience was not only very enjoyable but also underscored the real-world value of the research. Engaging with the broader computer vision community confirmed that releasing the open-source code-base will provide a meaningful, structured utility for researchers eager to work with this newly introduced dataset.

Finally, I want to thank my family and friends for their support and encouragement throughout my master's program. You kept me grounded and supported me through every challenge.

(a) Poster presentation at NCCV 2026.

(b) Explaining co-speech gesture generation.



(c) The research poster.

Figure 1: Presenting our research poster and discussing the work at the Netherlands Conference on Computer Vision (NCCV 2026).

Chapter 1

Introduction

The rapid growth of AI-driven conversational agents has fundamentally changed how we as humans interact with computers, moving past simple text-based interfaces towards sophisticated embodied systems. As virtual agents become more common in areas like digital communication, education, and entertainment, the focus of synthetic interaction has moved from verbal fluency to the details of how they appear physically [6]. For these agents to be truly effective in face-to-face environments, they must do more than simply generate text. They must embody the multi-layered nature of human communication, in which spoken language is deeply connected to facial expressions and body movements [37].

In human conversation, these non-verbal signals are essential to understanding the full picture because they work together to provide key spatial and emotional context [11]. Rather than being an addition to the conversation, these movements are mandatory for turn-taking conversation, and reinforcing the speaker's semantic intent [36]. As we develop more advanced virtual avatars for AR and VR applications, the challenge lies in shifting these agents from robotic, pre-programmed repetitions toward dynamic, expressive motion that feels genuine to the human observer [7].

However, the current approaches for co-speech gesture generation face a major bottleneck. Most state-of-the-art (SOTA) architectures are optimised for single-speaker monologue scenarios, learning motion patterns from rigid, discrete Speaker-ID labels [38]. This approach largely ignores the fluid, two-way nature of dyadic conversation, where gestures are not just determined by the speaker's individual habits, but are continuously adapted to the social stance, emotional state, and relational context of the conversational partner [5]. Furthermore, as the field shifts toward real-time interactive agents, relying on offline models that depend on future speech context creates a disconnect between the generation process and the rapid, two-way timing required for natural human-level responses [33].

1.1 Problem Definition

Current co-speech gesture generation frameworks struggle to balance real-time performance with the complexities of two-way social interaction and limited control over meaning [33]. Existing solutions often trade one dimension for another by using non-causal offline diffusion for visual quality, forcing two-way interactions into single-user architectures, or letting rhythmic movement drown out specific meaningful gestures [38, 20]. This chapter outlines this research gap by comparing the proposed framework against key SOTA models across three specific dimensions: the operational domain (monadic vs dyadic) (1.1.1), the generation task (beat vs semantic gestures) (1.1.2), and the underlying machine learning method (non-causal vs causal architectures) (1.1.3). These sections build the foundation for the main hypotheses: *a decoder-only auto-regressive transformer, supervised by a multi-component loss strategy, conditioned on continuous social stance vectors. Can achieve context-aware dyadic gesture generation while maintaining strict motion causality with chunk-based audio conditioning.*

The core challenge lies in generating natural, full-body gestures that align with speech in real-time dyadic environments [20, 29]. Beyond this, existing models struggle to balance the frequent beat gestures typical of conversational datasets, with rare, meaningful motions, such as pointing or iconic gestures that illustrate specific actions. Because of this, generation processes often let continuous audio rhythm take over, which washes out the specific signals that actually match what the speaker means [38]. Furthermore, these systems must achieve low-latency inference to maintain the rapid, two-way timing essential for human interaction, a requirement often unmet by non-causal or computationally heavy diffusion approaches [20, 38, 18, 27].

From a machine learning perspective, these continuous audio rhythms dominate, because low-level audio features like amplitude and pitch, track closely with rapid hand beats, providing a dense training signal that naturally overpowers rare semantic expressions. Meanwhile, diffusion-based and non-causal approaches remain structurally unfit for real-time streaming because they either require a global look-ahead window of future speech context, or rely on heavy, multi-step iterative denoising passes to generate a single motion sequence.

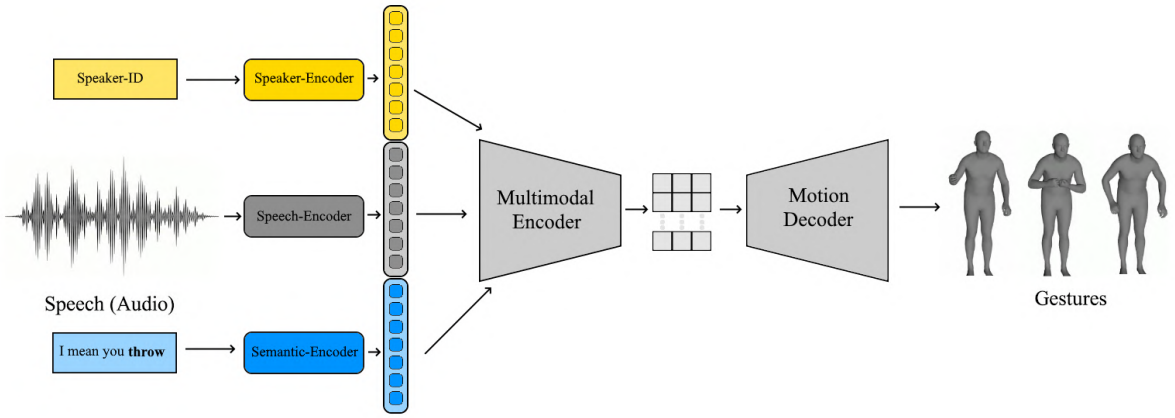


Figure 1.1: High-level overview of a speech-to-gesture generation framework, illustrating the parallel processing of Speaker-ID, acoustic features, and textual semantics through dedicated encoders before discrete mapping and motion decoding into 3D gestures.

1.1.1 The Domain Gap: Monadic vs. Dyadic Gesture Generation

Conversational gestures are naturally social, shaped by the speaker's personality and their dynamic relationship with the listener. Existing models train on monadic, monologue-heavy datasets and rely on rigid speaker-id embeddings. As a result, they fail to capture the emotional basis and social stance required for a virtual agent to adapt to changing conversational settings [5, 4]. Because a standard speaker identifier functions as a static, one-hot categorical index, it limits the architecture to reproducing a fixed personal average of an individual's motion habits. It possesses no inherent capacity to modulate that style based on conversational variables, rendering the generated gestures completely static regardless of whether the agent is addressing a stranger or a close companion.

Early gesture generation models focused almost exclusively on monadic environments, learning motion mappings from datasets where individuals speak in social isolation [26, 27, 38, 20]. This focus has been largely driven by the reliance of the *BEAT2* dataset [16], which has established itself as the industry standard for SOTA monadic frameworks such as *SemTalk* [38] and *GestureLSM* [20]. While these models perform well at modeling a single speaker, they still fail to capture the nature of two-way human communication.

Early attempts to close this gap toward dyadic interactions, faced data representation issues. Works like *ConvoFusion* [26] introduced early dyadic datasets, but their limited scale meant the broader research community largely remained focused on monadic benchmarks. Subsequently, *Audio2Photoreal* [27] advanced dyadic interaction gesture generation, but specifically optimized its pipeline for photorealistic avatar reconstruction rather than the parametric SMPL [23] body models preferred by the gesture generation community.

The recent introduction of the large-scale Seamless Interaction dataset by Meta [2], has allowed a shift toward dyadic gesture generation. By providing high-quality, two-way SMPL-H motion [32], paired with rich social annotations. A notable example using this new dataset is *DyaDiT* [29], which makes use of it to generate responsive gestures by incorporating both partners audio alongside discrete relationship tokens and Big Five personality scores [24], instead of a rigid Speaker-Id. However, *DyaDiT's* primary technical contribution focuses on resolving the acoustic entanglement of a dual-stream audio pipeline via Organizational Cross Attention (ORCA), treating the social context as a static, secondary conditioning signal.

As a result, a domain gap still remains in how dyadic agents interpret and manifest social dynamics. While the field has successfully started the transition to two-way audio-motion data, current models under-utilize the available interpersonal metadata. This research addresses this gap by investigating how continuous social stance features can go beyond the static labels and how they can be modeled to drive contextually appropriate gesture generation.

1.1.2 The Task Gap: Rhythmic Beat Dominance vs Sparse Semantic Control

Since the field has focused on monadic gesture generation, substantial research effort has been directed toward categorizing and accurately generating specific types of co-speech movement. Literature standardizes human gestures into five primary classifications: Iconic, Deictic, Beat, Emblematic and Metaphoric [36]. From a generative modeling perspective, these classifications broadly merge into two distinct behavioral groups, continuous rhythmic beats that align with the rhythm and tone of speech, and sparse semantic gestures which are used to represent physical or abstract ideas.

A limitation in current architectures is the distributional imbalance between these two groups. Rhythmic beat movements

make up most of the co-speech gestures during conversation. Whereas semantically rich actions are comparatively rare. Because standard deep learning networks optimize for the most frequent data patterns, they fall victim to this unevenness. Allowing continuous audio features to dominate the generation process. Consequently, the network averages out rare, meaningful movements, causing expressive semantic gestures to be washed out by a baseline of generic rhythmic beats [38].

To enforce semantic control, recent monadic frameworks like *SemTalk* [38] and *SemGes* [18] explicitly target this rhythmic dominance. For instance, SemTalk decouples motion into a rhythm-aligned base and a semantic-aware sparse layer, making use of a gating mechanism to adaptively trigger semantic gestures at specific key frames.

While the primary scope of this work does not focus on designing a novel semantic gating mechanism, addressing this task gap is relevant. The architectural insight and control mechanism developed to preserve sparse semantics in monadic pipelines, are theoretically transferable to dyadic environments.

1.1.3 The Method Gap: Non-Causal Frameworks vs. Causal Auto-regressive Transformers

The currently used machine learning architectures for audio-to-motion gesture generation have been dominated by non-causal frameworks, specifically diffusion-based strategies and bidirectional encoder-decoder models. Diffusion and flow-matching models such as *DyaDiT*[29], *ConvoFusion*[26], *Audio2Photoreal*[27], and *GestureLSM*[20] were adopted because they are good at at solving the "one-to-many" probabilistic mapping problem of human motion. In contrast, bidirectional masked models like *EMAGE*[17] rely on encoder-decoder architectures to leverage full past-and-future speech context, ensuring smoothed and synchronized motion. However, these architecture choices enforce an offline setup. Diffusion requires multi-step iterative refinement, while bidirectional encoders make use of a "look-ahead" window of future audio. This creates a latency limitation that is incompatible with the rapid, zero-look-ahead responsiveness required for true, live dyadic interaction.

The field's long-standing dependence on these non-causal, offline models is tied to the limits of available training data. As mentioned earlier, the *BEAT2*[16] dataset has served as the SOTA dataset for gesture generation. However, at roughly 60-76 hours of total interaction, it lacks the needed scale required to successfully train, data-hungry auto-regressive transformers without overfitting. Diffusion models and encoder-only architectures are comparatively more sample-efficient at extracting accurate priors for this limited data scale, forcing researchers to accept the latency trade-off in exchange for high-quality co-speech gesture generation [1].

Despite these data limits, recent breakthroughs show that auto-regressive generation is possible on the *BEAT2*[16] dataset. *MIBURI* [25] successfully trained an online, causal framework on this restricted dataset by offloading multi-modal feature extraction to the pre-trained *MOSHI* [8] model and aggressively quantizing the motion space, thereby avoiding the high data demands typical of auto-regressive transformers.

However, while *MIBURI*[25] and similar token-based models like *LiveGesture*[33] successfully achieve causal streaming, they remain limited to the monadic monologue inputs and rely on pre-trained foundation shortcuts. Transferring a strictly causal auto-regressive approach into the dyadic domain, where the trained model accurately learns the process of co-speech gesture generation requires a major increase in raw data. Therefore this work makes use of the recent release of the Seamless Interaction dataset [2], which provides approximately 4000 hours of conversational data. By making use of this massive dataset, this work explores whether an auto-regressive transformer can be scaled for co-speech gesture generation while maintaining strict motion causality, despite making use of chunk-based audio conditioning. This approach bypasses the latency constraints of diffusion and bidirectional transformer architectures, enabling real-time, expressive dyadic motion generation.

Because recent advancements have only just introduced the dyadic data necessary to capture these interaction complexities, there are no existing frameworks to adapt, requiring the introduction of a completely novel, built-from-scratch architecture and data pipeline.

1.2 Scope of Research

While context-aware gesture generation is a large field, this project focuses on testing a self-designed auto-regressive model for generating upper-body gestures in a two-way interaction. Through ablation studies, this work evaluates how speaker audio, partner audio, social stance, personality, and relationship affect its generation capabilities.

To capture these multi-modal interaction complexities, this study makes use of the recently released Seamless Interaction dataset [2], which marks a significant shift from the monologue-heavy, monadic benchmarks that historically dominated the field. By providing thousands of hours of authentic, face-to-face conversational pairings, this dataset offers a robust foundation for capturing behavioral loops, conversational rhythms, and turn-taking dynamics. Importantly, it moves beyond static speaker profiles by incorporating continuous relational metadata and psychological attributes, making it uniquely viable for driving context-aware generation.

Modeling motion in real-time requires framing gesture generation as a causal sequence task using a decoder-only transformer.

Training this token-based transformer on human motion needs an optimization pipeline that maps categorical choices back to physical coordinates. To keep the movements from looking robotic, the framework uses a dual optimization strategy pairing continuous space alignment with geometry-aware error correction. This setup defines the boundary conditions and experimental design choices used in this work.

To maintain the feasibility within the project's time-frame, this work introduces three primary constraints:

Project Scope and Constraints

Motion Domain: This work is limited to upper-body and hand gesture generation. Lower-body motion is held at a static pose to avoid limitations related to the available motion data, while facial expressions are excluded due to the restricted availability of the necessary interpretation models.

Dataset Selection: While the Seamless Interaction dataset [2] offers a lot of information, this research uses a smaller subset of the data. The project scope includes developing a pipeline to filter for high-motion clips and match audio with movement, ensuring the model learns from clear, expressive data instead of repetitive or noisy samples.

Generation Method: This project uses a decoder-only model to treat gesture generation as a sequence of predictions. The project focuses on a multi-component loss strategy, combining latent mean squared error loss with a distance based soft-label cross-entropy loss to simplify the process compared to more complex diffusion or encoder-decoder methods.

1.3 Research Questions & Objectives

While current SOTA co-speech gesture generation frameworks rely heavily on non-causal architectures like multi-step diffusion or bidirectional transformers [20, 38], they remain inherently limited by multi-step denoising latencies or a dependency on future speech context. A style-conditioned auto-regressive framework is uniquely suitable for this task, because it models gesture generation as a strict, causal sequence, enabling the low-latency streaming required for live virtual agent deployment while dynamically adapting body posture to live conversational cues. To evaluate the effectiveness of this causal approach in multi-modal, dyadic gesture generation, this work poses the following research questions and objectives:

Main Research Question

How can a context-aware, style-conditioned auto-regressive framework leverage curated dyadic interaction data to generate accurate, upper-body conversational gestures?

To address the main research question, this work is divided into two main objectives and four sub-questions, which together establish the necessary dataset and architecture to properly evaluate the topic.

Main Objective 1: Data Foundation & Motion Representation To answer the main research question, this work requires a strong data foundation to connect the raw two-way interactions with high-quality motion modeling. As detailed in Chapter 3, a custom pipeline is used to filter raw audio-motion streams from the Seamless Interaction dataset, ensuring pose accuracy and isolating high-intensity conversational behaviors. Building on this, a Hierarchical Residual Vector Quantized Variational Autoencoder (RVQ-VAE) is implemented to learn a latent motion space that captures expressive dynamics, a process described in Chapter 4. The modeling approach, covered in Chapter 5, is then validated through a custom rendering pipeline detailed in Chapter 6. This pipeline creates visuals of the reconstructed and generated SMPL-H body models for direct comparison against ground-truth data. To systematically achieve this, Main Objective 1 is broken down into three core sub-objectives:

Objectives of Data Curation & Representation

Data Foundation A pipeline is developed to filter and segment raw dyadic streams into a high-fidelity subset focused on upper-body gesture intensity.
Latent Representation Hierarchical RVQ-VAE is trained to derive a compressed, expressive latent space from the curated motion data.
Rendering Pipeline A visualization framework is constructed for SMPL-H and SMPL-X body models to perform qualitative assessments through side-by-side comparisons with ground-truth sequences.

Research Questions for Main Objective 1

RQ1 How can raw dyadic audio-motion streams be systematically filtered and segmented to maintain pose accuracy while isolating high-intensity upper-body conversational behaviors?
RQ2 To what extent can a Hierarchical RVQ-VAE achieve accurate generalization when trained exclusively on a limited subset of high-intensity conversational motion?

Main Objective 2: Generative Modeling & Contextual Inclusion To address the second part of this research, two more sub-objectives and two sub-questions are introduced. First, it evaluates how well a decoder-only, auto-regressive transformer. Models co-speech gestures using a multi-component loss strategy to improve gesture generation compared to alternative methods. Second, it measures how much, including partner audio and continuous personality vectors from the dataset improves the model's awareness of two-way social context. The evaluation of this contextual inclusion is addressed in Chapters 5 and 7.

To systematically achieve this, Main Objective 2 is broken down into two core sub-objectives:

Objectives of Architecture Design & Contextual Integration

Generative Modeling A decoder-only auto-regressive transformer is implemented to generate continuous upper-body gestures. Employing a multi-component loss strategy to ensure high-fidelity motion output.
Dyadic Context-Awareness Cross-modal partner audio and personality-conditioned embeddings are integrated to inform and refine the generation process.

Research Questions for Main Objective 2

RQ3 How effectively can a decoder-only auto-regressive transformer, leveraging a multi-component loss strategy, model continuous co-speech gestures compared to traditional generative architectures?
RQ4 To what extent does integrating partner audio and continuous personality vectors enhance the dyadic context-awareness of the generated gestures?

Chapter 2

Background

This chapter covers theoretical and mathematical concepts needed to understand this work, focusing on the overlap of cognitive communication science and generative modeling. Instead of a broad summary of deep learning architectures, this background defines the exact inputs, outputs, and pipeline representations across five primary sections.

First it outlines the cognitive realities of multi-modal communication, including human gesture taxonomies and the behavioral frameworks that dictate how personality and social stance manipulate physical motion (2.1). It then translates these concepts into kinematic human body representations, formally defining the SMPL body models used to project raw human motion into a digital coordinate space (**??**). From there, the chapter introduces the mechanics of neural discrete latent spaces, focusing on how standard Vector-Quantized Variational Autoencoders (VQ-VAEs) and the layered structure of Residual Vector Quantization (RVQ-VAEs), represent continuous motion in a learnable latent space (2.3). Afterwards, Section 2.4 establishes the mathematical approaches of sequence modeling, defining the functional boundaries between causal auto-regressive streaming and non-causal offline architectures. Finally, it outlines the multi-modal feature extraction pipelines that transform raw audio and text into the standardized, conditional embedding vectors driving the generative network (2.5).

2.1 Foundations of Multi-modal Communication

To create artificial agents, we must first understand the cognitive and social drivers of human movement. This sections explores how speech and gesture unify, how movements are categorized and how social context influences behavior, providing the physiological foundation for co-speech gesture generation.

2.1.1 Speech and Gesture as a Unified System

In face-to-face dialogue, language operates as a single integrated system combining the voice, hands, face, head, torso and lower-body. The human brain processes these multi modal signals as a joined package, even when visual and audio cues are not perfectly synchronized. This asynchronous nature introduces two important cognitive challenges. *The binding problem*, which requires the observer to link disparate visual and audio signals into a cohesive message, and the *segregation problem*, which involves isolating intentional communicative movements from accidental background actions, such as scratching. Despite these complexities, visual cues act as a signal path that speeds up language comprehension by letting the listener predict spoken words before they are fully articulated [11]. For generative modeling, this confirms that body movements are essential components of the communication system rather than optional additions, helping to reduce ambiguity during human-computer interaction.

2.1.2 Taxonomy of Co-Speech Gestures

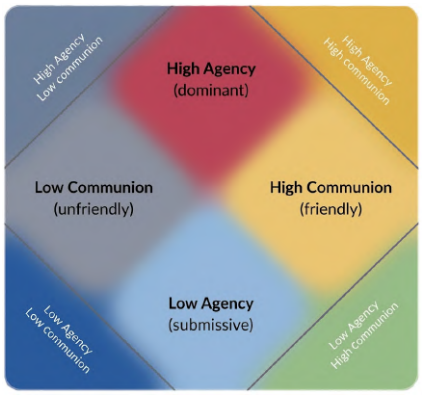
Co-speech gestures are visible actions produced concurrently with speech, grouped into five core types based on their meaning. Emblematic gestures have a conventionalized meaning, such as a thumbs up sign. These gestures function almost like words and can often be understood without accompanying speech. Iconic gestures resemble a physical aspect of the information being presented in the conversation. For example, a speaker might trace the shape of an object or mimic a specific movement direction with their hands to describe an entity. Metaphoric gestures are similar to iconic gestures in form but instead of showing real objects, these movements represent ideas. They give physical form to invisible concepts, holding an imaginary object to show a concept or moving the hands as if weighing two opposing arguments. Deictic gestures function as pointing movements. They direct attention to locations in space, which can refer to actual physical objects in the area or to abstract locations that represent specific topics. Beat gestures are simple, fast, and rhythmic movements. Unlike the other types, they do not directly relate to a semantic meaning. Instead, they refer to the process of speaking itself, synchronizing with the rhythm of speech to highlight words or structure the flow of the conversation [36]. From a machine learning perspective, these classifications broadly divide into two distinct operational targets, the frequent, continuous rhythmic beat gestures, and the much rarer, semantically rich sparse gestures (iconic, metaphoric, deictic, and emblematic) [38].

2.1.3 Theoretical Foundations of Social Context and Personality

From a cognitive and psychological perspective, generating gestures that are socially appropriate requires establishing a behavioral framework to model the speaker's internal traits and interpersonal environment, before these features are translated into technical conditioning signals. To map these human variables, the Five-Factor model (Big Five) summarizes personality through five core traits: Neuroticism, Extraversion, Openness, Agreeableness, and Conscientiousness. Extraversion, in particular, directly correlates with motion kinematics. Studies demonstrate that speakers with high extraversion scores produce gestures more frequently and with more expressive motion [24].

Furthermore, the dynamic relational stance between two speakers is modeled by the Interpersonal Circumplex (IPC). The IPC plots social interactions across two continuous dimensions, Agency (the vertical axis, ranging from dominant to submissive) and Communion (the horizontal axis, ranging from friendly to hostile) [22]. These relational factors dynamically adjust gesture behavior in real-time. For example, when speaking to a social superior, people often gesture less and keep their movements closer to their body [4]. Additionally, dyadic conversations naturally trigger the *chameleon effect*, a phenomenon where speakers unconsciously mimic the postures and mannerisms of their conversational partners [5]. Therefore, accurately generating realistic human motion requires conditioning architectures on continuous, dynamic social variables rather than static identity labels, ensuring the virtual agent adjusts its movement patterns to match changing social dynamics.

(a) Big Five personality model



(b) Interpersonal Circumplex model

Figure 2.1: Theoretical conditioning frameworks for personality and social context. (a) **Big Five personality model:** Overview of the Big Five personality model. These five traits (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) act as continuous conditioning variables within the framework, dynamically modulating the expressiveness, frequency, and overall style of the generated upper-body kinematics. (b) **Interpersonal Circumplex model:** Overview of the IPC model, mapping interpersonal dynamics along the continuous axes of Agency and Communion.

2.2 Kinematic Human Body Representation

To train gesture generation models, raw human movement must first be translated into a structured digital format. This section outlines the industry standard parametric body models and the kinematic hierarchies used to define the exact numerical outputs of co-speech gesture generation models.

2.2.1 The SMPL Body Model Family

The Skinned Multi-Person linear (SMPL) body model serves as the industry standard across modern gesture datasets, functioning as the mathematical framework to merge diverse motion recordings into a consistent 3D human representation [23]. The base SMPL model focuses exclusively on the main body joints and global shape. However, because human communication also relies on hand and finger movements, variations were developed to capture better anatomical detail.

The SMPL-H model introduces, fully articulated hands to the standard body model [32]. Building upon this, the SMPL-X model combines the body, hands, and an expressive face into a single, unified framework. This facial component is handled by the FLAME model, which ensures accurate capture of detailed facial expressions and jaw movements [28, 15].

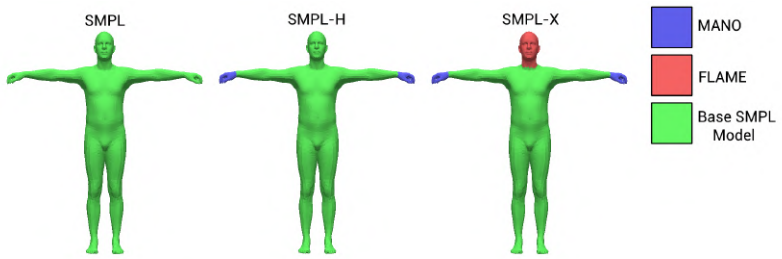


Figure 2.2: Comparison of the SMPL body model family, showing the base SMPL model (green), the included MANO hands in SMPL-H (blue), and the FLAME face used in SMPL-X (red).

2.2.2 Forward Kinematics and Rotation Spaces

The SMPL body model family parametrizes human postures through a hierarchal kinematic tree, where the overall configuration of the body is defined by the relative rotations of each joint with respect to its parent. To generate motion, generative networks do not typically predict raw 3D spatial coordinates directly, instead they output these local joint rotations to preserve bone-length consistency and physical plausibility [23].

Because this kinematic tree is interdependent, where the position of the fingers depends on the combined rotations of the shoulder, elbow and wrist. Generating cohesive full-body motion is computationally complex. This strict hierarchical dependency is exactly why some modern deep learning architectures choose to decouple and process different body regions, such as the hands and the torso, through separate generation channels before merging them [38, 19].

2.3 Neural Discrete Representation Learning

Generating 3D human motion is computationally challenging due to the high dimensionality of raw kinematic data. A single full-body pose requires processing hundreds of continuous variables per frame, such as high-dimensional vectors of 6D joint rotations. Mapping speech context directly to this massive state space is difficult. When standard neural networks attempt this direct mapping, they struggle to capture the underlying data distribution, resulting in flat, averaged movements known as the mean pose problem [9]. To avoid this, modern architectures transition from predicting raw continuous kinematics to learning a compressed latent motion space. Projecting large pose vectors into a low-dimensional representation allows networks to efficiently map audio to movement dynamics before decoding them back into full-body motion [34].

2.3.1 Vector-Quantized Variational Autoencoders (VQ-VAE)

Vector-Quantized Variational Autoencoders (VQ-VAEs) construct a discrete latent space by compressing continuous data into a finite codebook. Let $\mathcal{C} = \{e_1, e_2, \ldots, e_K\}$ represent a discrete codebook containing K unique embedding vectors, where each $e_i \in \mathbb{R}^D$ possesses a dimensionality of D. Given a continuous gesture posture feature vector $z \in \mathbb{R}^D$ extracted from an encoder network, the vector quantization operation $\mathcal{Q}(z; \mathcal{C})$ maps the continuous vector onto its nearest discrete code entry e_k using an L_2 distance metric [34]:

$$\mathcal{Q}(z; \mathcal{C}) = \text{argmin}_j \|z - e_j\|_2^2 \tag{2.1}$$

The selected code embedding e_k is then passed as input directly into a decoder network to reconstruct the continuous postural attributes. To train the encoder parameters, the decoder parameters, and the codebook entries simultaneously, the network optimizes a three-part objective function [34]:

$$\mathcal{L}_{\text{VQ-VAE}} = \mathcal{L}_{\text{recon}}(G, \hat{G}) + \|\text{sg}[z] - e_k\|_2^2 + \beta\|z - \text{sg}[e_k]\|_2^2 \tag{2.2}$$

where $\text{sg}[\cdot]$ represents the stop-gradient operator, which acts as an identity function during the forward computation pass but passes zero partial derivatives during backpropagation. The first term evaluates the structural reconstruction loss between the real and reconstructed gestures. The second term utilizes dictionary learning to pull the codebook embeddings toward the encoder outputs. The final term acts as a commitment constraint weighted by a hyperparameter β, forcing the encoder outputs to remain stable and adhere to the selected codebook indices [34].

2.3.2 Residual Vector Quantization (RVQ-VAE)

Standard VQ-VAEs require a codebook size that scales directly with the variety of gestures. This large vocabulary makes training unstable and causes codebook collapse when trying to capture detailed movements. Residual Vector Quantization (RVQ-VAE) solves this issue by using a fixed-size codebook $\mathcal{C}$ that recursively quantizes data across a defined sequence of D depth layers [14].

Starting with the initial continuous feature vector as the zero-indexed residual $r_0 = z$, the residual quantizer recursively computes the code selection k_d and the subsequent remaining residual vector r_d for each depth layer $d \in [1, D]$ [14]:

$$k_d = \mathcal{Q}(r_{d-1}; \mathcal{C}) \tag{2.3}$$

$$r_d = r_{d-1} - e_{k_d} \tag{2.4}$$

This layered approach refines the continuous motion vector step-by-step. Summing the codebook vectors up to depth d progressively rebuilds the original signal from a coarse outline to fine details [14]:

$$\hat{z}^{(d)} = \sum_{i=1}^{d} e_{k_i} \tag{2.5}$$

The final completely quantized output vector is represented as the full sum over all depth layers $\hat{z} = \hat{z}^{(D)}$. To keep the training path stable, the commitment loss evaluates the sum of quantization errors across all layers sequentially [14]:

$$\mathcal{L}_{\text{commit}} = \sum_{d=1}^{D} \|z - \text{sg}[\hat{z}^{(d)}]\|_2^2 \tag{2.6}$$

An RVQ setup running with a small codebook size of K entries across a quantization depth D possesses a total partitioning capacity equivalent to a standard VQ model configured with K^D unique entries. This shared codebook compresses motion data without increasing the parameter count. As a result, gesture generation models can process shorter token streams while still preserving fine details like subtle finger movements [14].

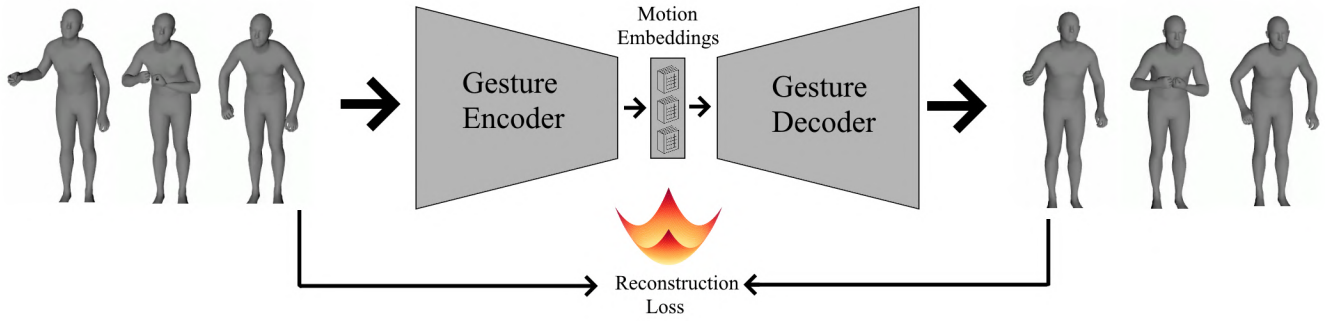


Figure 2.3: High-level overview of the discrete representation learning framework. Raw 3D human motion sequences are processed by a Gesture Encoder and compressed into a compact space of discrete motion embeddings. A Gesture Decoder then reconstructs the physical movement from these quantized tokens, with the entire system optimized via a reconstruction loss that minimizes the difference between the original and generated kinematics.

2.4 Causal vs. Non-Causal Approaches

In gesture generation, how a model handles time determines whether it can run in real-time. While many models achieve high quality by looking at the entire conversation at once, this type of approach prevents live streaming. This section analyzes the mechanics and trade-offs of the primary architectural approaches, that dominate co-speech gesture generation research, being bidirectional masking, sequence-to-sequence networks, and diffusion models. Evaluating these architectures provides a clear picture of how different modeling choices link audio and text to motion.

To mathematically standardize these approaches, the gesture generation task is defined using three core variables. Let $A = \{a_1, a_2, \ldots, a_T\}$ represent the speech audio sequence, and let $G = \{g_1, g_2, \ldots, g_T\}$ represent the target human gesture sequence, where T holds the total number of motion frames. Each motion frame g_t can be parameterized as continuous joint rotations or discrete motion tokens. Let $W = \{w_1, w_2, \ldots, w_M\}$ represent the corresponding text transcript tokens, where M holds the total number of words or linguistic tokens in the utterance.

Because spoken words and visual motion frames operate at completely different temporal scales, the sequence lengths M and T are usually different. While text sequence lengths are much shorter, some pre-processing pipelines use forced alignment to pad or duplicate the text tokens so they match the motion frame timeline T exactly. Across both setups, the objective of all approaches is to learn a mapping function $f : (A, W) \rightarrow G$.

Table 2.1: Conceptual comparison of sequential approaches in gesture generation.

Architectural Approach	**Input Vectors (X)**	**Output Target (Y)**	**Temporal Constraint**
Encoder-Only Masked	Full Audio $A_{1:T}$ + Masked Motion $G_{1:T}$	Reconstructed Motion $\hat{G}_{1:T}$	Non-Causal (Full Look-Ahead)
Encoder-Decoder	Full Audio $A_{1:T}$ + Full Text $W_{1:M}$	Reconstructed Motion $\hat{G}_{1:T}$	Non-Causal Bottleneck
Diffusion	Random Noise x_s + Audio Context $A_{1:T}$	Denoised Motion Trajectory x_0	Iterative Non-Causal Sampling

2.4.1 Bidirectional Encoder-Only Masked Architectures

An encoder-only model uses a single unified transformer network to process both the speech features and the motion data together. It is called encoder-only because it lacks a separate decoding stack. It directly maps the input sequences into the final output. In this setup, the architecture treats gesture generation as a sequence-completion task [17]. The network is given the complete audio sequence $A_{1:T}$ and text $W_{1:M}$ alongside a gesture sequence, where random frames are blanked out by a mask token e_{mask}. The model is then forced to fill in these corrupted gaps.

$$\hat{G}_{1:T} = f_\theta(G_{1:T}^{\text{masked}}, A_{1:T}, W_{1:M}) \tag{2.7}$$

This architecture processes the entire timeline at once because its self-attention layers calculate relationships across all timestamps simultaneously. This is driven by the scaled dot-product attention formula [35]:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.8}$$

Here, the queries (Q), keys (K), and values (V) are simply different linear projections of the input sequences. Because the resulting attention matrix (QK^T) cross-references every frame i with every frame j across the full timeline $[1, T]$, the model has full look-ahead access to future data. This requirement for the complete conversation timeline makes the framework entirely offline.

However, access to both past and future context naturally smooths the motion. This bidirectional view preserves body coherence and prevents the posture drift common in decoder-only models [17].

2.4.2 Sequence-to-Sequence Encoder-Decoder Transformers

Encoder-decoder models generate gestures by separating input processing from motion generation. This approach is split into two parts: an encoder that handles speech comprehension, and a decoder that handles motion generation. The encoder takes the raw audio $A_{1:T}$ and text $W_{1:M}$ and compresses them into a continuous, condensed mathematical representation. The decoder then reads this single representation to map out the final gestures [21].

First, the encoder E_ϕ analyzes long-range speech patterns and sentence structures across the entire timeline to produce conditional context embeddings:

$$F_{\text{speech}} = E_\phi(A_{1:T}, W_{1:M}) \tag{2.9}$$

Next, the decoder D_ψ reads this speech summary to generate the corresponding joint coordinates frame-by-frame. To ensure that the motion is continuous and smooth, the decoder generates the current posture $\hat{g}_t$ by conditioning on both the global speech embedding F_{speech} and its own previously generated motion history $\hat{g}_{<t}$ [21, 27]:

$$\hat{g}_t = D_\psi(\hat{g}_{<t}, F_{\text{speech}}) \tag{2.10}$$

Some pipelines take advantage of this setup to generate low-frequency, sparse guide poses at 1 fps as an intermediate structural step before running, high-frequency infilling models [27].

This architecture is strictly offline because of how the encoder handles time. The encoder uses bidirectional tracking to analyze how words and syllables relate to one another across the entire utterance. Because the resulting speech embeddings contain information from both the past and the future, the decoder cannot start generating the very first gesture frame until the encoder has processed the complete conversation clip [21, 20].

However, this separation makes the model effective for language alignment. It allows the network to map linguistic meaning and emotion directly to natural gestures without getting overwhelmed by raw joint noise [21, 27].

2.4.3 Diffusion Probabilistic Models

Diffusion architectures treat gesture generation as an iterative trajectory-denoising task. Instead of directly regressing joint coordinates from speech, these models define the generation task as the transformation of a random Gaussian noise sequence into a clean motion sequence [29]. Let X represent the temporal trajectory being generated, where the completely clean state X_0 matches the final target gesture sequence $G_{1:T}$.

During training, a forward diffusion process progressively corrupts the clean gesture sequence $G_{1:T}$ by injecting Gaussian noise over a defined sequence of steps s. A neural network ϵ_θ is then trained to predict the added noise or the latent velocity at step s, guided by a multi-modal speech context vector c extracted from the global audio $A_{1:T}$ and text $W_{1:M}$ streams:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{G_{1:T},s,\epsilon}\left[\|\epsilon - \epsilon_\theta(X_s, s, c)\|_2^2\right] \tag{2.11}$$

where X_s represents the noisy trajectory at step s. In conversational setups, the conditioning vector c combines the audio features of both speakers using cross-attention structures to explicitly model interpersonal dynamics, back-channeling, and turn-taking cues [29].

This probabilistic approach is effective because it directly solves the non-deterministic, "one-to-many" mapping problem of human communication [29, 27, 20]. In natural conversation, a single spoken phrase or text sequence can logically align with multiple completely different physical expressions. While deterministic models collapse to an unexpressive average posture, diffusion models capture this natural variety. By starting from different random noise inputs, the network maps the exact same speech signal to completely unique, realistic gesture tracks [29].

This architectural setup means diffusion models must operate entirely offline. Because the network denoises the entire gesture trajectory simultaneously, generating a posture at any given frame requires context vectors drawn from the full conversation timeline ($A_{1:T}$ and $W_{1:M}$) [29]. Furthermore, the multi-step iterative denoising pass must complete across the whole sequence before any single motion frame can be rendered, creating a massive computational delay that makes these models inherently unsuited for live streaming interactions [20].

2.5 Multi-modal Feature Extraction and Integration

To drive a co-speech gesture generation model, the raw features of human communication. Specifically raw audio waveforms and spoken words, must be compressed into structured, conditional embedding vectors. Beyond raw signal extraction, modern architectures require integration mechanisms to inject these diverse modalities into the transformer layers. This section outlines how multi-modal inputs are extracted and integrated to directly alter the underlying motion generation process.

2.5.1 Extraction of Audio and Semantic Encodings

The raw audio waveform of human speech is high-dimensional and contains acoustic noise irrelevant to physical motion. To capture the speech patterns that drive movement, modern models process the raw audio sequence A through self-supervised foundation models such as *Wav2Vec 2.0*[3] or *HuBERT*[12]. These architectures map the continuous audio into a dense, frame-aligned acoustic feature matrix:

$$F_{\text{audio}} = \text{Encoder}_{\text{audio}}(A), \quad F_{\text{audio}} \in \mathbb{R}^{T \times d_a} \tag{2.12}$$

where d_a represents the extracted feature dimension, aligned exactly with the target gesture timeline T. In dyadic conversational settings, this feature extraction expands to encode independent primary speaker and listener channels simultaneously to account for overlaps, interruptions, and back-channeling cues [29].

Concurrently, textual transcripts W are processed to extract high-level conceptual context, which is necessary to trigger sparse, semantically rich gestures (such as pointing or iconic shapes). Bidirectional language models like *BERT*[10] extract contextual word relationships, while contrastive frameworks like *CLIP*[31]] project the text tokens into a shared visual-textual embedding space:

$$F_{\text{text}} = \text{Encoder}_{\text{text}}(W), \quad F_{\text{text}} \in \mathbb{R}^{M \times d_w} \tag{2.13}$$

where d_w is the text feature dimension across word sequence length M.

2.5.2 Local Sequential Integration via Cross-Attention

To align motion with speech timing and rhythm, networks use multi-head cross-attention. This mechanism allows the developing motion features to dynamically query the speech encodings at each frame [38, 27].

The integration is calculated using the cross-attention formula [35]:

$$\text{CrossAttention}(Q, K_c, V_c) = \text{Softmax}\left(\frac{QK_c^T}{\sqrt{d_k}}\right)V_c \tag{2.14}$$

Here, the queries (Q) are a simple linear projection of the model's current gesture timeline, while the keys (K_c) and values (V_c) are linear projections of the local speech context (such as audio F_{audio} or text F_{text}).

Because the resulting attention matrix (QK_c^T) cross-references every gesture frame with every spoken timestamp, it calculates a direct correlation score between joint movements and audio features. This matrix multiplication alters the generation process, synchronizing gesture speed, stroke intensity, and physical pathing directly with specific words or rhythm changes in the spoken text [17, 38].

2.5.3 Global Stylistic Alteration via Feature-wise Linear Modulation (FiLM)

While cross-attention aligns local frames, it is too slow for applying global styles across the whole sequence. Instead, networks use Feature-wise Linear Modulation (FiLM) layers to inject global variables, such as diffusion timesteps t, relationship types f_{rs}, or personality traits f_{ps} [29, 27].

A FiLM layer dynamically alters the internal activation distributions of a transformer block by applying an affine transformation directly to its intermediate feature tensors. Given a global multi-modal context vector $c = [t, f_{rs}, f_{ps}]$, the layer computes feature-wise scaling (γ) and shifting (β) parameters via learnable linear projections [30, 27]:

$$\gamma(c) = W_\gamma c + b_\gamma, \quad \beta(c) = W_\beta c + b_\beta \tag{2.15}$$

These conditioning parameters are then applied channel-wise to the intermediate normalized hidden features $h \in \mathbb{R}^{d_h}$ of the network:

$$\text{FiLM}(h; c) = \gamma(c) \odot h + \beta(c) \tag{2.16}$$

where $\odot$ denotes an element-wise Hadamard product.

Modulating features this way lets FiLM change the gesture style without adding computational complexity. In diffusion, it adjusts the motion based on the noise step t [29, 27], scaling overall expressiveness to match specific identities while keeping the movement aligned with the audio.

Chapter 3

Data Preparation and Pre-processing Pipeline

The Seamless Interaction dataset offers a new approach to co-speech gesture generation by providing dyadic conversational data at a large-scale of 4000+ hours of data.

To accurately model human conversation movement, this framework exclusively makes use of the Naturalistic split of the dataset, intentionally discarding the Improvised actor sequences [2]. This ensures the generative architecture learns spontaneous, genuine co-speech gestures rather than exaggerated scripted performances.

Furthermore while previous datasets like *BEAT2*[16] provide semantic annotations, they lack deep psychological context beyond a basic speaker ID. Human gestures are inherently individual and adapt to the social context of the partner. Therefore, the Seamless Interaction dataset is uniquely viable for this task. It provides continuous interpersonal and relational metadata. Leveraging these rich social features should allow the framework to generate personalized, context-aware gestures that accurately reflect how different individuals physically express themselves depending on who they are speaking to.

Table 3.1: Detailed descriptions of the available data components in the Seamless Interaction dataset.

Component	Description
Video & Audio Data	High-definition 1080p face-to-face footage captured at 30 FPS as MP4 (H.264) files. Audio, separate-channel recordings are provided as 48kHz, 16-bit WAV files.
Body Pose & Keypoints	3D body model parameters (SMPL-H) including global orientation, body/hand pose, and translation parameters provided as 30 FPS NPY files. It also includes bounding boxes alongside detected facial and body keypoints in the COCO-Wholebody[13] format.
Imitator Features	Comprehensive movement data including emotion arousal and valence measurements ranging from -1 to 1. It features categorical emotion scores across 8 categories, 24 dimensions of Facial Action Units (FAUs), and neural encodings for gaze direction and head position.
Annotations Data	Human-annotated behavioral data covering first, and third-party internal states and rationales (1P-IS, 1P-R, 3P-IS, 3P-R) alongside visual annotations (3P-V). It also provides time-aligned speech transcriptions and 100 Hz voice activity detection in JSONL format.

Contributions 1: Multi-Modal Data Curation and Community Accessibility

1. **Reproducible Data Curation Framework:** A pipeline that bridges the gap between raw, un-mined dyadic interaction corpora and downstream gesture-generation modeling.
2. **Multi-Vendor Error Removal Protocol:** An automated filtering system that flags and isolates tracking dropouts and positional occlusions to guarantee upper-body and hand-pose accuracy.
3. **Role-Adaptive Temporal Segmentation:** A dynamic pipeline designed to isolate specific dyadic conversational states (active speaking, listening, or joint interactions), paired with Voice Activity Detection (VAD) padding to capture anticipatory physical wind-up phases.

3.1 The Vulnerability of Raw Conversational Datasets

While the Seamless Interaction dataset [2] provides a completely new foundation for modeling two-way human communication, making use of the raw data corpus introduces engineering and behavioral challenges for generative models. Because the large corpus was compiled across multiple external vendors, separate physical recording locations, and unconstrained environments,

the tracking quality differs largely based on the recording location. Physical constraints for the speakers varied significantly between sessions, some participants were seated while others stood, background noise levels fluctuated, and because the cameras were up high, they often could not see shorter participants well, causing the tracking to cut out when they stepped in. For standard video-based estimation pipelines, these inconsistencies produce unstable SMPL-H parameters, resulting in hand tracking dropouts and artificial joint distortions.

Figure 3.1: Visual examples of the diverse recording environments, lighting conditions, participant postures, and camera angles across different vendors in the dataset, which contribute to tracking inconsistencies.

These random tracking dropouts are especially bad for discrete, token-based gesture generation architectures. In this multi-stage pipeline, the motion representation model (Stage 1 explained in Chapter 4) compresses continuous joint trajectories into a discrete codebook of movement tokens. If raw tracking glitches or collapsed hand orientations get into this compression phase, the codebook tokens become permanently mapped to physically impossible postures. As a result, during the speech to gesture modeling phase (Stage 2 explained in Chapter 5), the auto-regressive transformer treats these structural artifacts as valid targets, learning to predict and replicate tracking errors over time.

On top of these tracking issues, timing is another problem since speakers do not move in perfect alignment with their speech. Instead, natural conversation includes a wind-up phase where a speaker raises their arms and shapes their hands right before making a sound. Relying on raw text transcripts or basic, un-padded VAD tracks cuts out these important preparatory motion windows. This starves the network of the exact signals needed to model the transition from listening to active speaking.

Additionally, modeling context-aware interactions requires separating different conversational roles. In a two-way setup, a person's gestures and posture change completely depending on whether they are speaking, listening, or interrupting. Mixing these states together stops the model from learning the clear differences between an active speaker and an attentive listener.

Since this dataset is so new, no open-source pre-processing tools exist to isolate these roles or filter out tracking errors. Building a dedicated data cleanup and segmentation pipeline is therefore necessary to turn the raw conversational data, into a stable foundation for upper-body motion modeling.

3.2 Algorithmic Curation and Pipeline Implementation

To turn raw multi-vendor recordings into a structured, frame-synchronized dataset, a dedicated data curation pipeline was designed. This pre-processing setup handles spatial noise filtering, multi-modal alignment, and role-based temporal segmentation. By breaking these processes into standalone modules, the pipeline systematically transforms raw audio and motion tracks into clean subsets tailored for the spatial representation and generative modeling tasks.

Experiments 1: Data Preparation and Pre-processing Pipeline

E1.1 Overlapping Window Slicing: Continuous tracking data is split into fixed-size clips using a sliding stride to double-sample motion transitions. This prevents sudden jumps at the sequence boundaries.

E1.2 Out-of-Screen Hand Filtering: Programmatic tracking of COCO-Wholebody 2D coordinate boundaries to drop any motion sequence containing off-screen clipping or glitched hand vertices.

E1.3 High-Fidelity Motion Isolation: Velocity variance is calculated across 42 COCO-Wholebody hand keypoints, applying a local 75th-percentile peak threshold filter. This filters the data down to a clean high-motion only representation dataset.

E1.4 Dyadic Stance Decoupling: Conversational states are sorted over time into active listening, solitary speaking, and overlapping interruptions to separate different body postures.

E1.5 Anticipatory VAD Padding: VAD intervals are expanded on both ends to cleanly capture the silent wind-up phases right before and after physical gesture execution.

E1.6 Lower-Body Kinematic Masking: Isolating and masking the global orientation, root translation, and 8 lower-body joints (indices 0, 1, 3, 4, 6, 7, 9, 10) to lock the legs into a uniform stance, getting rid of video-based tracking noise.

E1.7 Multi-Modal Vector Synchronization: Dense BERT semantic vectors, raw audio waveforms, 1024-dimensional HuBERT/Wav2Vec2 features, and 129-dimensional upper-body poses are aligned over time. This multi-modal data is then saved into an optimized Lightning Memory-Mapped Database (LMDB) binary storage layer.

3.2.1 Data Processing and Sub-System Breakdowns

Overlapping Temporal Windowing and Stride Mechanics

To prepare long conversational streams for sequence modeling, the pipeline cuts the unified timelines into fixed-length gesture clips. Instead of splitting these sequences into isolated, back-to-back blocks, the process uses a sliding stride (such as 10 or 20 frames) to step through the starting points.

Figure 3.2: Visual comparison of temporal segmentation strategies for data preparation. The left shows a continuous 320-frame timeline. The middle demonstrates standard back-to-back partitioning into five sequential, fixed chunks. The right illustrates the proposed approach, generating six overlapping windows. The red arrows indicate the uniform stride offset used to define the start of each subsequent window, which facilitates efficient data augmentation and ensures gesture blending across window boundaries.

For each slice i, the starting and ending frame boundaries are calculated as:

$$\text{Pose}_{\text{start}} = i \cdot \text{Stride} \tag{3.1}$$

$$\text{Pose}_{\text{end}} = \text{Pose}_{\text{start}} + \text{Length}_{\text{window}} \tag{3.2}$$

Using this overlapping stride setup helps solve two main issues found in standard sequence processing:

- **Preserving Timing and Blending Boundaries:** If a model trains only on separate, non-overlapping clips, it never learns how movements transition across the cuts. By overlapping the windows, the process makes sure that every gesture, curve, and movement phase is captured in multiple different training slices. This helps the generative model learn the continuous flow of a gesture, allowing it to smoothly connect sequential outputs during generation without sudden jumps or awkward transitions.
- **Implicit Data Augmentation:** Thoroughly cleaning the data, limits the total hours available for training (filtering the initial dataset down to a clean small subset). The sliding window acts as a temporal data augmentation tool. By shifting

the window slightly forward instead of skipping ahead by the full block length, the pipeline generates many more unique, overlapping training sequences from the same recording files. This gives the network a lot more exposure to complex movement transitions and audio alignments without needing extra raw recording sessions.

Out-of-Screen Hand Filtering Implementation

To stop the stage 1 model 4 from learning broken hand shapes or bad coordinate jumps, the pipeline uses an automatic boundary check. While compiling the timeline, the system evaluates the 2D bounding boxes and keypoint coordinates from the COCO-Wholebody [13] information, present in the Seamless Interaction metadata [2].

The system isolates the hand keypoints across left-hand nodes (indices 91 to 111) and right-hand nodes (indices 112 to 132), tracking 42 keypoints in total. For each frame, the maximum boundary limits (limit_x, limit_y) are pulled from the speaker's bounding box. The pipeline checks the bounds of each joint coordinate:

$$\text{IsOffScreen} = (x \leq 0) \vee (x \geq \text{limit}_x) \vee (y \leq 0) \vee (y \geq \text{limit}_y) \tag{3.3}$$

If a single keypoint goes outside these boundaries, a binary flag marks the frame as off-screen. When cutting the data into fixed-frame blocks, the pipeline checks these flags. If any frame in the block is flagged, the entire fixed-frame sequence is dropped. This setup removes tracking drops and edge-of-frame clipping, keeping corrupted coordinates out of the Stage 1 codebook.

High-Fidelity Motion Isolation

To create a clean dataset for efficient training, the pipeline uses a motion intensity filter. The system isolates 42, 2D hand keypoints and tracks their pixel positions across the timeline. A displacement score is calculated using the L2 norm of consecutive frame positions:

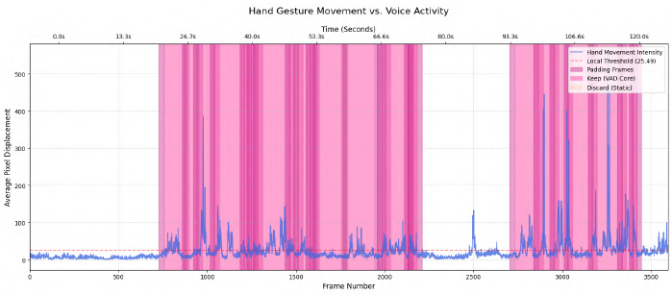
$$\text{Displacement}_t = \|P_t - P_{t-1}\|_2 \tag{3.4}$$

where $P_t \in \mathbb{R}^2$ represents the 2D coordinate vector of a hand keypoint at frame t.

By averaging these displacements across all 42 keypoints, each frame gets a continuous motion intensity score. The pipeline then looks at the full interaction to set a threshold at exactly the 75th percentile of the speaker's total movement.

When filtering for high-intensity segments, the pipeline checks the maximum movement value inside the active VAD window. If this peak value does not go above the 75th-percentile threshold, the entire sequence is dropped. This filter removes flat, unmoving pauses and cuts the training data down to a clean set of high motion data, making sure Stage 1 trains only on clear, expressive movements.

(a) Hand boundary detection



(b) Movement intensity thresholding

Figure 3.3: Overview of spatial and motion-based filtering. (a) **Hand boundary detection:** Evaluates 2D coordinates of the 42 COCO-Wholebody hand keypoints to identify and strictly remove 64-frame sequences where a speaker's hands exit the camera boundaries. (b) **Movement intensity thresholding:** Tracks the average keypoint displacement over time against a local 75th-percentile threshold (red dashed line). The pipeline retains segments that simultaneously feature high-intensity motion during active voice activity (pink shaded regions), safely removing static frames (white regions).

Decoupling Dyadic Conversational Stances

To model natural interaction context, the pipeline uses a role-based categorization setup, to split continuous tracking streams into distinct conversational states. This part processes raw VAD intervals from both the speaker and their partner to build a synchronized frame-level state tracker. Every frame across the timeline is put into one of three interaction states:

- **State 0 (Active Listening)**: The main speaker is quiet while the partner's VAD track shows active speech.
- **State 1 (Solitary Speaking)**: The main speaker is talking while the partner's VAD track is completely silent.
- **State 2 (Overlapping/Interruption)**: Both the main speaker and the partner have active VAD speech flags at the same time.

The pipeline uses a configuration parameter to filter the data based on these states. If it is set to solitary speaking, the system skips any data blocks where speech flags are missing. On the other hand, if listening mode is chosen, the pipeline drops active speech blocks and keeps conversational listening sequences.

To keep the dialogue behavior realistic, the listening filter handles short back-channels automatically. If a speech segment is less than 30 frames long (1.0 second of sound) and has no high-intensity movement flags, the pipeline re-classifies it as a listening back-channel, instead of an interruption. This clear separation makes it easy to isolate specific turn-taking mechanics, giving the downstream models distinct boundaries between speaking and listening roles.

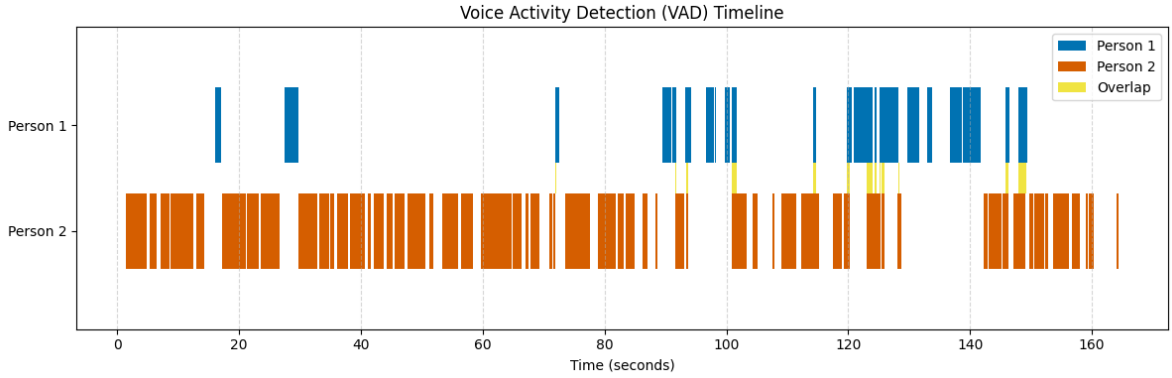


Figure 3.4: Timeline visualization of Voice Activity Detection (VAD) feature mapping. The system tracks active speech states for both conversational partners, including solitary speaking segments, passive listening windows, and overlapping speech/interruption regions. This timeline is used to construct the discrete conversational state features.

Anticipatory VAD Padding Mechanics

To fix the timing gap between speech and movement, the pipeline adds extra frames to both ends of the VAD intervals. Standard speech tracks usually cut out the initial movements a speaker makes before sound is produced. To capture these preparation movements, the pipeline takes a padding parameter and converts it into a frame count based on the 30 fps capture rate.

For every isolated VAD interval, the pipeline expands the boundaries outward to update the start and end frame indices:

$$\text{Start}_{\text{padded}} = \max(0, \text{Start}_{\text{raw}} - \text{Padding}_{\text{frames}}) \tag{3.5}$$

$$\text{End}_{\text{padded}} = \min(\text{Frames}_{\text{total}}, \text{End}_{\text{raw}} + \text{Padding}_{\text{frames}}) \tag{3.6}$$

This tracking makes sure the timeline saves the preparatory "wind-up" phase, where a speaker raises their arms or shifts their posture right before they start talking.

Lower-Body Kinematic Masking and Upper-Body Isolation

To handle tracking noise in full-body estimations, often caused by setup differences like participants sitting in chairs versus standing. The pipeline uses a kinematic masking step. Video-based pose estimators struggle to track legs when they are hidden or blocked, causing drift that can corrupt the overall body alignment and root stability. Since the generation task focuses on upper-body gestures, the pipeline isolates the active joints and removes lower-body movement.

The system targets a specific joint mask for the lower body:

$$\mathcal{M}_{\text{lower}} = [0, 1, 3, 4, 6, 7, 9, 10] \tag{3.7}$$

where these indices map directly to the pelvis, thighs, knees, and feet of the standard 21-joint SMPL-H body model. The pipeline overwrites these lower-body indices with zeros. At the same time, the global orientation and root translation arrays are zeroed out to lock the model into a stable, uniform stance.

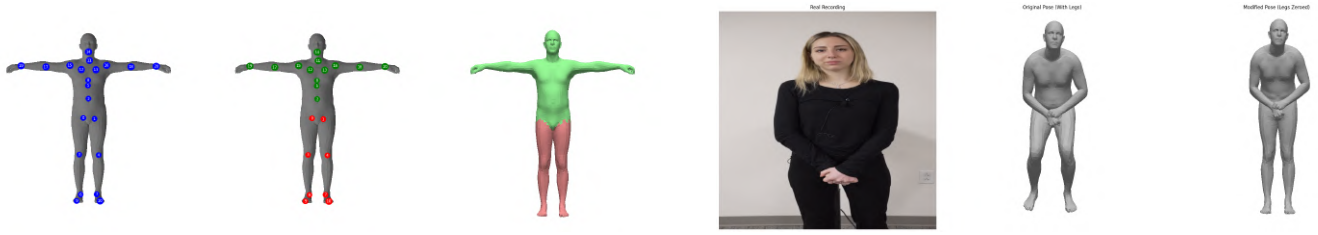
To build a clean joint vector for training, the system extracts the 13 active upper-body joints using a keep-indices mapping:

$$\mathcal{I}_{\text{keep}} = \{i \in \mathbb{Z} \mid 0 \leq i < 21 \text{ and } i \notin \mathcal{M}_{\text{lower}}\} \tag{3.8}$$

These 13 upper-body joints (39 dimensions, denoted as $\mathbf{J}_{\text{body}} \in \mathbb{R}^{39}$) are combined with the left-hand parameters (15 joints, 45 dimensions, denoted as $\mathbf{J}_{\text{left}} \in \mathbb{R}^{45}$) and right-hand parameters (15 joints, 45 dimensions, denoted as $\mathbf{J}_{\text{right}} \in \mathbb{R}^{45}$). This results in a flat 129-dimensional continuous feature vector representing the isolated upper-body posture:

$$\mathbf{V}_{\text{pose}} = [\mathbf{J}_{\text{body}} \,||\, \mathbf{J}_{\text{left}} \,||\, \mathbf{J}_{\text{right}}] \in \mathbb{R}^{129} \tag{3.9}$$

This 129-dimensional vector serves as the clean, motion-isolated representation used directly for training the generative models.



(a) SMPL-H joint index selection

(b) Lower-body zeroing effect

Figure 3.5: Upper-body kinematic isolation applied to mitigate video-based estimation noise within the SMPL-H pose data. (a) **SMPL-H joint index selection:** Illustrates the structural division of the body model, highlighting the retained upper-body joints (green) and the eight lower-body joints (red) targeted for zero-masking. (b) **Lower-body zeroing effect:** Demonstrates the practical application of this masking strategy by comparing the raw estimated pose containing leg tracking noise (left) against the final stabilized pose (right), where lower-body joints, global orientation, and root translation are zeroed out to obtain a clean 129-dimensional feature vector.

Timeline Alignment and LMDB Serialization

Once the filtering and padding steps are done, a synchronization loop aligns all the different channels into a single, frame-accurate array. The script resamples the raw audio, then runs it through pretrained models to get 1024-dimensional *HuBERT*[12] and *Wav2Vec2*[3] features at 50 Hz. To match these 50 Hz audio embeddings with the 30 fps motion frames, the pipeline finds a shared duration based on the shortest common length among the pose, audio, and transcript files.

For text alignment, the system parses individual words to get their start and end timestamps. It maps these timestamps to the 30 fps motion frames, inserting a 768-dimensional semantic vector into the specific frames where each word is spoken. Any frames without active words are filled with zeros.

At the same time, the SMPL-H pose data goes through the upper-body isolation step. This returns the clean 129-dimensional pose vector per frame, removing lower-body tracking noise.

To handle the massive dataset without running into memory issues, the pipeline processes the files in parallel. It packages the aligned data, including the poses, raw audio, sound embeddings, text vectors, and personality traits, and saves them directly into an optimized database. This setup avoids standard reading delays, allowing the system to load the multi-modal training data instantly.

3.3 Pipeline Configurations and Data Results

To show the pipeline's practical utility, this section outlines the user-configurable parameters, the standardized structure of its saved outputs, and the resulting data from different filtering options. By offering a flexible configuration setup, the pipeline can be customized to generate specialized datasets for either spatial representation learning or long-form dyadic sequence modeling.

3.3.1 Pipeline Customization and Serialization Schema

The data preparation pipeline is controlled via a centralized configuration file, allowing the user to programmatically manipulate the spatial filters, temporal windowing parameters, and conversational stance segmentations. The complete set of user-configurable parameters is detailed in Table 3.2.

Table 3.2: User-Configurable Pipeline Parameters

Parameter	Default	Description
`pose_fps`	30	Target motion frame rate.
`audio_sr` / `audio_hz`	16000 / 50	Audio sample rate and acoustic feature frequency.
`pose_length` / `stride`	64 / 20	Clip window frame size and step interval.
`segment_mode`	"speaking"	Conversational stance filter (`both`, `listening`, `speaking`).
`filter_out_of_screen_hands`	`false`	Discards clips with out-of-bounds hand tracking flags.
`keep_only_high_intensity_blocks`	`false`	Filters listener segments to high-movement sequences only.
`vad_padding_in_seconds`	1.0	Bilateral temporal expansion around active speech tracks.
`include_stage_2_partner_data`	`false`	Toggles dyadic co-speech track extraction.
`personality_detail_mandatory`	`true`	Enforces strict metadata validation rules.

By adjusting these parameters, researchers can easily isolate distinct behavioral environments. For instance, disabling hand filtering and enabling dyadic partner tracking generates a, continuous dataset ideal for modeling interactive conversational rhythms. Conversely, enforcing strict hand boundary validation and high-motion thresholding isolates a noise-free subset designed to train stable, Stage 1 models.

After processing, filtering, and synchronization, the aligned multi-modal features are serialized into binary structures and stored within a LMDB. This process packages all acoustic, semantic, and kinematic variables into a single, frame-accurate entry, structured as shown in Table 3.3.

Table 3.3: Serialized LMDB Entry Structure

Field Key	Data Type / Dimensions
`pose`	129-dimensional continuous upper-body posture vector array
`raw_audio`	Resampled raw acoustic waveform array segment
`wav2vec_audio` / `hubert_audio`	1024-dimensional deep acoustic embeddings
`bert`	768-dimensional tokenized word semantic vectors
`semantic_scores`	Frame-level 5-tier classification profile (0.0 to 1.0)
`emotions` / `conversation_states`	Categorical frame-level interaction indices
`sample_word` / `sample_sentence`	Text transcripts aligned to corresponding frames
`personality_traits` / `social_relationship`	Normalized metadata vectors and relationship mapping IDs
`segment_index` / `seamless_id` / `file_path`	Structural tracking indices and reference origin strings
`partner_*` (Conditional)	Parallel co-presence tracks (included if stage 2 data is enabled)

This shared setup ensures that downstream model architectures can immediately make use of a synchronized, self-contained multi-modal state vector at every training step, completely bypassing the need for complex runtime data alignment.

3.3.2 Dataset Filtering and final results

To demonstrate the progressive impact of each filtering layer, Table 3.4 charts the step-by-step reduction of the dataset from its raw, unrefined form down to the optimized subsets. This structured breakdown highlights how the pipeline eliminates spatial noise and idle intervals, resulting in a dense training set.

Table 3.4: Dataset Results and Filtering Progression from Raw to Cleaned Subsets

Curation Stage	Applied Filters / Constraints	Amount (Hours)	Retained (%)
Raw Naturalistic Pool	None (Raw multi-vendor recordings containing idle silence, speaker and listener clips, and tracking dropouts).	22.5	100.0%
Speaker-Only Subset	Basic Voice Activity Detection (VAD) alignment to isolate active speaking frames.	17.4	77.3%
Fully Cleaned Subset	Speaker-only frames + 2D out-of-screen hand filtering + 75th-percentile motion intensity thresholding (Used for Stage 1).	13.1	58.2%

Progression Analysis and Downstream Allocation

As shown in Table 3.4, the initial raw pool contains various non-expressive or corrupted frames. Applying VAD segmentation cuts the corpus to 17.4 hours, stripping out passive silence and long dialogue pauses.

The final curation phase contains the 13.1 hour fully cleaned subset. By enforcing strict 2D keypoint boundary checks alongside the 75th-percentile motion threshold, the pipeline prunes 24.7% of the speaker-only segments, leaving only active, expressive upper-body gestures.

This 13.1 hour corpus is used for Stage 1 training, preventing the spatial RVQ-VAE from wasting capacity on idle postures. Conversely, Stage 2 utilizes a much larger raw subset of the dataset, detailed in Chapter 5, to preserve continuous conversational transitions, listening behaviors, and social metadata.

3.4 Chapter Discussion & Conclusion

In this chapter, we presented a comprehensive, reproducible data curation and pre-processing pipeline designed to bridge the gap between raw, unrefined dyadic interaction corpora and downstream gesture-generation tasks. By transforming the volatile, multi-vendor Seamless Interaction dataset, into structured, frame-synchronized subsets, the pipeline resolves the challenges of spatial tracking noise, out-of-frame occlusions, and temporal misalignment.

The core strength of this pre-processing framework lies in its dual-stage optimization strategy, which directly solves the conflicting data needs of both modeling phases. For Stage 1 (4), the pipeline prioritizes clean geometric poses over continuous sequence length. By combining strict 2D out-of-screen hand filtering with the 75th-percentile motion intensity threshold, the raw dataset is distilled into a 13.1 hour high-fidelity subset. This filtering ensures that the vector quantization layers in Stage 1 train exclusively on clear, expressive movements, preventing the codebook from collapsing onto motionless resting poses or tracking noise.

For Stage 2 (5), the pipeline shifts its focus to preserving continuous interactive context, turn-taking dynamics, and multi-modal alignment. By using role-based stance decoupling and bi-directional VAD padding, the framework captures the natural "wind-up" phases of gestures alongside aligned sound, text, and social metadata. Saving these multi-modal streams into a unified, LMDB database completely avoids data-loading issues, keeping training loops efficient.

Ultimately, this curation pipeline establishes a clean, robust data foundation that directly enables the gesture generation models detailed in the following chapters. Furthermore, because the Seamless Interaction dataset is recent, no standardized, open-source pre-processing utility previously existed for it. By releasing this pipeline publicly, this work provides a distinct contribution to the research community, offering an immediate, structured foundation for any future work utilizing this dataset.

Chapter 4

Upper-Body Motion Modeling via Hierarchical RVQ-VAE

To generate realistic co-speech gestures, Continuous and high-dimensional physical movements must first be compressed into a structured, manageable representation. This chapter introduces the Stage 1 framework: a unified upper-body RVQ-VAE. By mapping continuous joint rotations into a discrete sequence of motion primitives, this model establishes a compact discrete vocabulary. This vocabulary serves as the essential foundation for the Stage 2 model presented in Chapter 5.

Contributions 2: A Combined Upper-Body Representation and Discrete Kinematic Quantization

1. **Custom Stage 1 Motion Reconstruction:** Designing and training a self-contained reconstruction model from scratch, overcoming the SMPL-X format incompatibility of existing benchmarks to establish a robust SMPL-H representation.
2. **Consolidated Upper-Body Architecture:** A single Stage 1 model that represents the torso, arms, and hands together, avoiding the training complexity and synchronization issues of separate sub-networks.
3. **Multi-State Generalization Evaluation:** A systematic assessment of the representation's generalization capacity across both active speaking and passive listening states using a compressed, high-motion 13.1 hour training subset.

4.1 The Problem With Previous Stage 1 Models and Motivation for Custom Reconstruction

Since the Seamless Interaction dataset is a newly introduced corpus, no pre-trained Stage 1 reconstruction models or checkpoints are available to build upon. In typical gesture-generation pipelines, researchers can re-use existing Stage 1 models to save training time. However, almost all current SOTA frameworks use *BEAT2*[16] as their main reference benchmark, which provides motion data in the SMPL-X format. In contrast, the Seamless Interaction dataset is formatted using the SMPL-H model, creating a direct compatibility issue that prevents the re-use of pre-trained checkpoints from other works.

To overcome this format limitation, a newly designed custom Stage 1 network had to be trained from scratch. This challenge provided the opportunity to make targeted design choices for the approach. While traditional frameworks often train separate, complex Stage 1 models for different body regions (such as the head, arms, and hands individually), this work's approach chose to model the entire upper body as a single, unified representation [20, 38]. By representing the torso, arms, and hands together in one unified codebook, it avoids the heavy synchronization issues and training overhead of multi-network setups.

Additionally, because the complete dataset is too large to train on, the concentrated 13.1 hour high-motion subset curated in Section 3.3.2 was used. This choice introduced an important question: *would a Stage 1 model trained exclusively on active motion sequences be able to generalize to the quiet, subtle states of a listener?* To succeed, the reconstruction model must perform robustly in both conversational modes, accurately reconstruct silent listening poses as well as expressive speaking gestures. Therefore, the experiments are designed to test both states, ensuring the learned vocabulary is truly representative of natural human interaction.

4.1.1 Hierarchical RVQ-VAE Methodology and System Architecture

This section explains the implementation of the first stage of the generative framework. To properly achieve the set goal of building a robust, high-fidelity reconstruction space, three core experiments are introduced:

Experiments 2: Stage 1 Upper-Body Joint Reconstruction and Generalization

E2.1 Unified Upper-Body Joint Integration: Testing if a single, unified RVQ-VAE network can stably reconstruct both coarse arm trajectories and detailed, 30-joint finger movements together without needing separate, complex sub-networks.

E2.2 High-Motion Curation Generalization: Evaluating whether training the model exclusively on the concentrated 13.1 hour high-motion subset allows the network to learn robust movement primitives and converge efficiently compared to using a massive uncleaned corpus.

E2.3 Cross-Stance Reconstruction Generalization: Testing the codebook's ability to reconstruct both active, expressive speaking gestures and passive, low-intensity listening postures to ensure the model generalizes across all conversational phases.

By performing these targeted experiments, the pipeline systematically resolves the tracking inconsistencies and representation gaps found in raw multi-vendor datasets. This process establishes a stable Stage 1 model, which is broken down in detail in the following subsections.

Spatial and Kinematic Compression

The foundational neural blocks of the encoder-decoder architecture originate from the *Photo2Real* [27] framework and its dyadic extension *DyaDit* [29]. However, the model configuration, optimization strategy, and training pipeline were created specifically for this work. Using the previously introduced custom pre-processing pipeline, the model was trained on the 13.1 hour subset of expressive upper-body motion. This specifically selected subset of the data allows the model to generalize effectively using only a fraction of the data typically required by traditional datasets like *BEAT2*[16], which generally demand 20 to 30 hours of motion to achieve comparable accuracy.

Before the model compresses the motion into a smaller subspace, the raw kinematic SMPL-H inputs undergo an important transformation to ensure continuous network gradients during training. The initial motion data is extracted as a 129-dimensional vector, representing 43 joints consisting of 13 upper-body joints and 30 finger joints, stored in 3D axis-angle format.

Formally, let the input sequence be $V = \{v_1, v_2, \dots, v_T\} \in \mathbb{R}^{T\times 129}$, where each joint rotation $r_{t,j} \in \mathbb{R}^3$ (for joint j at frame t) is represented in axis-angle format. During the forward pass, these axis-angle representations are converted into orthonormal rotation matrices $R_{t,j} \in \mathrm{SO}(3)$ and subsequently flattened into a continuous 6D rotation representation $f_{t,j} \in \mathbb{R}^6$ by extracting their first two column vectors. Concatenating these projections across all 43 joints results in a continuous 258-dimensional 6D rotation feature vector representing the full upper-body state:

$$x_t = [f_{t,1}^T \,||\, f_{t,2}^T \,||\, \dots \,||\, f_{t,43}^T]^T \in \mathbb{R}^{258} \tag{4.1}$$

The resulting sequence $X = \{x_1, x_2, \dots, x_T\} \in \mathbb{R}^{T\times 258}$ provides a numerically stable input topology for the convolutional layers of the RVQ-VAE.

The spatial and temporal compression is performed through a sequence of 1D temporal convolutions making use of a predefined down-sampling factor of $f = 4$. The encoder processes overlapping windows of $T = 64$ motion frames at 30 FPS, with a stride of 20 frames. These 64-frame sequences are then compressed into $T' = 16$ latent steps. Simultaneously, the spatial channel width is compressed into a $d = 64$-dimensional embedding space, which significantly shortens the sequences for later stages while preserving the dynamic features of the original upper-body movement.

Hierarchical Discrete Mapping

After the spatial and temporal compression, the continuous 64-dimensional latent representations $Z \in \mathbb{R}^{16\times 64}$ are converted into discrete tokens using the Residual Vector Quantization (RVQ) module. The network maps each of the 16 temporal steps to the shared codebook $\mathcal{C} = \{e_1, e_2, \dots, e_{512}\} \in \mathbb{R}^{512\times 64}$ introduced in the background section, maintaining a fixed vocabulary size of 512 entries. To clean up both large arm movements and finger details, the RVQ mechanism relies on a residual depth of $D = 4$ quantization layers.

Mathematically, at each compressed temporal step $t' \in \{1, \dots, 16\}$, the continuous encoder output vector $z_{t'} \in \mathbb{R}^{64}$ is quantized hierarchically across the layers. Setting the initial 0-indexed residual as $r_{t',0} = z_{t'}$, the loop recursively computes the code selection index $k_{t',d}$ and the subsequent remaining residual error $r_{t',d}$ for each depth layer $d \in \{1, 2, 3, 4\}$ using the shared vocabulary [14]:

$$k_{t',d} = \arg \min_{i\in\{1,\dots,512\}} \|r_{t',d-1} - e_i\|_2^2 \tag{4.2}$$

$$r_{t',d} = r_{t',d-1} - e_{k_{t',d}} \tag{4.3}$$

The final quantized latent vector $\hat{z}_{t'}$ is reconstructed as the direct sum of the selected codebook vectors across all 4 layers [14]:

$$\hat{z}_{t'} = \sum_{d=1}^{4} e_{k_{t',d}} \in \mathbb{R}^{64} \tag{4.4}$$

This hierarchical mapping ensures that the first layer captures broad structural postures, while subsequent layers iteratively refine the representation by quantizing residual tracking errors to inject fine motion detail. This process produces exactly 4 discrete code indices per temporal step, effectively compressing the 64-frame raw motion clip into a compact sequence of 64 discrete tokens (16 steps $\times$ 4 layers) that balances high-fidelity reconstruction with a compact, symbolic action space for the Stage 2 generative model.

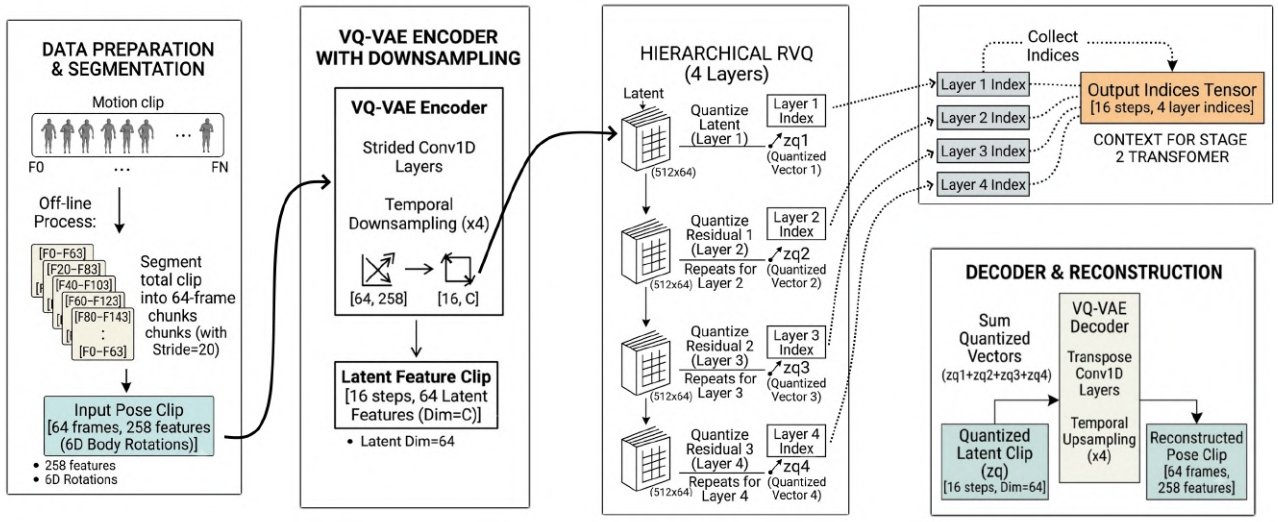


Figure 4.1: Architecture of the Stage 1 RVQ-VAE framework for upper-body motion compression. First, the data preparation pipeline segments raw motion into 64-frame chunks, formatted as 258-dimensional 6D body rotations. The VQ-VAE encoder then applies strided 1D convolutions to compress the data temporally by a factor of four, producing a 64-dimensional latent feature representation over 16 steps. Next, a four-layer Hierarchical RVQ module iteratively quantizes the continuous latents and their residuals into discrete tokens using a 512-dimensional codebook. These collected indices serve as context for the Stage 2 transformer, while the decoder uses the summed quantized vectors and temporal upsampling to reconstruct the original pose clip.

Multi-Objective Optimization and Training Configuration

To ensure the reconstructed gestures maintain motion accuracy, spatial stability, and temporal smoothness, the RVQ-VAE was optimized using a custom joint multi-objective loss function, inspired by the *SemTalk* architecture. The total Stage 1 objective balances the following competing losses:

$$L_{\text{stage1}} = \lambda_{\text{rec}} L_{\text{rec}} + \lambda_{\text{vel}} L_{\text{vel}} + \lambda_{\text{acc}} L_{\text{acc}} + \lambda_{\text{ver}} L_{\text{ver}} + \lambda_{\text{commit}} L_{\text{commit}} \tag{4.5}$$

where each λ represents the hyperparameter weighting for its respective loss component, all of which were set to 1.0 during training.

Kinematic and Temporal Losses The primary loss function is a reconstruction Mean-Squared Error (MSE) loss applied directly to the 6D rotation features. This calculates the error between the ground-truth 6D rotation pose vectors $x_t \in \mathbb{R}^{258}$ and the network's reconstructed pose vectors $\hat{x}_t \in \mathbb{R}^{258}$ at frame t:

$$L_{\text{rec}} = \frac{1}{T} \sum_{t=1}^{T} \|\hat{x}_t - x_t\|_2^2 \tag{4.6}$$

To heavily punish high-frequency jitter and temporal gaps, the loss is paired with L_1 Velocity and L_1 Acceleration losses calculated across consecutive frames:

$$L_{\text{vel}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \| (\hat{x}_{t+1} - \hat{x}_t) - (x_{t+1} - x_t) \|_1 \tag{4.7}$$

$$L_{\text{acc}} = \frac{1}{T-2} \sum_{t=1}^{T-2} \| (\hat{x}_{t+2} + \hat{x}_t - 2\hat{x}_{t+1}) - (x_{t+2} + x_t - 2x_{t+1}) \|_1 \tag{4.8}$$

Differentiable Vertex Loss Furthermore, a vertex loss has been included in the training process. By converting the reconstructed joint representations back into 3D surface coordinates using a frozen SMPL-H body layer, positional MSE, velocity, and acceleration losses were calculated directly on the mesh vertices. The spatial vertex loss applies a spatial MSE between the predicted ($\hat{V}$) and ground-truth (V) SMPL-H vertices:

$$L_{\text{ver}} = \frac{1}{T \cdot V_{\text{count}}} \sum_{t=1}^{T} \|\hat{V}_t - V_t\|_2^2 \tag{4.9}$$

In the training pipeline, positional vertex loss is directly augmented with temporal velocity and acceleration constraints on the vertex coordinates to preserve synchronized surface trajectories.

Codebook Commitment Loss To ensure the continuous encoder embeddings map cleanly to the shared codebook vectors, a multi-layer Residual Vector Quantization commitment loss L_{commit} is applied. Calculated across all 16 temporal steps and 4 depth layers, this penalty forces the continuous features $z_{t'}$ to adhere tightly to the progressively refined partial code sums $\hat{z}_{t'}^{(d)}$ [14]:

$$L_{\text{commit}} = \frac{1}{16} \sum_{t'=1}^{16} \sum_{d=1}^{4} \|z_{t'} - \text{sg}[\hat{z}_{t'}^{(d)}]\|_2^2 \tag{4.10}$$

where sg[·] denotes the stop-gradient operation. This penalty stabilizes training by keeping encoder outputs aligned with the discrete codebook manifold [34]. To prevent codebook collapse where the model maps everything to a single pose, we use a dead-code expiration mechanism to refresh unused clusters during training [29].

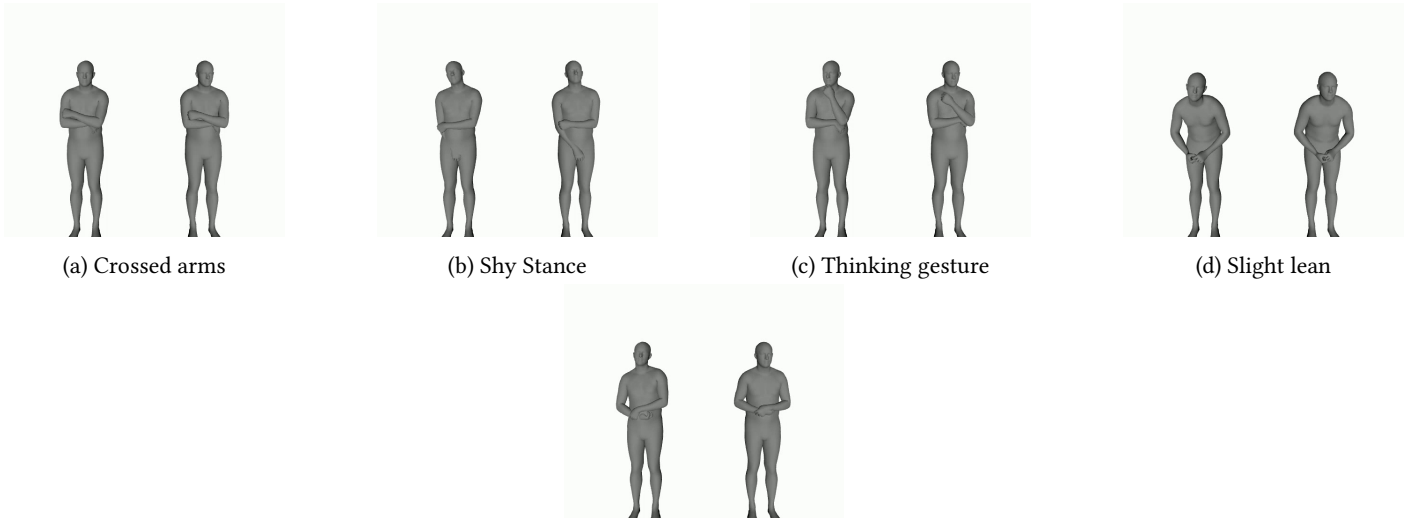
Training Configuration The model was trained for 14 hours, from scratch in a Distributed Data Parallel (DDP) environment across 4 A100 GPUs to maximize computational throughput. The network was optimized using Adam with a base learning rate of 0.0012, processing global batch sizes of 64 over a span of 800 epochs. To ensure stable gradient convergence and prevent exploding parameters, gradient clipping was enforced at a norm of 1.0. Additionally, a ReduceLROnPlateau scheduler configured with a decay factor of 0.75 and a patience of 50 epochs dynamically adjusted the learning rate based on the validation loss trajectory.

4.1.2 Qualitative Reconstruction Evaluation

While quantitative metrics provide a broad statistical view of network performance, evaluating generative physical movement relies heavily on visual perception. Quantitative values such as Mean Squared Error (MSE) can sometimes be misleading. For example, a model that reconstructs a completely frozen, static posture might obtain a low coordinate error, but it fails to capture natural movement. To thoroughly evaluate the visual quality of the Stage 1 model, a qualitative analysis was conducted using the rendering pipeline proposed in Chapter 6, focusing on three scenarios: passive listening poses, tracking artifact correction, and overall reconstruction capability.

Passive Listening Pose Reconstruction

Given that the Stage 1 model was trained on the 13.1 hour high-motion speaker subset, a main concern was whether the network could generalize to reconstruct quiet, passive listening states. Listening postures typically feature subtle, low-intensity movements. For example crossed arms, hands resting on the lap, or slight head tilts, which were heavily filtered out of the training corpus.



(a) Crossed arms (b) Shy Stance (c) Thinking gesture (d) Slight lean

(e) Passive stance

Figure 4.2: Reconstruction of passive listening postures. In each sequence, the ground truth frame containing low-intensity movement (left) is compared directly against the model's reconstruction (right). Despite being trained strictly on active speaking gestures, the model accurately reproduces stable, natural resting postures without collapsing or generating artificial jitter.

As visualized in Figure 4.2, the model handles these passive states remarkably well. The codebook is capable of representing resting and crossed-arm postures accurately. In several test sequences, the reconstructed pose is visually cleaner and more anatomically stable than the ground-truth tracking, which often suffers from minor coordinate drift when the participant remains still. This demonstrates that the high-motion training subset successfully taught the model the fundamental structural relationships of the joints, allowing it to generalize smoothly to unseen static positions.

Robustness and Denoising of Tracking Artifacts

A challenge when working with raw multi-vendor datasets is the presence of sudden tracking glitches, hand dropouts, and joint distortion artifacts. Because the pre-processing pipeline filtered out these corrupt frames, the Stage 1 model trained exclusively on physically plausible, clean postures. Therefore, to see the effects of this, an evaluation was conducted to see how the model reacts when presented with noisy, glitched ground-truth sequences during inference.

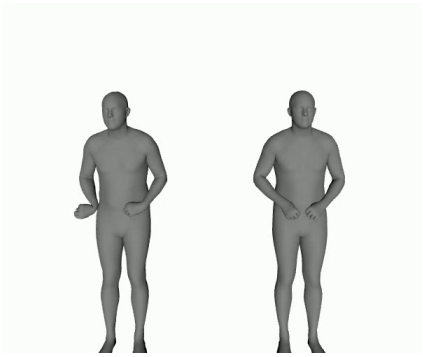


Figure 4.3: Visual artifact correction. The ground-truth sequence (left) contains a hand tracking dropout where the hand collapses and glitches out of its natural coordinates. The proposed Stage 1 reconstruction (right) successfully filters out this noise, mapping the sequence to a stable, biologically plausible joint representation.

As shown in Figure 4.3, when a hand coordinate "glitches" or collapses in the ground truth, the reconstruction remains stable, smooth, and physically controlled. Because the codebook possesses no discrete tokens representing impossible, broken joint configurations, the quantizer behaves as a natural denoising prior. It forces the corrupted input coordinates onto the nearest valid upper-body postural token in the codebook. This effectively prevents tracking anomalies from leaking into the generative loop of Stage 2.

Overall Reconstruction Capabilities

To assess the model's overall generalization capabilities across different speakers and conversational environments, a visualization was made to display several continuous sequences from the unseen test set.

Figure 4.4: Continuous reconstruction capabilities across five distinct test sequences. Each color represents a unique participant/interaction. For each color-coded pair, the top row represents the sequential ground-truth frames, while the bottom row represents the corresponding Stage 1 model reconstructions.

Figure 4.4 illustrates these comparisons across five distinct colored sequences. The top row of each colored pair shows the ground truth, and the bottom row shows the resulted reconstruction. The model consistently captures the flow and dynamics of the upper-body movements. While the observation could be made that the head rotation could be slightly improved and remains a bit stiff in some transitions, the overall arm trajectories and detailed finger poses are reproduced exceptionally well.

This success validates the core training strategy. By curating a high-motion 13.1 hour subset, the framework bypassed the need to train on tens of hours of redundant, static data. This approach reduces training times for research in this field. Furthermore, it ensures that every code in the discrete codebook represents an active, expressive pose, preventing codebook collapse and eliminating redundant "silent" tokens.

4.2 Chapter Discussion & Conclusion

In this chapter, the Stage 1 generative framework was presented: a unified upper-body RVQ-VAE designed to compress continuous, high-dimensional kinematic motion into discrete representation tokens. Forced to build a reconstruction network from scratch due to the SMPL-H format compatibility issues of the new Seamless Interaction dataset, this design shift was used to implement a streamlined and integrated system. By modeling the torso, arms, and hands together in a single codebook, the framework successfully avoided the synchronization complexity and training overhead of multi-network pipelines.

The qualitative evaluation of the Stage 1 model contains three key insights:

- **Stance Generalization:** Despite training strictly on the high-motion speaking subset, the codebook generalized exceptionally well to passive listening postures, capturing stable resting and crossed-arm poses with high geometric fidelity.
- **Postural Regularization:** The discrete quantizer behaves as a natural denoising prior. When presented with tracking glitches or hand coordinate dropouts during inference, the model reconstructs physically plausible joints, effectively preventing tracking noise from leaking into downstream stages.

- **Representation Efficiency:** By focusing training on a concentrated 13.1 hour high-motion subset, the framework reduced computational training overhead while ensuring that the codebook tokens remain mapped to active, expressive movements rather than redundant static poses.

In conclusion, the Stage 1 model approach successfully establishes a clean, stable, and expressive symbolic vocabulary of upper-body gesture primitives. By eliminating tracking noise and organizing physical movement into a compact latent sequence, this framework provides the essential structural foundation required for the Stage 2 model to learn the complex relationships of co-speech gesture generation.

Chapter 5

Dyadic Context-Aware Auto-regressive Transformer

With the high-dimensional upper-body motion successfully compressed into a discrete, hierarchical codebook in Stage 1, the generative task can be reframed as a sequence modeling problem. This chapter introduces the Stage 2 framework: a dyadic, context-aware, decoder-only causal transformer. The primary goal is to bridge the gap between offline gesture generation and real-time interactive systems. By making use of the structured discrete vocabulary established by the Stage 1 model, a network is designed that is capable of generating continuous, co-speech gestures. To capture the natural, uninterrupted temporal dynamics of human conversation, including active speaking, silent listening, and overlapping turn-taking phases, the Stage 2 model is trained on a complete, 150-hour unfiltered training set.

Contributions 3: Causal Sequence Modeling and a Setup Built for Real-Time Streaming

1. **Causal Auto-Regressive Framework:** Shifting gesture generation away from static, offline clip generation toward an auto-regressive, live-streaming framework by treating discrete motion primitives as a linguistic vocabulary within a causal transformer.
2. **Large-Scale Optimization from Scratch:** Leveraging the scale of the new dyadic Seamless Interaction dataset to optimize a data-hungry, decoder-only sequence model from scratch, bypassing traditional data scarcity limitations.
3. **Progressive Structural Foundation:** Establishing a clear developmental narrative that addresses core streaming and convergence hurdles first, laying a stable architectural foundation before analyzing multi-modal context variables in downstream ablation studies.

5.1 The Limitations of Existing Gesture Generation Pipelines

To understand why the Stage 2 architecture is necessary, it is first important to analyze the technical limitations holding back current SOTA gesture generation models. Traditional approaches, predominantly trained on monadic benchmarks like the *BEAT2*[16] dataset, suffer from a conceptual oversight: they treat co-speech gestures as a static, one-to-one translation of the speaker's audio. However in real-world dyadic communication, human movement is adaptive. Gestures are constantly shaped, interrupted, and re-framed by the behavioral cues and presence of the conversational partner. Modeling speech-to-motion without acknowledging this two-way relationship results in robotic, unresponsive behaviors that fail to capture natural conversational dynamics [11, 5].

Beyond this interaction gap, an engineering issue remains: the lack of streaming capabilities in existing pipelines. Traditional architectures typically generate gestures by processing an entire audio and text sequence simultaneously. Since these models rely on non-causal operations that require future acoustic context, they cannot generate motion frame-by-frame. This design renders them entirely unusable for live, interactive virtual agents where real-time, low-latency responsiveness is mandatory.

This specific limitation is what directly inspired this works choice of architecture. Given that the motion codes from the Stage 1 model are discrete, it became clear that they naturally function like an alphabetic vocabulary. This similarity suggested a logical path, gesture generation could be treated exactly like natural language processing, using an auto-regressive, decoder-only Large Language Model (LLM) style approach to predict the next motion token frame-by-frame. Although the interaction features of the Seamless Interaction dataset are very interesting, the absolute lack of streaming functionality in current frameworks felt like a critical gap that had to be addressed first. This led to a change in approach, before adding complicated context, the first step is showing that a basic transformer could reliably generate stable movement in real time.

In earlier approaches, this LLM-style architecture has remained untested in the gesture generation field, primarily because decoder-only transformers are data-hungry [1]. Without a large volume of training data, causal models quickly suffer from error accumulation and auto-regressive collapse. Because older datasets only provided thirty to forty hours of captured motion, attempting to train such an architecture was practically impossible. By making use of the newly introduced scale of the Seamless Interaction dataset, the massive data volume required to make this decoder-only sequence model converge is finally available. This shift allowed this work to validate a brand-new streaming approach, establishing a stable backbone before systematically exploring context-aware variables through downstream ablation studies.

5.1.1 Methodology Overview and Experimental Design

In this section, the transition is made from traditional, offline clip-based generation to an auto-regressive, frame-by-frame streaming approach. By treating the discrete motion indices as linguistic tokens, a lightweight, decoder-only causal transformer can be leveraged to generate gestures in real-time, a must for real-time interactive virtual agents.

While the ultimate objective is context-aware dyadic interaction, the methodology was structured to prioritize stabilizing this highly experimental auto-regressive foundation first. To evaluate the architectural design choices, custom loss formulations, and stabilization strategies required for successful convergence of this newly introduced approach, three key experiments were designed.

Experiments 3: Stage 2 Baseline Convergence and Sequence Stabilization

E3.1 Decoupled Hierarchical Token Prediction: Evaluating if splitting the 4-layer residual token space, by predicting the base pose first and the remaining three detail layers next, allows the model to learn movements correctly.

E3.2 Mitigating Exposure Bias via Scheduled Sampling: Testing whether transitioning the network from strict teacher-forcing to closed-loop auto-regressive feedback prevents long-term sequence decay and positional drifting during inference.

E3.3 Dual-Space Distance-Aware Optimization: Validating if balancing a distance-based categorical loss with a continuous coordinate loss allows the network to learn both token logic and physical boundaries together.

5.1.2 Decoder-Only Sequence Architecture and Decoupled Prediction

To evaluate the operational differences between streaming and non-causal systems, this framework implements a localized auto-regressive processing layer. A conceptual layout of how this configuration handles temporal dimensions relative to offline approaches is structured in Table 5.1.

Table 5.1: Sequential processing constraints across architectural paradigms.

Architectural Approach	Input Vectors (X)	Output Target (Y)	Temporal Constraint
Encoder-Only Masked	Full Audio $A_{1:T}$ + Masked Motion $G_{1:T}$	Reconstructed Motion $\hat{G}_{1:T}$	Non-Causal (Full Look-Ahead)
Encoder-Decoder	Full Audio $A_{1:T}$ + Full Text $W_{1:M}$	Reconstructed Motion $\hat{G}_{1:T}$	Non-Causal Bottleneck
Diffusion	Random Noise x_s + Audio Context $A_{1:T}$	Denoised Motion Trajectory x_0	Iterative Non-Causal Sampling
Decoder-Only Auto-Regressive	Past Poses $G_{<t}$ + Audio Chunk $A_{\leq t}$	Next-Frame Postural State g_t	Strictly Causal (Live Streaming)

The core sequence model is built around a causal, decoder-only transformer architecture designed to predict upper-body gesture tokens frame-by-frame. The network consists of two sequential transformer layers with a hidden embedding dimension of 256, 8 attention heads, and a feedforward network dimension of 1024. A dropout rate of 0.15 is applied throughout the layers, and training sequences are constrained to a maximum context length of 64 frames to ensure steady sequence bounds.

By framing gesture generation as a sequential next-token prediction problem, the auto-regressive framework models the sequence by factorizing the joint probability of the gesture tokens across the compressed timeline T' conditional on the chunk-based speech audio features F_{audio}:

$$P(K \mid F_{\text{audio}}) = \prod_{t'=1}^{T'} P(k_{t'} \mid k_{<t'}, F_{\text{audio},\leq t'}) \tag{5.1}$$

To ensure the network remains strictly online and never uses future context, the transformer enforces a unidirectional processing window. This constraint is implemented by applying a lower-triangular causal attention mask $M_{i,j}$ that zeroes out all future index positions across the internal attention timeline:

$$M_{i,j} = \begin{cases} 0 & \text{if } j \leq i \\ -\infty & \text{if } j > i \end{cases} \tag{5.2}$$

At each operational time step t', the decoder processes the causal history to build a hidden sequence state $h_{t'} = \mathcal{T}_{\text{decoder}}(k_{<t'}, F_{\text{audio},\leq t'})$. This state projects through a linear base classifier head to output categorical logits over the vocabulary, allowing the system to compute the predicted token probability distribution:

$$\hat{p}_{t'} = \text{Softmax}\left(\text{Linear}(h_{t'})\right) \tag{5.3}$$

By discarding the global future window, this token-by-token setup eliminates processing latency, generating motion frame-by-frame matching live streaming audio inputs in real-time.

A central architectural challenge arises from the hierarchical nature of the Stage 1 RVQ-VAE, which produces a matrix of $D = 4$ discrete codebook tokens per temporal frame. Flattening this matrix into a single linear sequence would quadruple the effective sequence length, causing the self-attention computational cost to scale quadratically while disrupting the temporal synchronization with the audio and text data. To resolve this complexity, the architecture implements a decoupled, two-step prediction framework that treats the token layers asymmetrically based on their structural priority.

First, the primary auto-regressive pathway handles the overall body sequence modeling. The causal transformer processes the combined multi-modal history and projects the resulting hidden states through the base classifier to predict the Layer 1 token for the current time step. This Layer 1 token establishes the broad geometric trajectory and structural skeleton of the gesture. Auto-regressive temporal continuity is maintained across the global timeline by feeding the predicted Layer 1 tokens of previous steps back into the transformer's input stream to compute subsequent frames.

Second, fine-grained physical features are resolved via a decoupled residual predictor module. Once the baseline Layer 1 token is computed for the current frame, its continuous codebook embedding is concatenated with the main transformer's hidden features. This joint spatial representation is passed through three dedicated cross-attention layers. Instead of extending the global timeline, the module iteratively predicts the remaining residual codes (Layers 2, 3, and 4) by appending the embeddings of previously resolved layers to the local cross-attention context.

Training results show a clear difference between these prediction layers. The optimization landscape allows the network to achieve a stable, low loss on the Layer 1 tokens, whereas the higher-level residual layers display higher categorical losses. This distribution is structurally acceptable. Because the foundational Layer 1 token dictates the global postural layout, its convergence is the most important factor for making the motion look clear and natural, while the higher-order residual blocks primarily handle non-structural spatial details such as hand shapes and finger articulations.

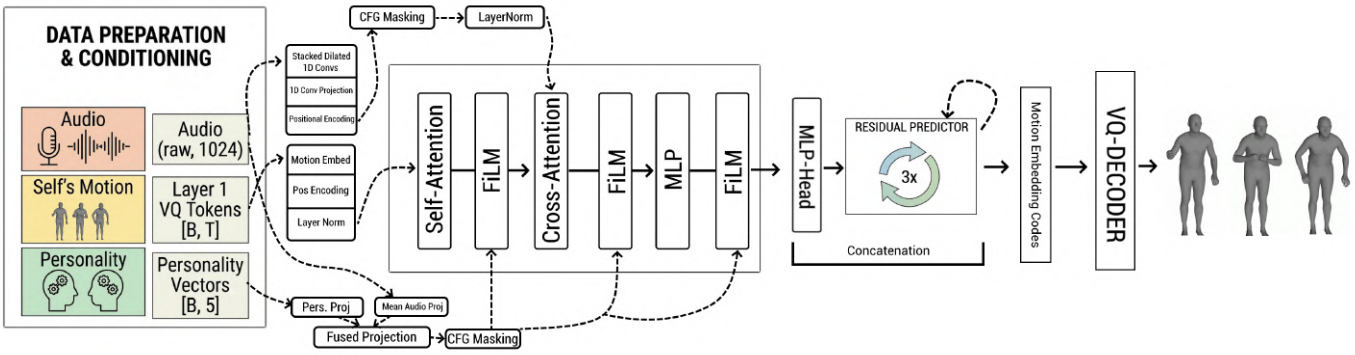


Figure 5.1: Architecture of the Stage 2 auto-regressive transformer. Multi-modal inputs (audio, discrete motion tokens, and personality vectors) are processed through dedicated embedding streams. Continuous audio is encoded via stacked dilated 1D convolutions to provide localized timing data for cross-attention, while personality traits and mean audio are fused into a global condition vector that modulates the network via FiLM layers. After applying Classifier-Free Guidance (CFG) masking, the causal transformer predicts the base motion code via an MLP head, which is subsequently processed by a decoupled residual predictor to generate the remaining fine-grained codes for the Motion Code-Decoder that was learned in Stage 1.

5.1.3 Multi-modal Conditioning and Classifier-Free Guidance

To ensure that the generated gestures mirror the natural rhythms of conversation and align with the speaker's profile, the sequence architecture integrates a dual-scale multi-modal conditioning framework. This system uses two separate processes to combine local speech tracking with overall behavioral traits.

Local synchronization maps the fine-grained, time-sensitive pacing of speech to physical motion beats. Continuous acoustic feature vectors F_{audio} are passed through a deep 1D convolutional network structured into two sequential blocks. Within each block, a series of convolutional filters operate over expanding dilated receptive fields with dilations scaling from 1 to 3. This dilation pattern allows the audio model to extract temporal contextual acoustic structures without loss of resolution. The resulting acoustic features are projected into a dense 256-dimensional matrix ($d_h = 256$) that serves directly as the keys K_c and values V_c within the transformer's cross-attention blocks, mapping audio changes to physical gesture movements.

At the same time, style conditioning controls the overall energy and behavior of the generated gestures. Static attributes, including the speaker's continuous personality traits f_{ps}, are normalized and mapped through an MLP to generate a continuous speaker identification embedding. This trait embedding is fused with a global intensity representation computed as the temporal mean of the localized audio sequence. The final concatenated condition vector c is processed by a series of FiLM layers. Within each transformer layer, the FiLM block computes dynamic scale $\gamma(c)$ and shift $\beta(c)$ vectors to linearly modulate the hidden feature

map h at three critical structural points: immediately following self-attention, after multi-head cross-attention, and within the feedforward network block.

To maximize the model's responsiveness to these control vectors, a Classifier-Free Guidance (CFG) masking strategy is implemented during training. With a fixed probability of 10%, the multi-modal conditioning inputs are randomly dropped and replaced with learned, blank parameter embeddings. This intentional degradation prevents the transformer from over-relying on continuous audio streams or locking into static speaker biases. By forcing the network to optimize both an unconditional motion distribution and a guided conditional distribution, CFG masking provides robust sequence stability.

5.1.4 Dual Latent-Space Optimization Strategy

Standard auto-regressive text models assume categorical orthogonality, optimizing sequence predictions via hard-label cross-entropy objectives where any mismatched token index incurs an identical penalty. In gesture generation, however, this assumption breaks down. Distinct indices within an un-ordered vector quantization codebook frequently map to continuous coordinate spaces that are physically adjacent or identical. To address this topology, a dual-space optimization strategy is implemented that penalizes categorical errors proportionally to their continuous spatial distance. The final combined objective function is formulated as:

$$L_{\text{total}} = \lambda_{\text{CE}} L_{\text{soft-CE}} + \lambda_{\text{MSE}} L_{\text{MSE}} \tag{5.4}$$

where $\lambda_{\text{CE}} = 1.0$ and $\lambda_{\text{MSE}} = 100.0$ represent the assigned weighting scales for the discrete and continuous supervision targets.

Distance-Aware Soft Cross-Entropy

To regularize the categorical classification space, the target hard labels are smoothed based on the geometric proximity of their corresponding vectors within the Stage 1 latent codebook. Prior to training, a global distance matrix is computed across the pairwise Euclidean distances between all codebook vectors in $\mathcal{C}$. For a ground-truth sequence token index y, a soft target probability distribution $q_{t'}(j)$ is dynamically generated by scaling the negative physical distances through a Softmax function with a tight temperature parameter $\tau = 0.05$:

$$q_{t'}(j) = \frac{\exp\left(-\frac{\|e_y - e_j\|_2}{\tau}\right)}{\sum_{k=1}^{K} \exp\left(-\frac{\|e_y - e_k\|_2}{\tau}\right)} \tag{5.5}$$

where:

- $q_{t'}(j)$ represents the smoothed target probability assigned to the codebook index j at compressed temporal step t'.
- e_y and e_j denote the continuous coordinate codebook vectors corresponding to the ground-truth index y and the candidate index j within the shared vocabulary $\mathcal{C}$.
- τ is the temperature scaling hyperparameter regulating the distribution smoothness ($\tau = 0.05$).
- K represents the total vocabulary size of the shared codebook ($K = 512$).

The discrete sequence target is subsequently optimized via the soft cross-entropy loss ($L_{\text{soft-CE}}$) across all batch items B, compressed temporal sequence steps T', and quantization layers D:

$$L_{\text{soft-CE}} = -\frac{1}{B \cdot T' \cdot D} \sum_{b=1}^{B} \sum_{t'=1}^{T'} \sum_{d=1}^{D} \sum_{j=1}^{K} q_{b,t',d}(j) \log \hat{p}_{b,t',d}(j) \tag{5.6}$$

where:

- B represents the training batch size ($B = 32$).
- T' denotes the maximum compressed sequence context length under evaluation ($T' = 16$).
- D is the depth of the residual vector quantization hierarchy ($D = 4$).
- $q_{b,t',d}(j)$ is the smoothed target probability value for a specific batch b, compressed time step t', layer d, and codebook entry j.
- $\hat{p}_{b,t',d}(j)$ represents the model's predicted categorical probability distribution over the codebook vocabulary, computed by applying a Softmax function over the raw classification logits.

By making use of this distance-aware smoothing, the transformer is not penalized heavily for selecting structurally similar pose variations, which guides the classification heads to naturally learn the underlying physical layout of the motion space.

Continuous Latent Supervision

To reinforce joint coordinate boundaries and prevent spatial drift, the categorical output distributions are tied directly to continuous latent coordinate supervision. For each discrete logit distribution predicted by the classification heads, a soft-quantized continuous latent vector prediction $\hat{z}_{b,t'}$ is computed as the mathematical expectation of the codebook space. This is achieved by taking the weighted sum of all continuous codebook vectors relative to their predicted probabilities $\hat{p}$ across all quantizer layers:

$$\hat{z}_{b,t'} = \sum_{d=1}^{D} \sum_{j=1}^{K} \hat{p}_{b,t',d}(j) e_j \tag{5.7}$$

where $e_j \in \mathbb{R}^{64}$ represents the dense coordinate vector corresponding to the j-th entry in the shared codebook. This predicted continuous coordinate is supervised directly against the true continuous latent target vector $z_{b,t'}$ extracted from the frozen Stage 1 encoder using a Mean Squared Error (MSE) objective:

$$L_{\text{MSE}} = \frac{1}{B \cdot T' \cdot d_{\text{feat}}} \sum_{b=1}^{B} \sum_{t'=1}^{T'} \sum_{i=1}^{d_{\text{feat}}} \left(\hat{z}_{b,t',i} - z_{b,t',i}\right)^2 \tag{5.8}$$

where:

- d_{feat} represents the latent feature dimensionality of the continuous codebook embedding space ($d_{\text{feat}} = 64$).
- $\hat{z}_{b,t',i}$ represents the value of the i-th dimension of the soft-quantized predicted continuous latent vector $\hat{z}_{b,t'}$.
- $z_{b,t',i}$ represents the value of the i-th dimension of the true ground-truth continuous target vector $z_{b,t'}$ mapped directly from the Stage 1 encoder.

This continuous spatial loss acts as a geometric anchor, heavily penalizing any distribution shift that approaches physically implausible body postures or tracking boundaries.

5.1.5 Training Configuration and Convergence Strategies

The Stage 2 model is optimized from scratch using the AdamW optimizer across a total of 200 training epochs, that finished in 8 hours of training. The data pipeline handles a complete, unfiltered 150-hour subset of the dyadic corpus, allowing the transformer to capture continuous interaction transitions across both active and passive conversational states. The global training batch size is set to 32. The learning rate is controlled by a structured schedule, beginning with a 5-epoch linear warmup to a maximum base rate of 0.0003, followed by a cosine decay phase across 150 epochs down to a minimum learning rate ratio of 0.05. To maintain optimization stability, gradients are monitored and strictly bounded at a maximum gradient norm clipping threshold of 1.0.

A engineering hurdle in training auto-regressive sequence models on continuous human motion is exposure bias. During standard teacher-forced training, the network is fed perfect ground-truth historical tokens at every step. During inference, however, the model must rely on its own previous predictions, causing minor tracking errors to accumulate rapidly, resulting in severe positional drift and eventual sequence collapse. Experimental runs confirmed that optimizing the transformer under strict, unadjusted teacher forcing completely failed to converge, resulting in large validation errors alongside low training losses due to severe overfitting.

To enforce network convergence, a dynamic scheduled sampling mechanism is integrated into the training pipeline. The model begins training under complete teacher forcing, but as the epochs progress, ground-truth historical tokens are gradually replaced with the model's own prior Layer 1 predictions. The replacement probability scales linearly relative to the training step percentage, capping at a maximum token replacement ratio of 50%:

$$P_{\text{sampling}} = P_{\text{max\_sampling}} \cdot \text{Step}_{\text{percentage}} \tag{5.9}$$

By forcing the residual networks and downstream temporal blocks to frequently learn from imperfect, self-generated tokens rather than ground truth, scheduled sampling trains the model to actively correct its own trajectory deviations. This stabilization strategy bridges the structural gap between training constraints and closed-loop inference, enabling long-term sequence stability during real-time generation.

5.1.6 Experimental Inference Pipeline: Hybrid Asymmetric Causality

To bring the generative architecture closer to real-time, "live" streaming capabilities, this work proposes a novel, highly experimental closed-loop auto-regressive inference pipeline.

Traditional auto-regressive models typically operate under strict, symmetric causal constraints across all input modalities. However, during the developmental phase, it was discovered that conversational human gestures rely on immediate acoustic anticipation. To leverage this, an asymmetric temporal causality approach is used. This strictly enforces causal boundaries on the physical motion history while permitting a short future window in the audio conditioning stream. This hybrid configuration is unique to this framework and directly addresses the synchronization challenges inherent to frame-by-frame streaming.

The overall inference mechanism is experimental and operates in a sliding-window fashion, generating the gesture stream frame-by-frame. A token history matrix of shape $B \times N_{\text{total}} \times d_s$ is pre-allocated, where B represents the batch size, N_{total} is the total sequence length in downsampled tokens, and $d_s = 4$ is the depth of the hierarchical RVQ-VAE quantizer. If a warm-start seed is selected, the generation is initialized by encoding the first 64 frames of ground-truth upper-body postures through the Stage 1 model, to write the initial 16 tokens directly into the history buffer. If warm-start is disabled, the first token is initialized with a random seed token, and the sequence builds up step-by-step from scratch. At each step, the sliding window ingests a history of exactly 15 tokens (corresponding to 60 raw pose frames) to predict the 16^{th} token (which represents the next 4 physical frames of upper-body motion).

Audio Future Context Alignment

To maintain a continuous streaming state, the approach feeds exactly 4 new raw frames of speech audio into the pipeline at each step. These 4 pose frames are mapped to the audio stream via a temporal downsampling ratio:

$$T_{\text{audio}} = \left\lfloor T_{\text{pose}} \times \frac{50\,\text{Hz}}{30\,\text{Hz}} \right\rfloor \tag{5.10}$$

where $T_{\text{pose}} = 64$, the audio sample rate is 50 Hz (via Wav2Vec), and the pose frame rate is 30 Hz. When predicting the target token index n, the dilated 1D convolutional audio encoder is permitted to process the speech envelope up to the end of the current target frame window.

This localized look-ahead provides the cross-attention layers with crucial, immediate speech and rhythmic context, allowing the network to align the physical acceleration peaks (the structural beat) precisely to the speech envelope. Because the motion stream remains strictly bound to past coordinates, this slight audio future alignment improves beat synchronization without introducing noticeable physical lag or compromising the causal integrity of the streaming agent.

Hybrid Token Sampling Approach

A custom, two-phase sampling protocol was developed to balance physical variation with motion stability, mitigating the risk of auto-regressive sequence collapse.

1. Base Layer Probabilistic Sampling The primary transformer head outputs a raw logit vector $l_{t'} \in \mathbb{R}^V$ representing the unnormalized classification scores for the structural Layer 1 token at step t'. The term V denotes the total vocabulary size of the discrete motion codebook ($V = 513$), which encompasses the 512 vector-quantized motion primitives along with an additional masking token. Co-speech gesture generation is fundamentally characterized by a one-to-many mapping problem, a single audio pattern or spoken word can naturally correspond to a vast multitude of physically valid gestures. To reflect this distribution and avoid deterministic, repetitive, or robotic movements, a probabilistic sampling strategy is used to decode the layer 1 motion codes. The raw logits are scaled by a temperature parameter θ, and subsequently filtered using a combination of Top-K and Nucleus (Top-P) sampling. The predicted base token index $\hat{y}_{t'}^{(1)}$ is then stochastically drawn from the resulting filtered probability distribution:

$$\hat{y}_{t'}^{(1)} \sim \text{Multinomial}\left(\text{Softmax}\left(\text{Filter}_{P,K}\left(\frac{l_{t'}}{\theta}\right)\right)\right) \tag{5.11}$$

where $\theta \in \mathbb{R}^+$ is the temperature parameter regulating the entropy of the distribution, $K \in \mathbb{Z}^+$ restricts sampling to the top K highest-scoring primitives, and $P \in (0, 1]$ represents the cumulative probability threshold for nucleus selection. Implementing this sampling framework enables the generation of diverse, creative, and natural motion variations under identical vocal conditioning contexts, preventing the model from collapsing into a single, uniform motion path.

2. Deterministic Residual Reconstruction Once the base structural token $\hat{y}_{t'}^{(1)}$ is established, it is mapped back to its continuous codebook representation to query the decoupled residual predictor module. Unlike the main pose modeled by Layer 1, the remaining fine-grained residual layers (Layers 2, 3, and 4) dictate detailed physical alignments, including small movements of the hands and finger articulations. To keep sampling errors from piling up and causing motion drift during generation, a strict greedy decoding process is used:

$$\hat{y}_{t'}^{(d)} = \arg\max_j \left(l_{t',j}^{(d)}\right) \quad \forall\ d \in \{2, 3, 4\} \tag{5.12}$$

where $l_{t'}^{(d)} \in \mathbb{R}^V$ represents the raw classification logit vector output by the residual predictor for the d-th layer of the hierarchy at time step t', and $\hat{y}_{t'}^{(d)}$ denotes the resulting chosen token index.

The final generated token indices $[\hat{y}_{t'}^{(1)}, \hat{y}_{t'}^{(2)}, \hat{y}_{t'}^{(3)}, \hat{y}_{t'}^{(4)}]$ are combined into a single hierarchical feature vector and written directly to the pre-allocated temporal history buffer. The sliding generation window is then advanced forward by exactly one token block (corresponding to 4 raw pose frames), feeding the newly generated motion tokens back into the causal auto-regressive stream as historical context for subsequent predictions. Splitting the decoding process, keeping the main body movements probabilistic and fine details deterministic, allows the pipeline to generate expressive gestures while maintaining physical correctness.

5.1.7 Training Convergence and Loss Behavior

To evaluate the learning dynamics and generalization capacity of the sequence model, the training and validation metrics were tracked across the 200-epoch optimization pipeline. Table 5.2 presents the corrected empirical progression of the loss parameters at key epoch values, capturing the initial state (Epoch 0), the optimization midpoint (Epoch 100), and the final convergence step (Epoch 199).

Table 5.2: Empirical Tracking of Causal Model Convergence Parameters Across Epochs

Metric Category	**Epoch 0 (Initial)**		**Epoch 100 (Midway)**		**Epoch 199 (Final)**	
	Train	**Val**	**Train**	**Val**	**Train**	**Val**
Averaged Cross-Entropy ($\mathcal{L}_{\text{CE, avg}}$)	4.8280	4.7040	3.1131	4.4527	3.1495	4.5047
Base Layer 1 Logits ($\mathcal{L}_{\text{CE, L0}}$)	2.5363	2.1546	1.5543	2.0169	1.6013	2.0566
Residual Layer 2 Logits ($\mathcal{L}_{\text{CE, L1}}$)	5.2708	5.2697	2.8828	4.7722	2.9450	4.8560
Residual Layer 3 Logits ($\mathcal{L}_{\text{CE, L2}}$)	5.6497	5.5781	3.6807	5.3095	3.7107	5.3522
Residual Layer 4 Logits ($\mathcal{L}_{\text{CE, L3}}$)	5.8554	5.8138	4.3347	5.7124	4.3408	5.7540
Coordinate Distance ($\mathcal{L}_{\text{MSE}}$)	0.0168	0.0144	0.0115	0.0148	0.0114	0.0145
Weighted Scalar Summary ($\mathcal{L}_{\text{total}}$)	6.5123	6.1419	4.2660	5.9324	4.2857	5.9535
Learning Rate Parameter (η)	1.20×10^{-4}		8.80×10^{-5}		1.50×10^{-5}	

Analysis of Categorical Losses

To understand the physical meaning of the categorical cross-entropy losses, the loss values can be interpreted in terms of information-theoretic perplexity. Since the classification heads are optimized using a natural logarithm-based soft cross-entropy objective, the effective size of the token search space selected by the model at any given step is mathematically represented as:

$$P(x) = \exp\left(\mathcal{L}_{\text{CE}}\right) \tag{5.13}$$

Given a total discrete vocabulary size of $|C| = 512$ codebook vectors per layer, a completely uniform random prediction would yield a maximum cross-entropy loss of $\log(512) \approx 6.24$.

By Epoch 199, the training classification loss for the foundational Base Layer (Layer 1) converges to 1.6013, which corresponds to an exceptionally narrow effective search space of $\exp(1.6013) \approx 4.96$ candidate tokens. On the validation partition, the Base Layer loss stabilizes at 2.0566, restricting the active prediction window to approximately 7.82 equivalent options. This significant reduction from the initial 512-token vocabulary demonstrates that the model successfully isolates the underlying structural main-pose layouts of the gesture space, reducing structural choices to roughly 5 to 8 plausible options at any given frame.

Hierarchical Loss Distribution and Generalization

The training results show a clear split between the main body movements and the fine details, proving that the model works exactly as designed. For the main Base Layer (Layer 1), the validation loss (2.0566) closely follows the training loss (1.6013). This shows strong generalization, meaning the model successfully links speech features to core body movements without overfitting.

As the depth increases into the residual sections (Layers 2-4), the losses increase, with validation losses reaching 4.8560 for Layer 2, 5.3522 for Layer 3, and 5.7540 for Layer 4. The average loss ($\mathcal{L}_{\text{CE, avg}}$) reflects this climb, landing at a training value of 3.1495 and a validation value of 4.5047. This happens because of the multi-scale setup:

- **The Base Layer (Layer 1)** handles the main, predictable body postures, which align closely with the overall audio and speech rhythm.
- **The Residual Layers (Layers 2-4)** handle the fast, detailed hand and finger movements.

Since a single spoken phrase can naturally be accompanied by many different hand movements, these deeper layers have higher losses because they capture the unpredictable variety of human gestures. Because the main body structure is kept stable by the low-loss Base Layer, this variation does not break the skeleton, instead, it should prevent rigid, robotic motions and keeps the gestures looking natural.

5.1.8 Chapter Discussion & Conclusion

This chapter has presented a hierarchical auto-regressive architecture designed to separate main body movements from fine details. By combining probabilistic decoding for the Base Layer with deterministic greedy reconstruction for the residual layers, the pipeline successfully handles the one-to-many problem of generating co-speech gestures. The training results in Section 5.1.7 confirm that the model stabilizes the motion space while keeping physical movements structurally sound.

While the initial results are promising, this multi-scale setup is still experimental, and several areas need further study. Currently, the inference pipeline relies on a static set of sampling settings that have not been fully optimized. Future work could explore using higher sampling temperatures (θ) for more fluid motion, or adjusting top-P and top-K limits to see where movement quality starts to drop.

Additionally, the training currently treats the losses across all layers with equal priority. Using different loss weights for each layer is a promising direction. Weighing the Base Layer more heavily than the unpredictable residual layers could make the overall movements even more stable.

For this work, however, the current setup is fully sufficient. The model converges reliably, stabilizes the movements, and creates a solid baseline. This baseline serves as the foundation for the comparisons and ablation studies in Chapter 7.

Chapter 6

Visual Rendering Pipeline

In gesture generation, quantitative metrics such as joint distance or cross-entropy losses often fail to capture the subtle physical nuances, structural deformities, and coordinate drifts that define overall generation quality. Therefore, directly watching the generated motion is necessary to check the meshes, fix timing issues, and evaluate the overall interaction. To make qualitative evaluation fast and repeatable, this chapter presents a web-based rendering pipeline designed to bypass the slow workflows of traditional 3D software.

Contributions 4: Open-Source Unified Gesture Visualization Platform

1. **Interactive Web Application:** The development of a, interactive web interface designed to completely replace manual animation suite workflows and dramatically accelerate visual evaluation loops.
2. **Cross-Dataset Unified Rendering Support:** Standardized visual rendering capable of parsing both SMPL-H and SMPL-X topologies, enabling evaluation across different benchmark datasets.
3. **Multi-Stage Pipeline Diagnostic Interfaces:** The implementation of debugging tabs to verify raw database entries, check autoencoder reconstructions, and test interactive real-time sampling.

6.1 Motivation and Limitations of Existing Visualization Pipelines

While gesture generation algorithms have improved quickly, standard visualization tools have not kept up. Because the field is still new, fragmented software tools make it difficult for researchers to visually inspect their models.

This fragmentation especially shows in the lack of unified support for diverse SMPL formats. While popular benchmarks like the *BEAT2* dataset have established SMPL-X as a widely supported standard, other modern datasets, such as the Seamless Interaction dataset [2], make us of the the SMPL-H body model. Currently, there is a shortage of open-source, easily deployable rendering solutions that can bridge these two formats directly without requiring custom software or manual data preparation.

The primary limitation in existing research workflows is the reliance on offline 3D animation software, most notably Blender. While Blender is technically capable of rendering meshes from SMPL-H and SMPL-X parameters, using it for iterative model development introduces workflow challenges:

- **Disrupted Hyperparameter Feedback Loop:** Even with local scripts that output standard `.npz` motion files, seeing the immediate visual impact of sampling hyper-parameters (such as temperature, Top-K, or Top-P) requires a slow offline loop. To test a single parameter change, researchers have to manually modify the script, run the model to save a new file, import it into Blender, and refresh the timeline, which slows down parameter tuning.
- **Complex Scene Setup:** Researchers have to manually build scenes, position virtual cameras, configure lighting, and map joint rotations onto skeletal armatures for every single test generation.
- **De-synchronized Multi-Modal Inputs:** Aligning the generated skeletal motion with its corresponding vocal audio track is an offline, manual step. This makes frame-accurate, multi-modal verification (such as checking rhythmic synchronization) slow and difficult.

Because of this, the lack of an open-source visualization tool, slows down qualitative evaluation. Without a way to quickly turn model coordinates into rendered motion, finding errors like joint warping, frozen poses, or motion drift is difficult. This gap highlights the need for a unified, real-time rendering pipeline designed specifically for gesture generation.

6.2 Interactive Rendering Architecture and Method

To solve the visual evaluation bottleneck, a custom, real-time rendering and evaluation pipeline was developed. The backend uses Pyrender with an Embedded-System Graphics Library (EGL) driver to enable headless, offscreen OpenGL rendering directly on

the GPU. This setup allows for fast frame generation without requiring an active server or display window, making it easy to run on remote servers.

The system maps predicted joint parameters directly onto the SMPL-H body mesh to match the Seamless Interaction dataset, while keeping modular support for the SMPL-X standard used in the BEAT2 dataset. An automated script then pulls the matching input audio from the database, synchronizes it frame-accurately with the rendered video, and outputs a unified multimedia file for instant review.

Experiments 4: Interactive Pipeline Verification and Diagnostics

E4.1 Isolated Sequence Evaluation (Tab 1): Isolating individual interaction sequences to check localized joint motion, camera positioning, and mesh rendering quality.

E4.2 Dyadic Interaction Auditing (Tab 2): Rendering two character meshes facing each other in a shared 3D space to visually evaluate conversational turn-taking and distance.

E4.3 Autoencoder Reconstruction Validation (Tab 3): Loading the trained Stage 1 RVQ-VAE weights to map the predictions back onto the skeleton, comparing the reconstructed mesh side-by-side with the ground truth.

E4.4 Auto-regressive Parameter Sensitivity (Tab 4): Testing the generative Stage 2 transformer in real-time to see the visual effects of sampling temperature, Top-K, Top-P, and history-context seeding on motion behavior.

E4.5 Dataset Diagnostic Slicing (Tab 5): Visualizing raw binary LMDB chunks by interaction ID to check that the sliding-window slicing and boundary indexing do not introduce breaks in the motion.

6.2.1 Tab 1: Isolated Sequence Visualization (Single Visualizer)

The main interface for analyzing motion detail is the Single Visualizer tab (shown in Figure 6.1a). This component handles rendering of individual, isolated interaction sequences. By mapping joint rotations onto a single SMPL-H body-model, the visualizer allows researchers to closely inspect skeletal proportions, joint angles, and coordinate-space transitions. This tab serves as the primary diagnostic tool to verify that the model's spatial outputs translate into natural body postures without joint warping, self-collisions, or clipping, before evaluating two-person interactions.

6.2.2 Tab 2: Conversational Dyadic Interaction Analysis

Evaluating conversational gestures requires analyzing not just individual postures, but how two speakers coordinate in space and time. The Dyadic Interaction panel (Figure 6.1b) renders two distinct SMPL-H character meshes facing each other within a shared 3D coordinate space. This dual-character layout is important for verifying how the two interact, showing turn-taking, gestural mimicry, and bodily orientation. It allows the researcher to visually check if the generated gestures align naturally with the flow of the conversation and the physical distance between both speakers.

6.2.3 Tab 3: Stage 1 Reconstruction Auditing

The Stage 1 Reconstruction tab (Figure 6.1c) provides a visual baseline for the discrete latent space. It loads the trained weights of the Stage 1 Residual Vector Quantized Variational Autoencoder (RVQ-VAE) and decodes the quantized vectors, back onto the skeleton. By displaying a synchronized, side-by-side rendering of the original ground-truth sequence and the reconstructed mesh, this interface allows for checking the compression quality. Any loss of expressive detail, hand gestures, or upper-body structural deformation introduced during quantization can be immediately spotted and analyzed.

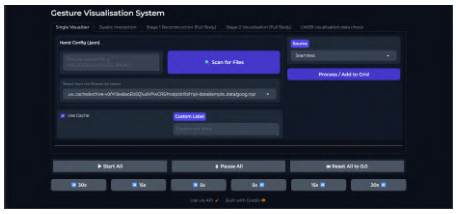
6.2.4 Tab 4: Interactive Autoregressive Sampling and Generation

The Stage 2 Visualization tab (Figure 6.1d) serves as the interactive environment for the generative model. Instead of relying on a slow, offline command-line loop, this panel allows researchers to run the Stage 2 causal transformer. Users can interactively tune sampling hyper-parameters. Specifically the generation temperature (θ), Top-K boundaries, Top-P nucleus thresholds, and the length of historical context seeding.

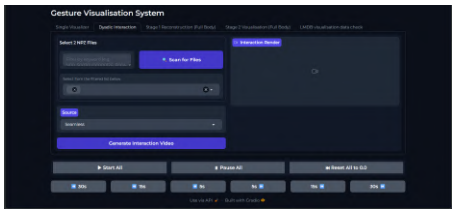
6.2.5 Tab 5: Low-Level LMDB Data Diagnostic and Integrity Auditing

To guarantee that downstream generation errors do not stem from upstream pre-processing mistakes, the LMDB Diagnostic tab (Figure 6.1e) acts as a low-level binary data visualizer. It directly accesses the raw LMDB database, extracting and re-stitching 64-frame binary chunks based on specific interaction IDs. By rendering the raw, unprocessed dataset entries, this tab allows researchers to visually inspect the temporal sliding-window slicing logic and step-indexing. This ensures that the transitions between windows are entirely seamless and free of indexing shifts, data corruption, or breaks in the motion.

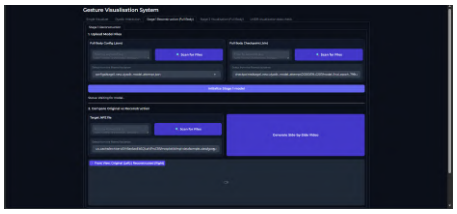
Finally, to allow for comparative analysis, a global control block using custom JavaScript synchronizes the timeline playback across all active rendering panels. This allows the researcher to instantly observe how slight changes in sampling statistics influence motion fluidity, diversity, and physical stability in real time, dramatically accelerating the parameter tuning loop.



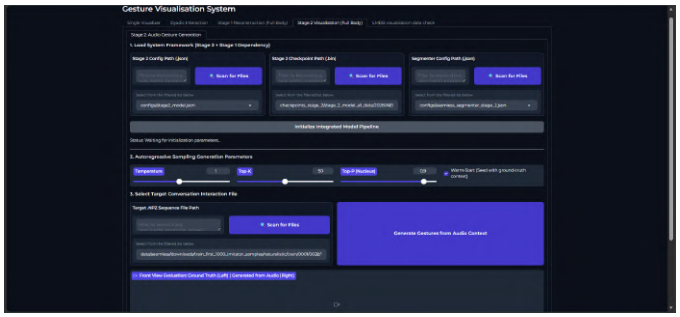
(a) Single Visualizer



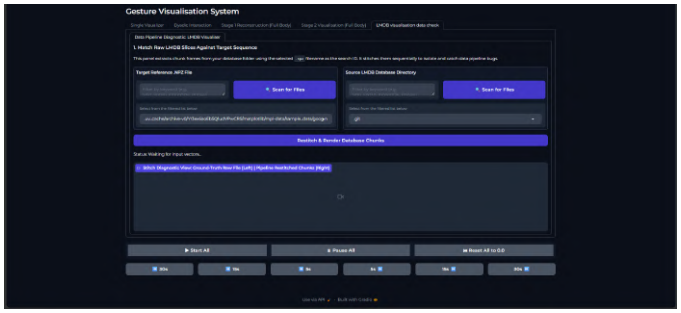
(b) Dyadic Interaction



(c) Stage 1 Reconstruction



(d) Stage 2 Visualisation



(e) LMDB Diagnostic

Figure 6.1: Overview of the custom Gesture Visualisation System. (a) **Single Visualizer:** Allows for isolated inspection of individual interaction sequences. (b) **Dyadic Interaction:** Renders two SMPL-H character meshes in a shared 3D space to evaluate conversational dynamics. (c) **Stage 1 Reconstruction:** Provides a qualitative baseline by rendering side-by-side comparisons of ground-truth files and model-decoded meshes. (d) **Stage 2 Visualisation:** Enables interactive testing of the auto-regressive transformer by adjusting generation parameters such as temperature, Top-K, and Top-P. (e) **LMDB Diagnostic:** Extracts and restitches raw database chunks to verify temporal alignment and data integrity across the pipeline. This rendering pipeline supports both SMPL-H and SMPL-X body models, providing an accessible, out-of-the-box evaluation tool for researchers utilizing datasets like BEAT2 and serving as a broader contribution to the gesture generation community.

6.3 Comparative Visual Results

The practical value of this custom rendering pipeline is shown through its real-time outputs, which give immediate visual feedback at different stages of the research. Instead of requiring slow manual importing, key-frame alignment, and camera setup in software like Blender, the system automatically converts coordinate streams into synchronized, high-quality animations.

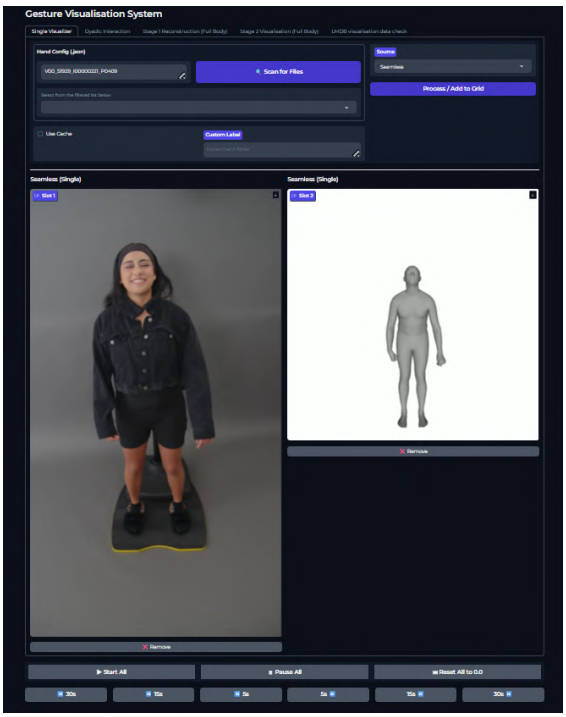


Figure 6.2: The custom Gesture Visualization System. The interface displays a synchronized, side-by-side comparative grid showing the original reference video of a human participant from the Seamless Interaction dataset (left, Slot 1) directly alongside the rendered SMPL-H mesh (right, Slot 2). Integrated control bars enable synchronized playback, timeline seeking, and interactive visualization adjustments.

As illustrated in Figure 6.2, the Single Visualizer interface bridges the gap between raw data and physical performance by positioning the original ground-truth video directly next to the rendered 3D mesh. This side-by-side layout allows researchers to qualitatively verify anatomical accuracy, tracking alignment, and gestural expression relative to the physical participant.

(a) Dyadic Interaction Layout

(b) Side-by-Side Reconstruction/Recreation

Figure 6.3: Visual configurations generated directly by the headless rendering backend. (a) **Dyadic Interaction Layout:** Renders two interacting subjects facing each other in a shared environment to evaluate conversational interaction. (b) **Symmetric Comparison Layout:** Positioned side-by-side to visually inspect differences between Ground-Truth (GT) sequences and autoencoder reconstructions, generated outputs, or raw database structures.

The custom backend supports multiple camera and character configurations tailored to specific analytical tasks:

- **Dyadic Evaluation:** As shown in Figure 6.3a, the system renders two avatars facing each other within a unified virtual

coordinate space. This is essential for checking how the two speakers interact, showing turn-taking cues and conversational timing that are completely lost when looking at individual sequences in isolation.

- **Symmetric Reconstruction Tracking:** As shown in Figure 6.3b, the interface supports a clean side-by-side mesh comparison. This is valuable for comparing ground-truth targets directly against model predictions (such as Stage 1 VAE reconstructions, Stage 2 auto-regressive generations, or raw LMDB records). Because these renders are generated from identical camera perspectives and lighting angles, even minor spatial errors, joint jitter, or motion drift become immediately obvious.

By bypassing the disconnected workflows of traditional animation software, this integrated pipeline turns qualitative assessment from an occasional, labor-intensive chore into a fast, interactive loop that can be run at any point during model training.

6.4 Chapter Discussion & Conclusion

This chapter detailed the design and implementation of a custom rendering pipeline designed specifically for co-speech gesture generation. By combining a headless Pyrender engine and an EGL-enabled OpenGL backend within an interactive web app, this tool easily bridges the gap between raw coordinates and actual human motion. Supporting both SMPL-H and SMPL-X skeleton formats resolves a major software fragmentation issue in the research community.

This tool turns qualitative model auditing from a slow, offline chore into an interactive, real-time feedback loop. Instead of dealing with tedious manual importing and alignment in Blender, researchers can quickly check dataset structures, inspect VAE reconstruction errors, and test auto-regressive hyper-parameters. Specialized layouts, like the two-avatar space and side-by-side ground-truth comparison panels, make features like conversational turn-taking, body alignment, and motion stability immediately obvious.

Ultimately, this pipeline is more than just a convenience. It is a key part of this thesis. Removing the manual rendering constraint enables fast, reliable qualitative analysis and sets a high standard for visual verification.

Chapter 7

Experimental Setup and Ablation Studies

With the Stage 1 model completed and the rendering backend established, the focus of this work shifts to evaluating the generative performance of the Stage 2 model. This chapter presents a systematic ablation framework designed to evaluate how different contextual cues from the newly introduced Seamless Interaction dataset shape the quality, style, and natural rhythm of co-speech gestures.

Contributions 5: Empirical Ablation and Multi-Modal Contextual Integration

1. **Systematic Multi-Modal Feature Isolation:** Designing a four-stage progressive ablation study to isolate the impact of temporal cues, semantic embeddings, relational metadata, and partner-track feedback.
2. **Empirical Benchmarking on Seamless Interaction:** Establishing the first comprehensive baseline results (measuring realism, diversity, and rhythm) on this newly released dataset.
3. **Monadic-to-Dyadic Complexity Comparison:** Providing a clear evaluation comparison by contrasting a simplified speaker-only (monadic) setup against the full two-way conversational (dyadic) interactive loop.

7.1 Motivation and Experimental Scope

The main goal of these ablation studies is to evaluate how the multi-modal metadata in the new Seamless Interaction dataset affects gesture generation. While traditional benchmarks rely almost entirely on audio and a rigid Speaker-ID, this dataset includes personal metadata, relational profiles, and conversational timelines to enable more contextualized gestures. Because co-speech gesture generation is complex and lacks standard baselines, testing these features requires establishing a detailed evaluation framework.

This validation is important given the experimental nature of the proposed Stage 2 auto-regressive transformer. Although Chapter 5.1.7 showed that the dual-space optimization converges, numerical convergence does not guarantee natural or stable gestures during generation. These ablation studies serve as a direct test to see if the model actually uses the contextual metadata to improve gesture quality, or if some inputs just introduce noise.

This architecture is experimental, and the feature-fusion pipelines are an initial attempt rather than an exhaustive solution. Within this work, setting up a clear, functional baseline using direct multi-modal integration is prioritized. While more complex fusion methods, hyperparameter variations, and decoding algorithms warrant future study, this chapter focuses on testing the established setup. Benchmarking how these features affect physical metrics provides a solid foundation for future improvements.

7.2 Experimental Setup and Evaluation Framework

To validate the effectiveness of the proposed gesture generation pipeline, a empirical evaluation framework is established. This section outlines the quantitative metrics used to assess motion realism, temporal diversity, and rhythmic synchronization, followed by the specific structural configurations of the monadic and dyadic ablation studies.

Experiments 5: Monadic Baselines and Progressive Dyadic Ablations

E5.1 Monadic Baselines (Speaker-Only): Establishing foundational benchmarks for isolated speaker profiles (Baseline and BERT variants) to evaluate the core auto-regressive model in a simplified monadic setting.

E5.2 Model 1 (The Timing Foundation): Integrating discrete conversational Voice Activity Detection (VAD) states to act as temporal structural boundaries during dyadic generation.

E5.3 Model 2 (Semantic Grounding): Incorporating dense BERT semantic word embeddings to assess the transition from rhythm-driven beat gestures to context-specific, iconic motion.

E5.4 Model 3 (Continuous Relational Persona): Mapping dynamic interpersonal variables, specifically relationship indices and continuous emotional dimensions, to evaluate personalized motion style.

E5.5 Model 4 (The Full Dyadic Loop): Conditioning generation on partner audio and VAD streams to evaluate real-time interactive responsiveness and back-channeling behaviors.

7.3 Architectural Implementation of Ablation Models

To implement the effects of the multi-modal features, each ablated model modifies the core generative forward pass. Building upon the integration mechanisms (Cross-Attention and FiLM) established in Section 2.5, this section defines the specific formulations and temporal alignments required for each configuration.

The base auto-regressive transformer operates over a discrete motion sequence of $T' = 16$ latent tokens, corresponding to $T = 64$ raw kinematic frames representing approximately 2.13 seconds of motion.

7.3.1 Model 1: The Timing Foundation (Kinematic VAD Fusion)

Model 1 makes use of the VAD feature introduced in 3.2 to provide structural boundaries for speaking versus listening. The VAD state $V \in \{0, 1, 2\}^T$ is sampled at 30 FPS alongside the motion. To align V with the compressed timeline T', the state is down-sampled by taking the statistical mode across corresponding 4 frame chunk windows:

$$v_i = \text{mode}(V_{4i-3}, V_{4i-2}, V_{4i-1}, V_{4i}) \quad \text{for } i \in \{1, \dots, T\} \tag{7.1}$$

These discrete temporal states are projected through a learned embedding layer and added to the base motion embeddings prior to self-attention, fixing the generated motion to the conversational flow.

7.3.2 Model 2: Semantic Grounding and Timeline Alignment

Model 2 makes use of BERT semantic word embeddings $F_{\text{text}} \in \mathbb{R}^{T \times d_w}$. To ensure chronological synchronization between the semantic meaning of the words and the spoken acoustic beats in the cross-attention memory, the semantic timeline must be aligned with the convolutional audio features.

The 30 FPS semantic sequence is first upsampled to match the raw 50 Hz audio timeline ($T_{a\_raw} = 106$) using nearest-neighbor interpolation, maintaining exact physical timestamps. To match the audio convolution's center-alignment, the interpolated text is trimmed by an offset $c = (T_{a\_raw} - T_a)/2 = 24$. This produces a time-aligned semantic matrix $\hat{F}_{\text{text}} \in \mathbb{R}^{T_a \times d_w}$, which is concatenated with F_{audio} to form the combined cross-attention memory.

7.3.3 Model 3: Continuous Relational Persona

Model 3 evaluates the influence of dynamic variables on motion style. It takes static vectors for Personality (P) and Social Relationship (S), alongside a Emotion sequence $E \in \mathbb{R}^{T \times d_e}$. To create a stable underlying behavioral style for the given chunk, the emotional sequence is collapsed into a global state via the statistical mode:

$$e_{\text{global}} = \text{mode}(E) \tag{7.2}$$

This dominant emotion is concatenated with P and S, and projected into a continuous conditioning vector c. This vector acts as the stylistic prior injected into the network via the FiLM layers.

7.3.4 Model 4: The Full Dyadic Loop (Macro-Gating)

Model 4 extends the architecture to a full dyadic interaction by conditioning generation on both target audio $F_{\text{audio, target}}$ and partner audio $F_{\text{audio, partner}}$. To dynamically route the network's attention to the active speaker, a gating mechanism is implemented based on the VAD state.

The $T = 64$ frame VAD sequence is collapsed into a single dominant conversational state for the generative window:

$$v_{\text{chunk}} = \text{mode}(V) \tag{7.3}$$

This global state is projected through a fully connected layer and bounded by a sigmoid activation to obtain continuous volume multipliers $w_{\text{target}}, w_{\text{partner}} \in (0, 1)$. These weights are broadcast element-wise across the temporal dimension of the audio features:

$$F_{\text{fused}} = (w_{\text{target}} \odot F_{\text{audio, target}}) + (w_{\text{partner}} \odot F_{\text{audio, partner}}) \tag{7.4}$$

By gating the audio streams using a single combined chunk state, the model pushes partner audio into the background during active target speech, maintaining environmental context while establishing interactive responsiveness.

7.4 Quantitative Evaluation Metrics

Evaluating the auto-regressive transformer requires processing test sequences into motion clips of two lengths: 300 frames (10 seconds) and 900 frames (30 seconds). The 900 frame clips help analyze long-term physical stability and detect potential posture collapse over longer periods.

Each sequence is initialized with a warm-start, using the previous 64 ground-truth frames as history context. This priming helps the network establish initial joint velocities and postures before beginning open-loop generation.

Before calculating metrics, the generated 6D rotation matrices are mapped through the SMPL-H model to extract 3D coordinates for the 21 primary joints. The sequences are then evaluated on distribution realism, sequence diversity, and rhythmic synchronization.

7.4.1 Fréchet Distance (FD)

The overall realism and distribution match of the synthesized gestures relative to the ground-truth motion is measured via the Fréchet Distance (FD). Assuming the underlying motion features follow a multivariate Gaussian distribution, the metric computes the distance between the generated distribution (μ_p, Σ_p) and the reference ground-truth distribution (μ_g, Σ_g):

$$\text{FD} = \|\mu_p - \mu_g\|_2^2 + \text{Tr}\left(\Sigma_p + \Sigma_g - 2\sqrt{\Sigma_p \Sigma_g}\right) \tag{7.5}$$

This metric is calculated at two operational levels. **Static FD** evaluates the anatomical realism of individual postures by operating directly on the absolute 3D joint coordinates. **Kinematic FD** assesses temporal realism by computing the distribution distance over the first-order temporal derivative (joint velocity) across consecutive frames. Lower FD values indicate closer alignment with the natural human data distribution.

7.4.2 Motion Diversity

To ensure the auto-regressive transformer avoids mode collapse and retains physical expressiveness, motion diversity is calculated. This metric measures the average Euclidean distance between randomly sampled pairs of generated gesture sequences:

$$\text{Diversity} = \frac{1}{N} \sum_{i=1}^{N} \|x_i - y_i\|_2 \tag{7.6}$$

where x_i and y_i represent distinct, randomly selected motion sequences in coordinate space, and N represents the total number of evaluated pairs. Identical to the FD setup, **Static Diversity** measures the variety of individual frame postures, while **Kinematic Diversity** measures the variation in velocity and temporal expression. Higher values indicate a wider, more expressive generative range.

7.4.3 Beat Alignment

To quantify the rhythmic alignment between the generated motion and the acoustic speech track, the Gaussian Audio-Human Rhythm (GAHR) alignment score is calculated. Audio beats $a \in \mathcal{B}_a$ are extracted from the input speech using onset strength envelopes. Pose beats $p \in \mathcal{B}_p$ are identified as the local minima of the joint velocities for the left and right wrist coordinates. The GAHR score computes the Gaussian-smoothed temporal distance between each extracted audio beat and its temporally nearest gestural pose beat:

$$\text{Beat Alignment} = \frac{1}{|\mathcal{B}_a|} \sum_{a \in \mathcal{B}_a} \exp\left(-\frac{\min_{p \in \mathcal{B}_p} (p - a)^2}{2\sigma^2} \right) \tag{7.7}$$

where $\mathcal{B}_a$ is the set of extracted audio beat timestamps a with cardinality $|\mathcal{B}_a|$. $\mathcal{B}_p$ is the set of kinematic pose beat timestamps p identified via local wrist velocity minima. The term $(p - a)^2$ denotes the squared temporal distance between beats, and $\sigma = 0.1$ s is the Gaussian temporal tolerance parameter controlling the alignment decay rate.

A higher Beat Alignment score indicates that the model's generated physical gestures are synchronized with the rhythmic peaks of the spoken audio.

To keep the testing accurate and make sure all experimental runs can be compared fairly, all models are initialized with a fixed random seed of 42. The validation suite evaluates a test partition comprising 160 dyadic interaction sequences, matching the distribution split size established in the original *DyaDiT*[29] framework.

7.5 Internal Control and Benchmarking Limitations

Although the Seamless Interaction dataset is a groundbreaking step for conversational gestures, comparing it to previous research is difficult. While certain architectures (such as *DyaDiT* [29]) have published results on this dataset, their codebase, trained weights, and exact metric implementations remain private and unreleased. Consequently, running a contemporary SOTA benchmark model on the identical testing infrastructure was not possible within the scope of this work.

Therefore, analysis is limited to internal baselines to highlight how different multi-modal inputs influence performance metrics.

7.6 Quantitative Results and Comparative Analysis

this section presents the results of the gesture generation experiments. To isolate the effects of multi-modal features, the performance across both monadic (speaker-only) and dyadic (interactive) contexts is analyzed. Furthermore, comparing 300 frame (10 second) and 900 frame (30 second) generation clips evaluates the structural longevity of the auto-regressive transformer.

The quantitative results for the monadic benchmarks are presented in Table 7.1 and Table 7.2. The results for the dyadic ablated approaches are presented in Table 7.3 and Table 7.4.

Results for 300 frame clips (Monadic)					
Method	BC	FD		Diversity	
		Sta. ↓	Kine. ↓	Sta. ↑	Kine. ↑
Baseline Model	0.5277	4.1452	0.0048	1.9389	0.0801
Model 2 (Isolated Semantics)	0.5473	4.2511	0.0042	2.0001	0.0622

Table 7.1: Quantitative results for 300 frame (10 second) generated clips within a simplified monadic (speaker-only) context.

Results for 900 frame clips (Monadic)					
Method	BC	FD		Diversity	
		Sta. ↓	Kine. ↓	Sta. ↑	Kine. ↑
Baseline Model	0.6265	3.8609	0.0035	1.9541	0.0583
Model 2 (Isolated Semantics)	0.6426	3.8882	0.0034	2.0441	0.0474

Table 7.2: Quantitative results for 900 frame (30 second) generated clips within a simplified monadic context, assessing long-term kinematic stability.

Results for 300 frame clips (Dyadic)					
Method	BC	FD		Diversity	
		Sta. ↓	Kine. ↓	Sta. ↑	Kine. ↑
Baseline Model	0.5509	3.9570	0.0039	1.9224	0.0736
Model 1 (Timing Foundation)	0.5342	4.1132	0.0042	2.0442	0.0770
Model 2 (Isolated Semantics)	0.5502	3.8908	0.0040	1.8698	0.0548
Model 2 (Semantic Grounding)	0.5252	3.9010	0.0042	1.8555	0.0663
Model 3 (Relational Persona)	0.5192	4.5728	0.0046	2.1766	0.0798
Model 4 (Full Dyadic Loop)	0.5178	4.2374	0.0042	1.9866	0.0651

Table 7.3: Quantitative results for 300 frame (10 second) generated clips in a dyadic context, showing performance across models.

Results for 900 frame clips (Dyadic)					
Method	BC	FD		Diversity	
		Sta. ↓	Kine. ↓	Sta. ↑	Kine. ↑
Baseline Model	0.5952	3.8548	0.0027	1.9590	0.0657
Model 1 (Timing Foundation)	0.5724	3.6381	0.0030	2.0471	0.0669
Model 2 (Isolated Semantics)	0.6610	3.8518	0.0034	1.9015	0.0380
Model 2 (Semantic Grounding)	0.5806	3.5762	0.0031	1.8409	0.0554
Model 3 (Relational Persona)	0.5731	4.1082	0.0032	2.2090	0.0707
Model 4 (Full Dyadic Loop)	0.5707	3.7522	0.0031	1.9331	0.0515

Table 7.4: Quantitative results for 900 frame (30 second) generated clips in a dyadic context, highlighting long-term behavior.

Analysis of Monadic Configurations

As shown in Tables 7.1 and 7.2, the monadic configurations demonstrate stable baseline generation quality but generally perform slightly worse than their dyadic counterparts. For instance, the monadic baseline shows lower beat consistency and a higher Static FD compared to the dyadic baseline at 300 frames, suggesting that restricting the generative space to speaker-only features limits the model's capacity to anchor postures effectively without broader conversational context. Despite this, the architecture maintains consistent generation quality when moving from 10 second (300 frame) clips to extended 30 second (900 frame) sequences. This long-term stability indicates that the hybrid teacher-forced training strategy, introduced in Chapter 5. Successfully conditions the auto-regressive transformer to generate extended sequences without suffering from posture collapse when feeding back its own predictions.

An interesting pattern emerges when including textual features into this simplified space. Model 2 (Isolated Semantics) explicitly improves the BC score across both temporal horizons, rising from 0.5277 to 0.5473 at 300 frames and from 0.6265 to 0.6426 at 900 frames. This suggests that introducing semantic embeddings, provides contextual anchors that help the model synchronize its physical motions more tightly with spoken speech rhythms. However, these metric gains come with a distinct trade-off, as seen in the reduction of Kinematic Diversity, implying that semantic constraints guide the model toward more specific, structured motion paths at the expense of raw movement variance. To conclude, these monadic benchmarks prove that the underlying transformer network functions as a reliable motion generator, laying a stable foundation before scaling to multi-modal interactive environments.

Analysis of Progressive Dyadic Ablations

As shown in Tables 7.3 and 7.4, the dyadic experiments reveal that each individual multi-modal ablation distinctly alters the metrics, illustrating a clear trade-off between structural motion realism, diversity, and rhythm. As the feature space expands, the overall BC score experiences a mild compression, dropping slightly from 0.5509 in the baseline to 0.5178. Implying that the model naturally de-prioritizes strict, low-level acoustic beat synchronization to accommodate more complex semantic and interactive behaviors.

Model 2 (Isolated Semantics) achieves the strongest Static FD score (3.8908 at 300 frames) among the ablated models, demonstrating that aligning physical gestures directly with textual word meanings forces individual postures to stay closer to the real human data distribution, matching the semantic grounding trends noted in *SemTalk* [38]. Conversely, Model 3 (Relational Persona) introduces continuous emotional and social variables, which act as a powerful diversity amplifier. It secures the highest Static Diversity across both the 300 frame and 900 frame evaluations (2.1766 and 2.2090, respectively). While these stylized movements increase the Static FD score, Model 4 (The Full Dyadic Loop) reduces this score back down to 4.2374 at 300 frames and 3.7522 at 900 frames while maintaining good diversity. This shows that including partner interaction features helps stabilize the postures.

7.7 Chapter Discussion & Conclusion

In this chapter, the proposed Stage 2 auto-regressive transformer model was evaluated on the newly released Seamless Interaction dataset. Due to the novelty of this dataset and the lack of publicly available codebases from other models, a direct external comparison was unfeasible.

The progressive ablation studies confirm that the rich multi-modal metadata included within the Seamless Interaction dataset, for example conversational timeline states, semantic word embeddings, and relational personas, have a clear, trackable impact on movement patterns. Each ablated configuration introduced distinct, predictable behavioral trends, such as the semantic grounding observed in Model 2 and the diversity amplification driven by the relational and emotional variables in Model 3.

However, the quantitative results also reveal that integrating these distinct modalities is far from easy. The simplified fusion

methods evaluated in this experimental study. Relying primarily on direct projection, concatenation, and simple gating mechanisms, often introduced a trade-off. Where optimizing for semantic or stylized variety temporarily degraded static pose realism or rhythmic synchronization. Much like how the *SemTalk* [38] framework achieved superior results by introducing a specialized semantic gating mechanism rather than basic feature concatenation, future iterations of this architecture will require more sophisticated, modality-specific integration schemes to fully make use of the dataset's contextual variables without losing basic motion stability.

Chapter 8

Conclusion & Outlook

This work investigated the transition of co-speech gesture generation from static, offline architectures to a real-time, context-aware streaming approach. By employing a custom pre-processing pipeline, a unified upper-body representation, and a strictly causal auto-regressive generator, this framework moves closer to bridging the gap between raw conversational data and live virtual agents.

The main research question is resolved through a decoupled, two-stage generative pipeline optimized entirely from scratch. Stage 1 compresses high-dimensional continuous kinematics into a discrete, hierarchical multi-layer vocabulary. Stage 2 subsequently models these discrete motion tokens as a causal sequence, making use of a lightweight decoder-only transformer that generates gestures frame-by-frame with no temporal look-ahead. By using continuous style vectors (personality traits) and cross-modal audio features, the network shifts away from static speaker-IDs, presenting a more context-aware approach for co-speech gesture generation.

Answering Research Question 1: To answer the first research question, *how can raw dyadic audio-motion streams be systematically filtered and segmented to maintain pose accuracy while isolating high-intensity upper-body conversational behaviors?* An automated pre-processing pipeline was implemented to resolve the instability and tracking dropouts present in raw multi-vendor dyadic recordings.

This pipeline first applies out-of-screen filtering, programmatically checking 2D hand coordinates via COCO-Wholebody boundaries to drop frames with hand-pose clipping. To eliminate lower-body tracking drift, kinematic masking is applied to zero-mask the pelvis, thighs, knees, and feet of the SMPL-H model, freezing the lower body into a uniform stance. A motion intensity selection step then calculates hand velocity variance relative to a local 75th-percentile peak threshold, distilling the raw conversational pool into an expressive, noise-free subset. Finally, an anticipatory padding mechanism expands Voice Activity Detection (VAD) boundaries using bilateral temporal padding, ensuring the physical wind-up phases of gestures are cleanly captured. In combination, these processing steps ensure that the raw interaction data is properly filtered and fully optimized for subsequent model training.

Answering Research Question 2: To answer the second research question, *To what extent can a Hierarchical RVQ-VAE achieve accurate generalization when trained exclusively on a limited subset of high-intensity conversational motion?* The 4-layer RVQ-VAE was evaluated after training on the curated 13.1 hour high-motion subset. Despite being trained solely on active speaking gestures, the codebook successfully reconstructed subtle, passive listening postures, such as crossed arms or resting hands. Because the codebook only maps physically plausible joint configurations, the quantizer behaves as a natural denoiser, resolving severe joint warping and hand collapses during inference. Additionally, the unified model successfully compresses 64 frames of continuous upper-body rotations into a compact sequence of 16 temporal steps and 4 depth layers without requiring complex, separate sub-networks. Together, these properties prove that the Hierarchical RVQ-VAE establishes a robust and expressive spatial motion representation, making it fully prepared for downstream sequence modeling.

Answering Research Question 3: To answer the third research question. *How effectively can a decoder-only auto-regressive transformer, leveraging a multi-component loss strategy, model continuous co-speech gestures compared to traditional generative architectures?* The proposed causal, decoder-only transformer generates upper-body gesture tokens frame-by-frame. To prevent quadratic attention scaling, the main transformer predicts the structural Layer 1 token auto-regressively, while a decoupled cross-attention module iteratively generates the remaining detail layers. Furthermore, by pairing distance-aware soft cross-entropy with continuous latent MSE supervision. The model penalizes categorical prediction errors proportionally to their physical coordinate distance, mitigating exposure bias during training. Long-term sequence generation is stabilized by making use of scheduled sampling, which successfully lowers the effective perplexity search space for the structural base posture to roughly 5 to 8 equivalent options at any given frame. Collectively, these architectural and optimization choices prove that the auto-regressive transformer can model continuous gestures with stability, establishing it as a capable alternative to traditional offline, non-causal generative architectures.

Answering Research Question 4: To answer the fourth research question. *to what extent does integrating partner audio and continuous personality vectors enhance the dyadic context-awareness of the generated gestures?* The integration of partner feedback and relational traits was systematically evaluated, revealing clear behavioral modulations alongside distinct modeling trade-offs.

Making use of continuous speaker personality traits and partner VAD states enables the transformer to adapt overall gesture frequency, and energy during active-speaking and passive-listening states. Empirically, Model 2 (Isolated Semantics) achieves the best Static FD score, proving that aligning gestures directly to dense linguistic meaning keeps the generated postures closest to the data distribution, which directly aligns with the semantic grounding trends established in SemTalk [38]. Meanwhile, Model 3 (Relational Persona) obtains the highest Static Diversity, showing that incorporating dynamic social and emotional profiles acts as a diversity amplifier that prevents gestural repetition. These baseline results indicate that using the newly introduced Seamless Interaction features provides a viable starting point for context-aware, two-way interactions.

8.1 Contributions & Limitations of the Study

While the proposed pipeline achieves reliable real-time streaming, several technical limitations must be acknowledged. First, the spatial scope of motion generation is strictly constrained to the upper body and hands, excluding facial expressions and lower-body movement. Second, due to computational and processing boundaries, the models were trained and evaluated on concentrated subsets rather than the full 4000-hour Seamless Interaction corpus, restricting the network from potentially capturing more rare or complex social behaviors. Third, the generative output remains sensitive to static inference hyperparameters (θ, Top-K, Top-P), creating risks of stylistic drift if poorly configured. Fourth, the implemented methods for incorporating partner and personality metadata represent simplified, direct fusion strategies that may be too basic to fully capture the complex, non-linear feedback loops of live human interaction. Finally, the novelty of the dataset and the lack of open-source baselines highlight the need for future benchmarking against external models to truly evaluate how well the proposed Stage 2 model performs.

Contributions 6: The Summary

1. **Reproducible Data Curation Framework:** A pipeline bridging the gap between raw, un-mined dyadic interaction corpora and downstream gesture-generation modeling.
2. **Multi-Vendor Error Removal Protocol:** An automated filtering system that flags and isolates tracking dropouts and positional occlusions to guarantee upper-body and hand-pose accuracy.
3. **Role-Adaptive Temporal Segmentation:** A dynamic pipeline designed to isolate specific dyadic conversational states (active speaking, listening, or joint interactions), paired with Voice Activity Detection (VAD) padding to capture anticipatory physical wind-up phases.
4. **Custom Stage 1 Motion Reconstruction:** A self-contained motion reconstruction model built and trained from scratch, overcoming the SMPL-X format incompatibility of existing benchmarks to establish a robust SMPL-H representation.
5. **Consolidated Upper-Body Architecture:** A single Stage 1 model representing the torso, arms, and hands together, avoiding the training complexity and synchronization issues of separate sub-networks.
6. **Multi-State Generalization Evaluation:** A systematic assessment of the representation's generalization capacity across both active speaking and passive listening states using a compressed, high-motion 13.1 hour training subset.
7. **Causal Auto-Regressive Framework:** A sequence-generation architecture shifting gesture synthesis away from static, offline clip generation toward a live-streaming framework by treating discrete motion primitives as a linguistic vocabulary within a causal transformer.
8. **Large-Scale Optimization from Scratch:** An empirical training process leveraging the scale of the new dyadic Seamless Interaction dataset to optimize a data-hungry, decoder-only sequence model from scratch, bypassing traditional data scarcity limitations.
9. **Progressive Structural Foundation:** A clear developmental narrative addressing core streaming and convergence hurdles first, establishing a stable architectural foundation before analyzing multi-modal context variables in downstream ablation studies.
10. **Interactive Web Application:** A interactive web interface designed to replace manual animation suite workflows and accelerate visual evaluation loops.
11. **Cross-Dataset Unified Rendering Support:** A standardized visual rendering pipeline capable of parsing both SMPL-H and SMPL-X topologies, enabling unified evaluation across different benchmark datasets.
12. **Multi-Stage Pipeline Diagnostic Interfaces:** Interactive debugging tabs designed to verify raw database entries, check autoencoder reconstructions, and test interactive real-time sampling.
13. **Systematic Multi-Modal Feature Isolation:** A four-stage progressive ablation study designed to isolate the impact of temporal cues, semantic embeddings, relational metadata, and partner-track feedback.
14. **Empirical Benchmarking on Seamless Interaction:** The first comprehensive baseline results (measuring realism, diversity, and rhythm) established on this newly released dataset.
15. **Monadic-to-Dyadic Complexity Comparison:** A detailed evaluation comparing and contrasting a simplified speaker-only (monadic) setup against the full two-way conversational (dyadic) interactive loop.

8.2 Outlook & Research Directions

Several promising directions remain to extend this framework beyond its current baseline performance. First, future work could explore going beyond just upper-body movement by including facial expressions and full-body kinematics through the SMPL-X and FLAME models. Another proposed approach could investigate dynamically scaling the Stage 2 cross-entropy loss weights based on quantization depth to stabilize the coarse base trajectory (Layer 1) before learning unpredictable, fine-grained finger movements (Layers 2-4). Furthermore, scaling the pipeline to a larger subset of the Seamless Interaction dataset could help the causal model learn more rare social behaviors and subtle gestures. To improve dyadic context integration, future work could

also design advanced fusion methods to better balance the rich metadata provided by the Seamless Interaction dataset with the speaker's audio stream. Finally, the proposed model could be evaluated directly against external baseline models from other papers once their codebases are publicly released, allowing for a proper comparative test of its capabilities.

Bibliography

[1] Merveilles Agbeti-Messan, Pierrick Tranouez, Stéphane Nicolas, Clément Chatelain, and Thierry Paquet. Scaling state-space models from lines to paragraphs: An ablation of mamba-based ocr. *arXiv preprint arXiv:2606.23524*, 2026.

[2] Vasu Agrawal, Akinniyi Akinyemi, Kathryn Alvero, Morteza Behrooz, Julia Buffalini, Fabio Maria Carlucci, Joy Chen, Junming Chen, Zhang Chen, Shiyang Cheng, et al. Seamless interaction: Dyadic audiovisual motion modeling and large-scale dataset. *arXiv preprint arXiv:2506.22554*, 2025.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[4] Lucien Brown, Hyunji Kim, Iris Hübscher, and Bodo Winter. Gestures are modulated by social context: A study of multimodal politeness across two cultures. *Gesture*, 21(2-3):167–200, 2022.

[5] Tanya L Chartrand and John A Bargh. The chameleon effect: The perception–behavior link and social interaction. *Journal of personality and social psychology*, 76(6):893, 1999.

[6] Lipeng Cui and Jiarui Liu. Virtual human: A comprehensive survey on academic and applications. *IEEe Access*, 11:123830–123845, 2023.

[7] Elisa De Stefani and Doriana De Marco. Language, gesture, and emotional communication: An embodied view of social interaction. *Frontiers in psychology*, 10:2063, 2019.

[8] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

[9] Ylva Ferstl, Michael Neff, and Rachel McDonnell. Multi-objective adversarial gesture generation. In *Proceedings of the 12th ACM SIGGRAPH conference on motion, interaction and games*, pages 1–10, 2019.

[10] Nadia Mushtaq Gardazi, Ali Daud, Muhammad Kamran Malik, Amal Bukhari, Tariq Alsahfi, and Bader Alshemaimri. Bert applications in natural language processing: a review. *Artificial Intelligence Review*, 58(6):166, 2025.

[11] Judith Holler and Stephen C Levinson. Multimodal language processing in human communication. *Trends in cognitive sciences*, 23(8):639–652, 2019.

[12] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460, 2021.

[13] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *European Conference on Computer Vision*, pages 196–214. Springer, 2020.

[14] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11523–11532, 2022.

[15] 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):194–1, 2017.

[16] 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.

[17] 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.

[18] Lanmiao Liu, Esam Ghaleb, Asli Ozyurek, and Zerrin Yumak. Semges: Semantics-aware co-speech gesture generation using semantic coherence and relevance learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13963–13973, 2025.

[19] Lanmiao Liu, Esam Ghaleb, Aslı Özyürek, and Zerrin Yumak. Semconflow: Semantic grounding of holistic co-speech gesture generation with contrastive flow-matching. *arXiv preprint arXiv:2603.26553*, 2026.

[20] 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.

[21] 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.

[22] KD Locke. Interpersonal circumplex models and measures of personality. *EBSCO Pathways to Research in Psychology*, pages 1–9, 2023.

[23] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.

[24] Robert R McCrae and Paul T Costa Jr. A five-factor theory of personality. *Handbook of personality: Theory and research*, 2 (1999):139–153, 1999.

[25] M Hamza Mughal, Rishabh Dabral, Vera Demberg, and Christian Theobalt. Miburi: Towards expressive interactive gesture synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 40031–40041, 2026.

[26] Muhammad Hamza Mughal, Rishabh Dabral, Ikhsanul Habibie, Lucia Donatelli, Marc Habermann, and Christian Theobalt. Convofusion: Multi-modal conversational diffusion for co-speech gesture synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1388–1398, 2024.

[27] Evonne Ng, Javier Romero, Timur Bagautdinov, Shaojie Bai, Trevor Darrell, Angjoo Kanazawa, and Alexander Richard. From audio to photoreal embodiment: Synthesizing humans in conversations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1001–1010, 2024.

[28] 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.

[29] Yichen Peng, Jyun-Ting Song, Siyeol Jung, Ruofan Liu, Haiyang Liu, Xuangeng Chu, Ruicong Liu, Erwin Wu, Hideki Koike, Kris Kitani, et al. Dyadit: A multi-modal diffusion transformer for socially favorable dyadic gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10932–10942, 2026.

[30] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[32] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

[33] Muhammad Usama Saleem, Mayur Jagdishbhai Patel, Ekkasit Pinyoanuntapong, Zhongxing Qin, Li Yang, Hongfei Xue, Ahmed Helmy, Chen Chen, and Pu Wang. Livegesture: Streamable co-speech gesture generation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2264–2273, 2026.

[34] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

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

[36] Petra Wagner, Zofia Malisz, and Stefan Kopp. Gesture and speech in interaction: An overview, 2014.

[37] Dong Zhang, Zhaowei Li, Pengyu Wang, Xin Zhang, Yaqian Zhou, and Xipeng Qiu. Speechagents: Human-communication simulation with multi-modal multi-agent systems. *arXiv preprint arXiv:2401.03945*, 2024.

[38] 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.