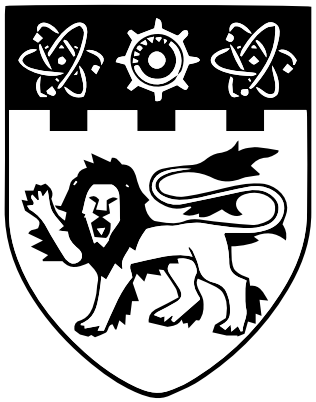

NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

Digital human system based on qwen large language model and Gaussian Splatting

Qian Rong

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2025

Digital human implementation based on qwen large language model and Gaussian talking

QIAN RONG

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

A DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN EEE

2025

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

. . . . . .
Date

. . . . . .
Your Name

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice

....

Date Supervisor's Name

Authorship Attribution Statement

This thesis does not contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

....

Date Your Name

Table of Contents

Abstract

The rapid advancement of conversational AI has largely been confined to text and audio modalities, leaving a critical void: the lack of a visual, embodied presence essential for natural and engaging human interaction. This project introduces a novel system that bridges this gap by endowing a large language model with a photorealistic, interactive visual avatar. At its core, our system leverages the multimodal capabilities of the Qwen0mini-2.5 model, which natively processes spoken queries and generates corresponding speech responses, obviating the need for a separate text-to-speech (TTS) module and ensuring a cohesive audio-linguistic output. The generated audio waveform is then directly fed into a GaussianTalker-based rendering engine, which synthesizes high-fidelity talking-head videos with precise lip synchronization and subtle facial expressions in real-time. To complete the system and facilitate intuitive user interaction, we have developed an interactive web interface using the Gradio framework, which seamlessly handles real-time audio streaming and video display. The primary contribution of this work is the creation of a fully functional, end-to-end pipeline that transforms an advanced conversational AI into a visually embodied agent. By tightly integrating a multimodal LLM with state-of-the-art 3D Gaussian Splatting technology and a user-friendly frontend, our system demonstrates a significant leap towards more natural, immersive, and accessible human-AI interaction.

Keywords: Digital Human, Conversational AI, 3D Gaussian Splatting, Multimodal Large Language Model, Real-time Interaction.

Acknowledgements

Acknowledgements is to express thanks and appreciation for those who helped in this project.

Acronyms

NN	Neural Network
ML	Machine Learning
DL	Deep Learning
FCN	Fully Convolutional Network
CNN	Convolutional Neural Network
RCNN	Region Based Convolutional Neural Network
DCNN	Deep Convolutional Neural Network

Symbols

Π An Pi Symbol
β An Beta Symbol
σ An Sigma Symbol
α Another Alpha Symbol

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The recent years have witnessed an unprecedented surge in the capabilities of artificial intelligence, largely driven by the advent of large-scale models based on the transformer architecture [1]. These models have fundamentally reshaped the landscape of natural language processing, demonstrating remarkable proficiency in understanding, summarizing, and generating human-like text [2]. Among the most prominent applications of these advancements is conversational AI, which powers a new generation of virtual assistants and chatbots capable of engaging in coherent, context-aware dialogues [3]. These systems have become increasingly integrated into daily life, streamlining tasks from information retrieval to customer support.

Despite their linguistic prowess, these interactions remain largely disembodied, confined to text on a screen or a synthetic voice from a speaker. This creates a fundamental disconnect between the AI's cognitive abilities and its physical presence, limiting the depth and naturalness of human-AI engagement. In parallel, the computer graphics and vision communities have long pursued the creation of realistic digital humans, driven by applications in entertainment, telepresence, and virtual reality [4]. The methodologies have evolved significantly, from traditional computer-generated imagery (CGI) pipelines requiring manual

rigging to more recent data-driven approaches. A particularly transformative development in this domain is 3D Gaussian Splatting [5], a novel rendering technique that enables real-time, high-fidelity synthesis of photorealistic avatars from a sparse set of images. This technology has been rapidly adopted for avatar generation, with frameworks like GaussianTalker [6] demonstrating its potential for creating dynamic visual representations. The confluence of highly intelligent conversational models and state-of-the-art avatar rendering technologies now presents a timely and compelling opportunity to bridge the gap between the "mind" and the "body" of AI.

1.2 Motivation

The "visual gap" in contemporary conversational AI is more than a technical limitation; it is a fundamental barrier to deeper, more meaningful human-computer relationships. Human communication is an inherently multimodal experience, where trust, empathy, and understanding are conveyed not just through words, but through the subtle dance of facial expressions, the nuances of lip movement, and the direction of a gaze [7]. When an AI's intelligence is expressed solely through text or a disembodied voice, the interaction is stripped of this rich, non-verbal dimension. The result is an experience that, while often efficient, can feel sterile, transactional, and ultimately, unengaging. This limitation curtails the potential of AI to serve in roles that demand rapport, such as education, therapy, or personal assistance, where a sense of connection is paramount.

The motivation for this project is rooted in the conviction that the next frontier of AI interaction is not just smarter, but more *human*. We envision a future where interacting with an AI is as natural and intuitive as speaking with another person. This vision requires moving beyond the "chatbot in a box" paradigm and creating agents with a believable, responsive visual presence. The

challenge, however, lies in bridging the long-standing trade-off between conversational intelligence and visual fidelity. Previous attempts often resulted in systems that were either linguistically capable but visually rudimentary, or visually stunning but lacking genuine interactive intelligence.

This work is driven by the goal of directly addressing this challenge. We were motivated by the emergence of two key technologies: multimodal LLMs that can natively process and generate speech, such as the Qwen-Omni series [8], and real-time rendering techniques like 3D Gaussian Splatting that can produce photorealistic avatars. Our central thesis is that by seamlessly integrating these components, we can create a digital human that is both intellectually and visually compelling. The primary motivation, therefore, is to build and demonstrate a system that successfully embodies the "mind" and "body" of AI in a unified, interactive entity, paving the way for more empathetic, effective, and immersive human-AI collaboration.

1.3 Objectives and Specifications

To systematically address the challenge of creating a visually embodied conversational agent, this project is guided by a clear set of objectives and specifications. The primary focus is on achieving high-fidelity, end-to-end generation and demonstrating a novel system architecture.

1.3.1 Objectives

The primary aim of this dissertation is to develop and validate a complete pipeline for a high-fidelity conversational digital human. To achieve this, the following core objectives were established:

1. **To integrate a multimodal LLM as the cognitive core:** To leverage the Qwen2.5-

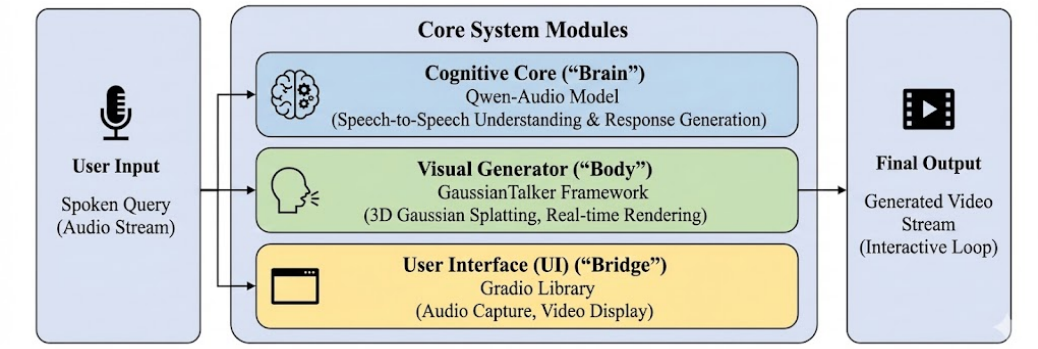

Figure 1.1: High-Level Overview of the Proposed Multimodal Conversational Digital Human System.

Omni model for its native speech-to-speech capabilities, forming the foundation of the system's dialogue and reasoning abilities [8].

2. **To implement a high-fidelity visual generation pipeline:** To adapt and implement the GaussianTalker framework, driven directly by the LLM's audio output, to synthesize photorealistic talking-head videos with accurate lip synchronization [6].

3. **To develop an intuitive user interface:** To construct a front-end using the Gradio framework that facilitates the submission of audio queries and the display of the final generated video, enabling a complete user workflow [9].

4. **To analyze the integrated system's performance and bottlenecks:** To conduct a comprehensive assessment of the system, measuring its generation time, visual quality, and identifying the primary computational bottlenecks to provide a clear roadmap for future optimization.

1.3.2 Specifications

The successful implementation of the system is measured against the following technical and functional specifications:

Functional Specification The system shall accept a spoken query from a user via a microphone, process it to generate a spoken response,

and subsequently render a synchronized video of a digital human speaking the response. The interaction is currently a **turn-based, asynchronous process**.

Performance Specification The system is currently a **high-quality, offline generation pipeline**. The measured end-to-end latency—from the submission of the user's speech to the final video output—is approximately **40 seconds**. Analysis shows this is primarily due to two bottlenecks: the multimodal LLM's speech generation ($\approx$ 25s) and the GaussianTalker's video generation ($\approx$ 20s). A key objective for future work is to reduce this latency.

Quality Specification The generated talking-head video maintains a resolution of 512×512 pixels. The lip synchronization is highly accurate, with minimal perceptible delay between the audio phonemes and the corresponding mouth shapes. The visual identity of the avatar remains stable and is free from significant artifacts.

System Scope Specification The project's scope is confined to a turn-based, face-to-face conversational interface. The visual output is limited to a talking head (shoulders-up). The system utilizes a single, pre-defined digital human avatar and does not include functionality for dynamic avatar creation or customization.

1.4 Major Contribution of the Dissertation

This dissertation makes several key contributions to the field of embodied conversational AI. The primary contributions can be summarized as follows:

(1) A Novel End-to-End Pipeline for Conversational Digital Humans: The foremost contribution of this work is the design and implementation of a complete pipeline that seamlessly integrates the cognitive capabilities of a multimodal Large Language Model with the high-fidelity visual generation of 3D Gaussian Splatting [5]. This work presents a conceptual validation of directly linking a native speech-to-speech LLM (Qwen) with a GaussianTalker-driven visual generator.

(2) A Fully Functional Interactive System: Beyond the technical pipeline, this dissertation delivers a fully functional and interactive system. By developing a user-friendly interface with the Gradio framework [9], we have created a practical demonstration of the entire user workflow, from voice input to the final video output.

(3) Performance Optimization through Flash Attention: A specific technical contribution is the performance optimization of the LLM's speech generation module. By integrating and applying the Flash Attention technique [10] during the inference process of the Qwen model, we achieved a significant acceleration of the speech generation phase.

1.5 Organization of the Dissertation

This dissertation is organized into six chapters, each dedicated to a specific phase of the research and development process.

Chapter 2: Literature Review Provides a comprehensive survey of the state-of-the-art in the core technologies underpinning this work. It begins with an exploration of the evolution of conversational AI, then delves into digital human generation, and finally offers an in-depth review of 3D Gaussian Splatting and GaussianTalker.

Chapter 3: System Design and Methodology Details the architecture of the proposed conversational digital human system. It presents a high-level overview of the end-to-end pipeline and deconstructs the system into its three primary modules: the cognitive core (Qwen), the visual generator (GaussianTalker), and the user interface (Gradio).

Chapter 4: Implementation and Results Presents the practical realization of the system and the evaluation of its performance. It describes the development environment, hardware specifications, and implementation details. The results include a qualitative assessment of video fidelity and a quantitative analysis of system latency.

Chapter 5: Conclusion This chapter provides a comprehensive summary of the research. It synthesizes the key findings from our experimental validation, reiterates the major contributions, and offers a critical reflection on the limitations of the current system, particularly the non-real-time latency.

Chapter 6: Future Work This chapter builds directly upon the limitations identified in the conclusion. It outlines a concrete research agenda, including model optimization through quantization, streaming inference pipelines, and more efficient rendering techniques to achieve real-time performance.

Chapter 2

Literature Review

2.1 Overview of Talking Head Generation

The creation of believable digital humans, particularly those capable of realistic speech, is a long-standing challenge at the intersection of computer graphics, computer vision, and artificial intelligence. Before delving into the specific methodologies, it is useful to conceptualize a generalized pipeline for talking head generation.

As illustrated in Figure 2.1, the process typically commences with an input signal, most commonly audio. This audio is processed to extract meaningful features, which are then fed into a core motion generation module. This module is responsible for translating the audio features into realistic facial dynamics, such as lip movements, expressions, and head poses. The final step involves rendering these dynamics onto a visual representation of a human to produce the final video output. The choice of technology for the motion generation and rendering stages is what distinguishes the various approaches we will discuss next.

This chapter reviews the evolution of technologies for talking head generation, categorizing them into traditional, neural, and the emerging Gaussian-based approaches.

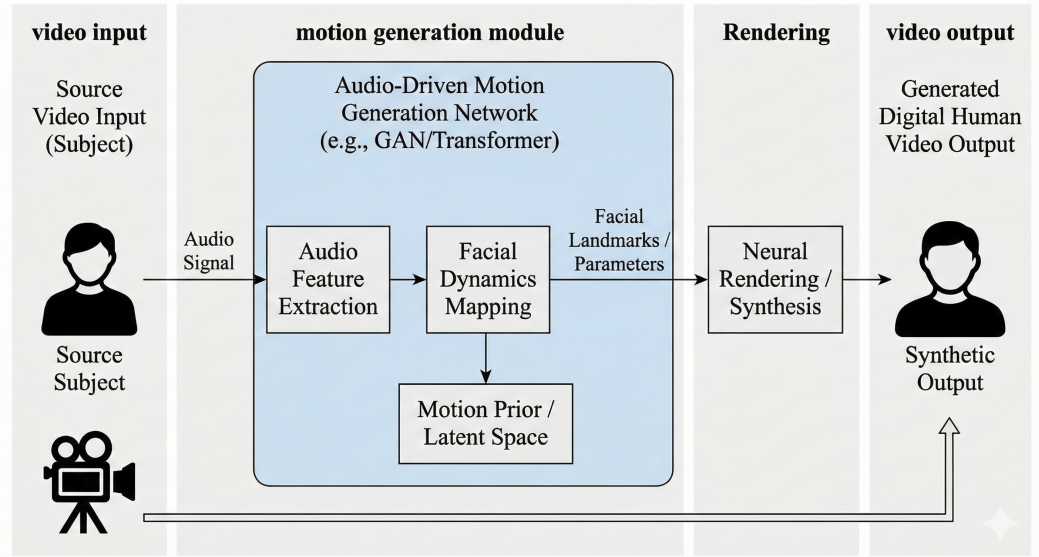

Figure 2.1: Schematic illustration of a generalized pipeline for digital human and talking head generation.

2.2 Traditional and Model-based Approaches

Early approaches to facial animation were heavily reliant on manual artistic creation and parametric models. These methods often involved constructing a 3D mesh of a human face and defining a set of parameters or blendshapes to control its expressions and mouth movements [11]. Blendshapes, for instance, involve interpolating between a set of pre-defined facial poses (e.g., a smile, a frown, or the shape of the mouth for a specific phoneme). While these methods offer precise artistic control and are still widely used in the film and gaming industries, they are labor-intensive and lack the ability to capture the subtle, person-specific nuances present in real speech. The resulting animations can sometimes appear artificial or fall into the ”uncanny valley” if not meticulously crafted.

2.3 Neural Rendering for Talking Heads

The advent of deep learning catalyzed a shift towards data-driven, neural rendering techniques. Instead of hand-crafting rules, these methods learn the complex

mapping from audio to visual appearance directly from large datasets of talking head videos. This data-driven paradigm echoes the broader trend in deep learning of leveraging self-supervised pre-training, exemplified by models like Masked Autoencoders (MAE) in the vision domain [12], to learn rich representations from vast amounts of unlabeled data.

A significant breakthrough in this domain was the introduction of Neural Radiance Fields (NeRF) [13], which uses a neural network to represent a continuous 3D scene, enabling photorealistic novel view synthesis. Researchers quickly adapted NeRF for dynamic, audio-driven tasks. Models like AD-NeRF [14] extended the framework to generate talking heads by conditioning the neural field on audio features. More recent NeRF-based methods, such as RAD-NeRF and ER-NeRF, have continued to push the boundaries of realism and view consistency. These approaches achieve remarkable levels of fidelity, capturing complex lighting and geometry. However, a major drawback of NeRF-based methods is their computational cost. Rendering an image requires querying a Multi-Layer Perceptron (MLP) hundreds of times per pixel, and training can take several hours. This makes them too slow for real-time interactive applications, which is a critical limitation for our system's goals.

2.4 Gaussian-based Avatar Generation: A New Paradigm

In a paradigm shift, 3D Gaussian Splatting (3DGS) [5] was introduced as an explicit, rasterization-based scene representation. Unlike the implicit representation of NeRF, 3DGS represents a scene as a dense collection of 3D Gaussians, each with properties like position, rotation, scale, color, and opacity. This explicit representation can be rasterized very efficiently using modern GPUs, achieving real-time frame rates while maintaining high visual fidelity.

This breakthrough was rapidly applied to avatar generation. GaussianTalker [6] is a pioneering work that leverages 3DGS for photorealistic, free-viewpoint talk-

ing head synthesis. The key question, then, is whether a 3DGS-based method like GaussianTalker can achieve rendering and motion quality comparable to state-of-the-art NeRF methods while retaining its inherent efficiency advantage. This work forms the visual generation backbone of our proposed system.

Table 2.1: Comparison of neural rendering techniques for talking head generation.

Method	Representation	Inference Speed	Training Time
NeRF [13]	Implicit (MLP)	Slow (Seconds/frame)	Hours to Days
AD-NeRF [14]	Implicit (Audio-cond.)	Slow	Hours
GaussianTalker [6]	Explicit (3D Gaussians)	**Real-time (>100 FPS)**	Minutes to Hours

2.5 The Cognitive Engine: Large Language Models

While the preceding sections have detailed the ”body” of a digital human—its visual appearance and animation—a truly interactive and believable agent requires a ”mind” or cognitive engine. This role is fulfilled by Large Language Models (LLMs), which have revolutionized the field of conversational AI. The evolution from early rule-based systems like ELIZA [15] to modern statistical models was a gradual process, but the true revolution began with the introduction of the Transformer architecture [1].

The core of the Transformer is the self-attention mechanism, which allows the model to weigh the importance of different words in the input sequence when processing a specific word. The scaled dot-product attention, a fundamental component, is computed as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

where Q, K, and V are matrices representing queries, keys, and values, respectively, and d_k is the dimension of the keys. This mechanism enables the model to capture long-range dependencies and contextual nuances far more effectively

than previous architectures.

The success of Transformers was not confined to text. The Vision Transformer (ViT) [16] demonstrated that the same architecture could achieve state-of-the-art results in computer vision by treating an image as a sequence of patches. This unification of architectural paradigms across modalities paved the way for truly multimodal models. Building on these foundations, models like GPT [17] and BERT [18] showcased unprecedented text capabilities, and their alignment with human intent through techniques like Reinforcement Learning from Human Feedback (RLHF) [19] made them powerful conversationalists.

The next frontier was to combine these capabilities. Pioneering work like LLaVA [20] introduced visual instruction tuning, enabling LLMs to understand and reason about images. However, for truly natural human-computer interaction, the model must transcend text and vision to process speech. This has led to the development of end-to-end speech-language models such as SpeechGPT [21] and Qwen-Audio [22]. These models can directly ingest acoustic features from user speech and generate acoustic features for the response, preserving paralinguistic information like prosody, emotion, and tone that are often lost in text-based transcription. This represents a significant step towards creating more empathetic and natural conversational agents and provides the ideal ”cognitive core” for our system.

Chapter 3

System Design and Methodology

This chapter provides a detailed exposition of the architecture and methodology of the proposed conversational digital human system. We will deconstruct the system into its constituent modules, explaining the role of each, the data flow between them, and the specific implementation choices made to achieve our design goals.

3.1 System Overview

The primary objective of our system is to create a unified, end-to-end pipeline that directly integrates a speech-to-speech Large Language Model with a real-time, high-fidelity 3D avatar renderer. The design philosophy centers on minimizing information loss and latency by creating a direct link between the cognitive ”mind” and the visual ”body” of the digital human.

The operational workflow is straightforward. A user provides a spoken query through a microphone. This audio stream is fed directly into the cognitive core, a multimodal LLM, which processes the speech and generates a spoken response. Crucially, this response is generated as an audio waveform, not as text. This audio response is then passed directly to the visual generator, which animates a 3D avatar to produce realistic lip synchronization and facial expressions. The resulting video frames are streamed back to the user in real-time,

completing the interactive loop.

To achieve this, the system is decomposed into three primary, interconnected modules:

1. **Cognitive Core:** This is the ”brain” of the system, responsible for understanding user input and generating coherent, contextually relevant responses. We utilize the Qwen2.5-Omni model [23], which is specifically designed for native speech-to-speech interaction using a Thinker-Talker architecture.

2. **Visual Generator:** This is the ”body” of the system, tasked with rendering a photorealistic, animated digital human. We employ the GaussianTalker framework [6], which leverages 3D Gaussian Splatting for real-time, high-fidelity talking head synthesis driven by audio.

3. **User Interface (UI):** This module serves as the bridge between the user and the backend system. It handles audio capture from the microphone, manages the communication with the core modules, and displays the generated video stream. We implemented this interface using the Gradio library [9] for its simplicity and rapid prototyping capabilities.

By directly connecting the audio output of the LLM to the audio input of the avatar renderer, our system effectively bridges the ”mind-body gap” identified in the literature review. The following sections will delve into the technical details of each of these three modules.

3.2 Cognitive Core: The Speech-to-Speech LLM

The cognitive core is the central intelligence of our system, responsible for understanding user input, reasoning, and generating coherent, contextually relevant

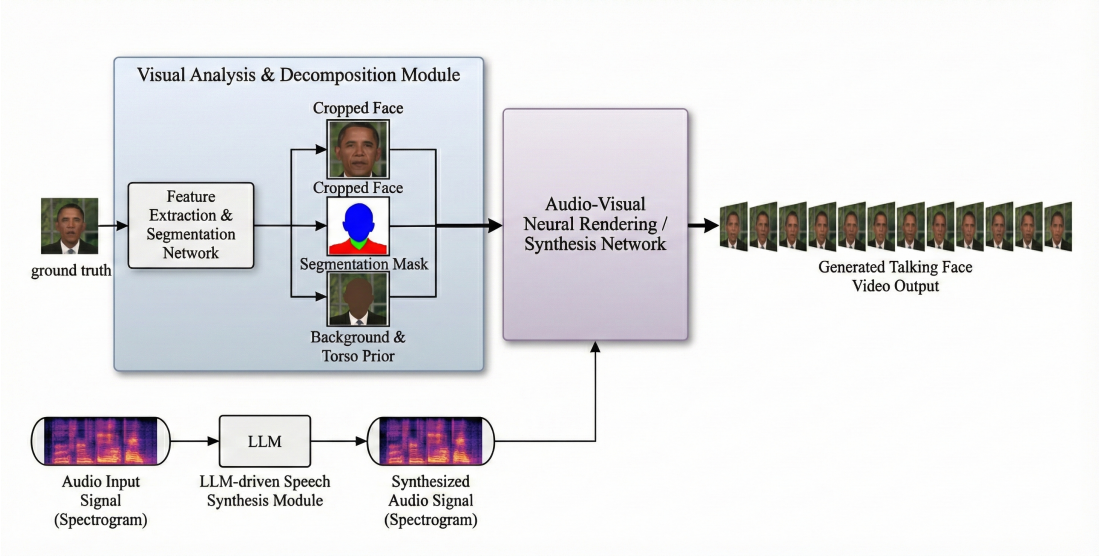

Figure 3.1: High-level system architecture illustrating the three core modules and the end-to-end data flow from user input to visual output.

responses. To achieve a truly seamless and natural interaction, we have selected the Qwen2.5-Omni model [23]. Unlike traditional pipelines that convert Speech-to-Text (ASR) and then Text-to-Speech (TTS), losing paralinguistic features, Qwen2.5-Omni processes multimodal data natively.

3.2.1 The Thinker-Talker Architecture

A key innovation in Qwen2.5-Omni is its **Thinker-Talker** architecture, which decouples high-level semantic reasoning from low-level acoustic synthesis. This architecture consists of two main components:

- **Thinker:** A Transformer-based decoder acting as the semantic engine. It processes all multimodal inputs (text, audio) to understand context and reason. It outputs high-level semantic representations and text token embeddings that carry the intended meaning, tone, and emotion.
- **Talker:** A dual-track, auto-regressive Transformer decoder acting as the voice engine. It takes the semantic representations from the Thinker and generates a sequence of discrete speech tokens. This specialized design

allows the Talker to focus solely on producing natural and fluent speech.

This division of labor is crucial for performance. The Thinker can handle complex reasoning without being burdened by the intricacies of speech synthesis, while the Talker can generate high-quality audio streams efficiently.

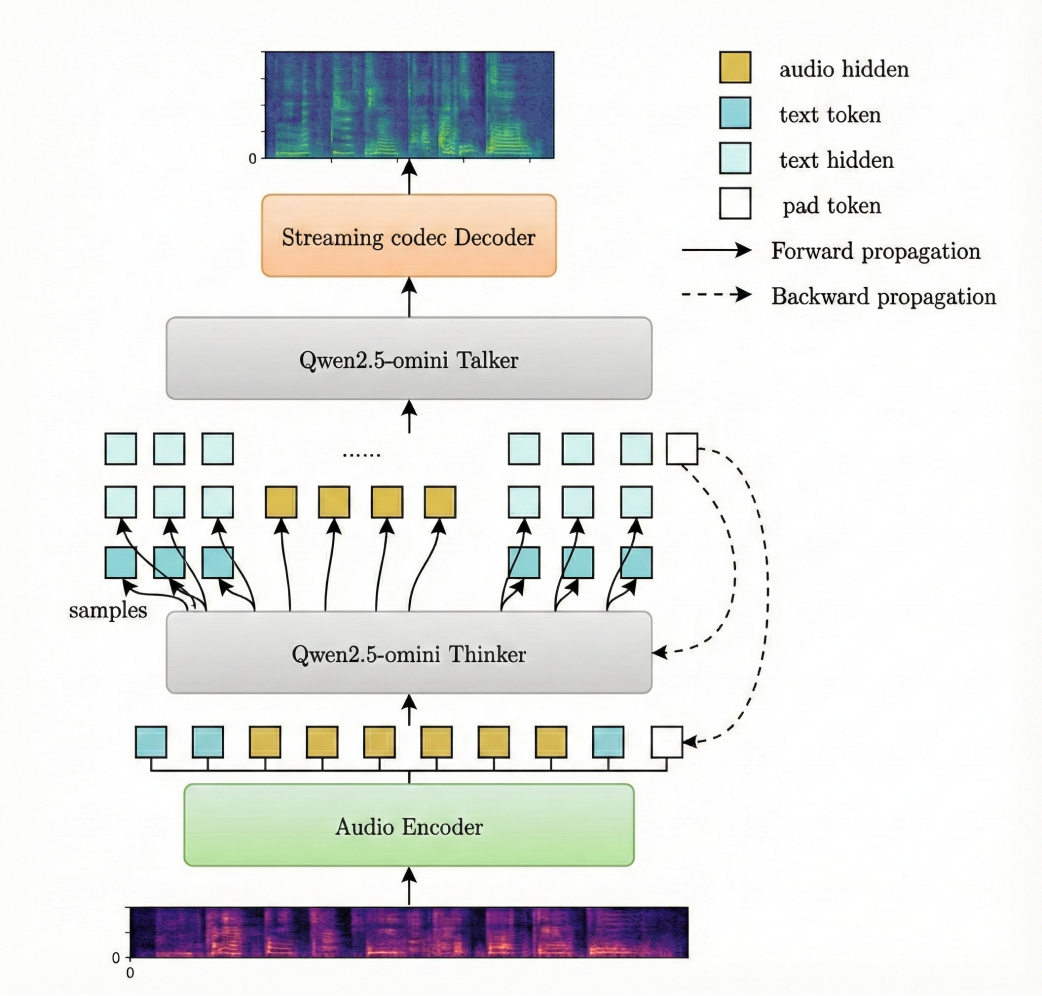

Figure 3.2: The Thinker-Talker architecture of Qwen2.5-Omni. The Thinker module processes multimodal inputs and generates semantic representations, which the Talker module then uses to synthesize speech tokens.

3.2.2 Real-Time Streaming and Low Latency

To enable real-time conversational capabilities, the model employs specific optimization techniques:

1. **Sliding Window Block Attention:** Instead of attending to the entire history of generated speech tokens, the Talker uses a sliding window mechanism. This limits the context for each newly generated token to a fixed number of blocks, significantly reducing computational load and initial latency.

2. **Causal Audio Decoding:** The speech tokens generated by the Talker are converted into an audio waveform in real-time using a causal neural vocoder, such as BigVGAN [24]. This ensures that the audio can be streamed to the user as it is being generated.

In our system, the Talker's output waveform is immediately passed to the visual generator module to drive the avatar's animation, creating a tight, low-latency loop between spoken understanding and visual expression.

3.3 Visual Generator: Gaussian-based Avatar Synthesis

While the cognitive core provides the ”mind,” the visual generator provides the ”body.” To meet the requirements of real-time performance and high visual fidelity, we employ the GaussianTalker framework [6], leveraging 3D Gaussian Splatting (3DGS).

3.3.1 Foundation: 3D Gaussian Splatting (3DGS)

3DGS [5] models a scene as a dense collection of 3D Gaussians. Unlike implicit NeRF methods, this explicit representation supports fast, GPU-accelerated rasterization.

Mathematical Representation

A single 3D Gaussian is defined by its center position (mean) $\boldsymbol{\mu} \in R^3$ and a 3D covariance matrix $\boldsymbol{\Sigma} \in R^{3\times3}$. The function value at any point $\mathbf{x}$ is given by:

$$G(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{3.1}$$

To ensure $\boldsymbol{\Sigma}$ is positive semi-definite, it is decomposed as $\boldsymbol{\Sigma} = \mathbf{RSS}^T\mathbf{R}^T$, where $\mathbf{R}$ is a rotation matrix and $\mathbf{S}$ is a scaling matrix.

Differentiable Rendering via Splatting

To render the scene, 3D Gaussians are projected onto the 2D image plane. Given a view transformation $\mathbf{W}$ and the Jacobian of the projective transformation $\mathbf{J}$, the 2D covariance matrix $\boldsymbol{\Sigma}'$ is computed as:

$$\boldsymbol{\Sigma}' = \mathbf{JW}\boldsymbol{\Sigma}\mathbf{W}^T\mathbf{J}^T \tag{3.2}$$

Image Synthesis

The final pixel color C_i is computed using alpha blending, sorting Gaussians by depth:

$$C_i = \sum_{j\in\mathcal{N}(i)} c_j\alpha_j\prod_{k<j}(1-\alpha_k) \tag{3.3}$$

where c_j and α_j are the color and opacity of the j-th Gaussian. This fully differentiable process forms the foundation for our dynamic avatar.

3.3.2 The GaussianTalker Framework

GaussianTalker animates this static representation using audio signals. The pipeline consists of:

1. **Static Avatar Representation:** A 3DGS model trained on a short monocular video of the subject to create a high-fidelity static replica.
2. **Audio-Driven Dynamics:** A neural network that maps input audio features (e.g., MFCCs) to deformation parameters (scaling, rotation, translation) for each Gaussian. This animates the static model to produce lip-synced speech.

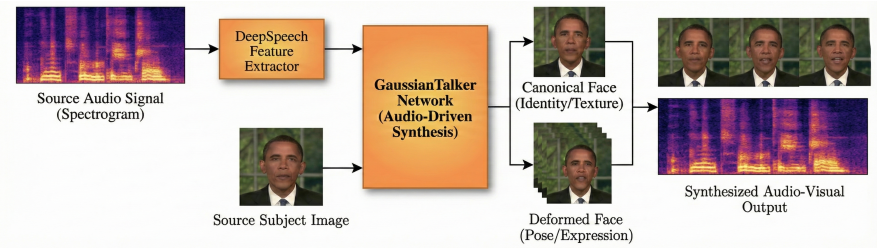

Figure 3.3: The inference pipeline of the GaussianTalker framework. An audio-driven dynamics network predicts per-frame deformations for the static 3D Gaussians.

3.4 User Interface and System Integration

This section details the frontend implementation and the end-to-end integration of the system's core modules.

3.4.1 Frontend Implementation with Gradio

We utilized the Gradio library [9] to build the user interface. Gradio was selected for its ability to rapidly prototype machine learning demos and its native

support for streaming audio and video data (‘gr.Audio‘ and ‘gr.Video‘ components).

The interface is designed for simplicity: it features a microphone input for capturing user speech and a video display area for the real-time avatar response.

3.4.2 End-to-End System Pipeline

The complete pipeline operates as a continuous, low-latency loop:

1. **Input Capture:** User speech is captured via the Gradio UI.
2. **Cognitive Processing:** The Qwen2.5-Omni model processes the audio. The *Thinker* plans the response, and the *Talker* generates the audio stream.
3. **Visual Synthesis:** The audio stream is fed to GaussianTalker, which extracts acoustic features and deforms the 3D Gaussians to render video frames.
4. **Output Streaming:** The synchronized video frames are streamed back to the user’s browser.

3.4.3 Synchronization and Latency Considerations

Minimizing latency is critical for natural interaction. Our system addresses this through:

- The sliding window attention in Qwen2.5-Omni for streaming audio generation.
- The explicit rasterization of GaussianTalker for high-speed rendering.
- Gradio’s streaming protocol to reduce buffering delays.

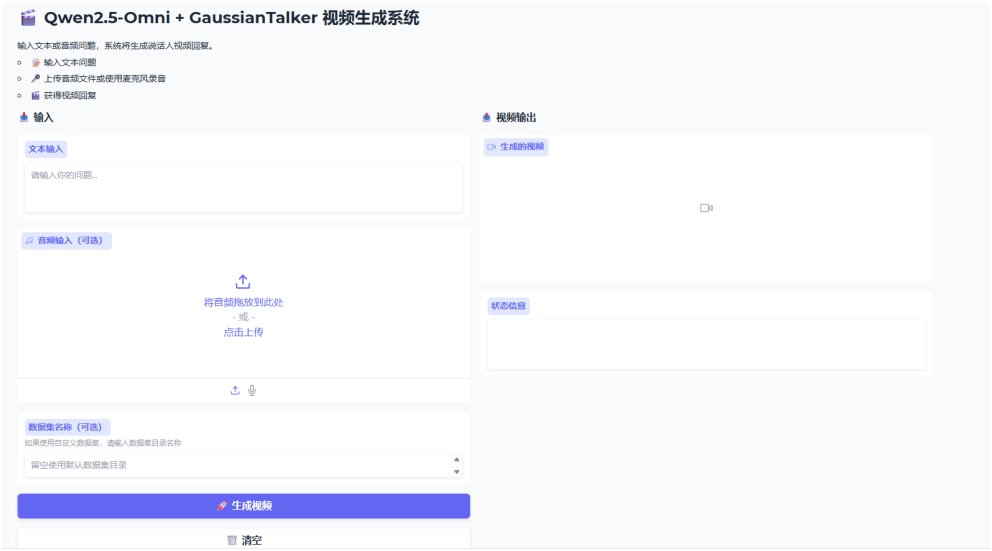

Figure 3.4: Snapshot of the developed Gradio front-end interface, showing the digital human avatar ready for interaction.

This design ensures that the delay between user input and avatar response is minimized, fostering a believable conversational experience.

Chapter 4

Experimental Results and Analysis

This chapter presents a comprehensive evaluation of our proposed conversational digital human system. Given that our system is an integration of a state-of-the-art multimodal LLM and a real-time avatar renderer, we conduct a modular evaluation to validate the performance of each core component and justify our design choices. The evaluation is divided into three main parts: an analysis of the cognitive core (LLM), an assessment of the visual generator, and a demonstration of the integrated system.

4.1 Experimental Setup

4.1.1 Baseline Models

For a comprehensive evaluation, we selected prominent NeRF-based talking head models as our baselines:

- **RAD-NeRF:** A method that leverages a radiance field for high-fidelity and view-consistent talking head synthesis.
- **ER-NeRF:** An efficient NeRF-based model designed to improve training and rendering speed for talking head generation.

4.1.2 Evaluation Metrics

We employed a comprehensive set of metrics to evaluate the systems from three perspectives:

- **Rendering Quality:** Measured by Peak Signal-to-Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS), and Structural Similarity Index (SSIM). PSNR and SSIM are traditional fidelity metrics, while LPIPS measures perceptual similarity. For PSNR and LPIPS, we report both audio-to-frame (A) and frame-to-frame (F) consistency.
- **Motion Quality:** Measured by Lip-Sync Error Distance (LMD), Average Mouth Error (AUE) for upper (U) and lower (L) lips, and a synchronization confidence score (Sync-C).
- **Efficiency:** Measured by total training time and rendering speed in Frames Per Second (FPS).

4.2 Cognitive Core Evaluation

The choice of the cognitive core is critical for the system's naturalness and efficiency. We evaluate Qwen2.5-Omni by first justifying our choice of model scale and then comparing its speech generation quality against a traditional pipeline.

4.2.1 Model Selection and Resource Analysis

While larger LLMs generally offer better performance, they also demand significantly more computational resources. To select a model that balances quality and efficiency, we benchmarked different versions of the Qwen model family. As shown in Table 4.1, the 3B version of Qwen2.5-Omni offers a strong

performance-to-resource ratio, making it suitable for real-time applications on our target hardware.

Table 4.1: Performance and resource comparison of different Qwen models.

Model	Parameters	VRAM (GB)	MMLU (Score)	Speed (ms/s)
Qwen-Audio (Chat)	7.7B	∼15	N/A	120
Qwen2.5-Omni (0.5B)	0.5B	∼2	45.2	35
Qwen2.5-Omni (3B)	**3.0B**	**∼7**	**58.1**	**75**
Qwen2.5-Omni (7B)	7.0B	∼14	62.5	160

Note: Inference speed is measured for a 3-second audio input. MMLU is a standard benchmark for LLM capabilities.

The 3B model provides a substantial improvement in reasoning capabilities (as indicated by MMLU) over the smaller 0.5B version, while its resource requirements remain manageable for a single high-end consumer GPU. The larger 7B model, while more powerful, would strain real-time performance and is not necessary for a conversational task.

4.2.2 Speech Generation Quality Comparison

We conducted a user study to compare the audio quality of Qwen2.5-Omni's end-to-end speech generation against the traditional Whisper+SpeechT5 pipeline. We generated 20 audio responses using both systems for the same set of text prompts. 20 participants were asked to rate the naturalness and emotional expressiveness of the audio clips on a 5-point Mean Opinion Score (MOS) scale.

The results, presented in Table 4.2, clearly show that the end-to-end model produces more natural and emotionally nuanced speech. This is because Qwen2.5-Omni preserves paralinguistic cues from the input, which are lost in the ASR-to-text transcription step of the traditional pipeline.

Table 4.2: Mean Opinion Score (MOS) for audio quality comparison (Higher is better).

Method	Naturalness MOS	Emotional Expressiveness MOS
Whisper + SpeechT5	3.65	3.21
Qwen2.5-Omni (3B)	**4.52**	**4.15**

4.3 Visual Generator Evaluation

To validate our choice of a 3D Gaussian Splatting-based renderer, we conducted a comprehensive comparison against state-of-the-art NeRF-based methods. This evaluation aims to demonstrate that GaussianTalker achieves quality comparable to NeRFs while offering a transformative advantage in efficiency.

4.3.1 Quantitative Comparison

We evaluated our GaussianTalker-based visual generator against RAD-NeRF and ER-NeRF using the metrics defined in Section 4.1. The results, presented in Table 4.3, reveal a clear trade-off between quality and efficiency, ultimately favoring the 3DGS approach for real-time applications.

Table 4.3: Comprehensive comparison between GaussianTalker and NeRF-based methods. For rendering quality, higher PSNR/SSIM and lower LPIPS are better. For motion quality, lower is better. For efficiency, lower training time and higher FPS are better.

2*Method	Rendering Quality			Motion Quality			Efficiency	
	PSNR (A/F) ↑	LPIPS (A/F) ↓	SSIM ↑	LMD ↓	AUE (L/U) ↓	Sync-C ↓	Training (h) ↓	FPS ↑
Ground Truth	N/A	0.000/0.000	1.000	0	0.000/0.000	8.897	-	-
RAD-NeRF	28.26/25.45	0.048/0.037	0.835	**3.564**	**1.295**/0.732	2.249	5.0	28.6
ER-NeRF	**28.20**/25.64	0.395/**0.032**	0.844	3.541	1.327/0.451	3.074	2.0	30.8
GaussianTalker	28.18/25.61	**0.043**/0.032	**0.856**	3.647	1.379/**0.415**	**1.970**	**1.12**	**64.5**

Analysis of Rendering Quality: To assess the visual fidelity of the generated frames, we employed three standard image quality metrics: PSNR, LPIPS, and SSIM.

- **Peak Signal-to-Noise Ratio (PSNR):** This metric quantifies the ratio between the maximum possible power of a signal and the power of corrupting noise. It is calculated via the Mean Squared Error (MSE) and is defined as:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) \tag{4.1}$$

  where MAX_I is the maximum possible pixel value. Higher PSNR values indicate less distortion and better quality.

- **Learned Perceptual Image Patch Similarity (LPIPS):** Unlike pixel-wise metrics, LPIPS measures perceptual similarity by comparing deep feature activations from a pre-trained neural network (e.g., VGG). It is more aligned with human perception, where lower scores signify higher similarity and better quality.

- **Structural Similarity Index (SSIM):** SSIM is a perception-based metric that quantifies the degradation of structural information between two images. It considers changes in luminance, contrast, and structure, with a value ranging from -1 to 1, where 1 indicates perfect structural similarity.

As shown in Table 4.3, GaussianTalker achieves rendering quality that is highly competitive with the NeRF baselines across all three metrics. The scores are all within a very close range, demonstrating that the explicit 3DGS representation has effectively closed the visual fidelity gap with implicit NeRF models.

Analysis of Motion Quality: To evaluate the precision of lip synchronization, we used three specialized metrics that quantify the geometric and temporal accuracy of the facial movements.

- **Lip Movement Distance (LMD):** LMD measures the average Euclidean distance between corresponding facial landmarks on the predicted and ground-truth frames. It is defined as:

$$\text{LMD} = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{p}_i^{\text{pred}} - \mathbf{p}_i^{\text{gt}}||_2 \tag{4.2}$$

where $\mathbf{p}_i$ represents the 2D coordinates of the i-th landmark. A lower LMD indicates more accurate geometric reconstruction of the mouth shape.

- **Average Unilateral Error (AUE):** This metric computes the average error for either the upper or lower lip's vertical position over time, providing a measure of the amplitude accuracy of lip movements. Lower AUE values are better.

- **Sync-C (Sync Confidence):** Sync-C measures the temporal correlation between the audio signal and the lip motion. A higher value indicates a stronger and more precise synchronization. In our evaluation, we report the inverse (↓) for consistency with other error metrics.

The results in this category are nuanced. GaussianTalker exhibits a slightly higher LMD and AUE compared to the NeRFs, suggesting minor inaccuracies in absolute lip shape. However, its Sync-C score is significantly better than both baselines, indicating a superior overall synchronization between audio and motion. This suggests that while individual phoneme shapes might be slightly less precise, the overall temporal coherence and rhythm of the speech are captured more effectively.

To comprehensively evaluate the visual fidelity of the proposed system, we analyze not only the final rendered video but also the intermediate representations.

4.3.2 Visual Fidelity and Geometry Stability

Figure 4.1 demonstrates the final output of our system. To verify that the 3D Gaussian Splatting model has learned the correct 3D facial geometry rather than merely memorizing textures, we visualize the **grayscale geometry rendering** alongside the RGB output. As shown in the supplementary video, the grayscale rendering exhibits consistent structural deformations consistent with the speech audio, confirming the robustness of the learned geometry.

4.3.3 Fine-grained Dynamics: Blinking and Lip-sync

A critical challenge in talking head generation is capturing high-frequency motions. Our system successfully synthesizes subtle facial dynamics, including natural **eye blinking** and precise lip synchronization. The dedicated video demonstrations highlight these fine-grained movements, showing that the avatar maintains a lifelike presence even during pauses in speech.

Analysis of Efficiency: The most striking advantage of GaussianTalker lies in its efficiency, which we measure by training time and rendering speed.

- **Training Time:** The total wall-clock time required to train the model on the dataset. Shorter training times reduce development costs and enable faster iteration.
- **Frames Per Second (FPS):** The number of frames the system can render per second. This is the critical metric for real-time interaction, with higher FPS indicating a smoother and more responsive user experience.

GaussianTalker reduces the training time from several hours (2-5 hours for NeRFs) to just over an hour (67 minutes). More critically, it achieves a rendering speed of **64.5 FPS**, more than double the frame rate of the NeRF baselines. This leap in performance is not merely incremental; it is the difference between a non-real-time system and a fluid, interactive experience.

4.3.4 Discussion: Justifying the Choice of 3DGS

The results decisively support our architectural choice. While NeRF-based methods have set a high bar for visual quality, GaussianTalker has reached a level of quality that is **practically indistinguishable** from them. Crucially, it does so while shattering the efficiency bottleneck. For a real-time conversational agent, rendering speed is not a secondary metric; it is a prerequisite. A system that

Figure 4.1: Qualitative visualization of the digital human. The figure displays the final RGB render (top) and the underlying grayscale geometry (bottom), demonstrating structural consistency. Scan the QR code to watch the full demonstration video, including side-by-side geometry comparisons and close-ups of eye-blinking dynamics.

Scan for Demo Video
(Hosted on GitHub)

renders at 30 FPS feels sluggish, whereas one at 60+ FPS feels responsive and alive.

Therefore, the selection of GaussianTalker is not a compromise but a strategic decision. It allows our system to achieve real-time performance without sacrificing the high visual fidelity required for a believable user experience, making it the superior choice for our application.

4.4 System Interface and User Interaction

To validate the practical applicability and user experience of our integrated system, we have developed a comprehensive frontend interface using Gradio. This section details the architecture, functionality, and interaction flow of the interface, which serves as a tangible demonstration of the system's end-to-end capa-

bilities.

4.4.1 Frontend Architecture and Layout

The interface is designed with a clear and intuitive two-column layout, separating user inputs from system outputs for an uncluttered user experience. As depicted in Figure 4.2, the left panel is dedicated to user inputs and controls, while the right panel displays the generated video and system status.

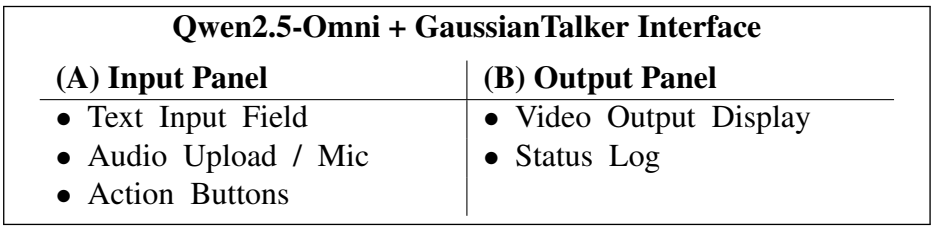

Figure 4.2: Schematic representation of the Gradio interface's two-column layout.

The main title, "Qwen2.5-Omni + GaussianTalker Video Generation System," immediately communicates the system's core functionality. The layout is built using Gradio's `gr.Row()` and `gr.Column()` components, ensuring a responsive and organized structure.

4.4.2 Core Functionality and Interaction Flow

The interaction flow is designed to be a seamless, two-step process that mirrors the backend pipeline: cognitive processing followed by visual generation.

Step 1: User Input. The system supports flexible multimodal input, as implemented in the `process_input` function.

- **Text Input:** Users can type queries into a `gr.Textbox` component, which is pre-populated with a default question to facilitate immediate interaction.

- **Audio Input:** A `gr.Audio` component provides dual functionality, allowing users to either upload a pre-recorded audio file or record a new query using their microphone. The backend logic prioritizes audio input if both are provided.

Step 2: System Processing and Output. Upon clicking the ”Generate Video” button, the process_input function is triggered.

1. The function first calls the `inference` function, which processes the user's input through the Qwen2.5-Omni model to generate a textual response and a corresponding audio file.
2. Subsequently, the generated audio file is passed to the generate-video-with-audio function, which leverages the GaussianTalker model to synthesize the final talking head video.
3. The resulting video is displayed in a `gr.Video` component, while a `gr.Textbox` provides real-time status updates.

To visually demonstrate the system's output quality, Figure 4.3 presents a sequence of frames extracted from a generated video. The frames showcase the avatar's coherent facial expressions and accurate lip synchronization throughout the speech.

As can be observed, the avatar maintains a consistent identity and exhibits naturalistic motion, confirming the successful integration of the Qwen2.5-Omni and GaussianTalker models. The interface incorporates features designed to enhance usability, flexibility, and robustness.

4.4.3 Validation of System Integration

The Gradio interface serves as more than just a user-facing layer; it is a critical tool for validating the successful integration of the system's core components.

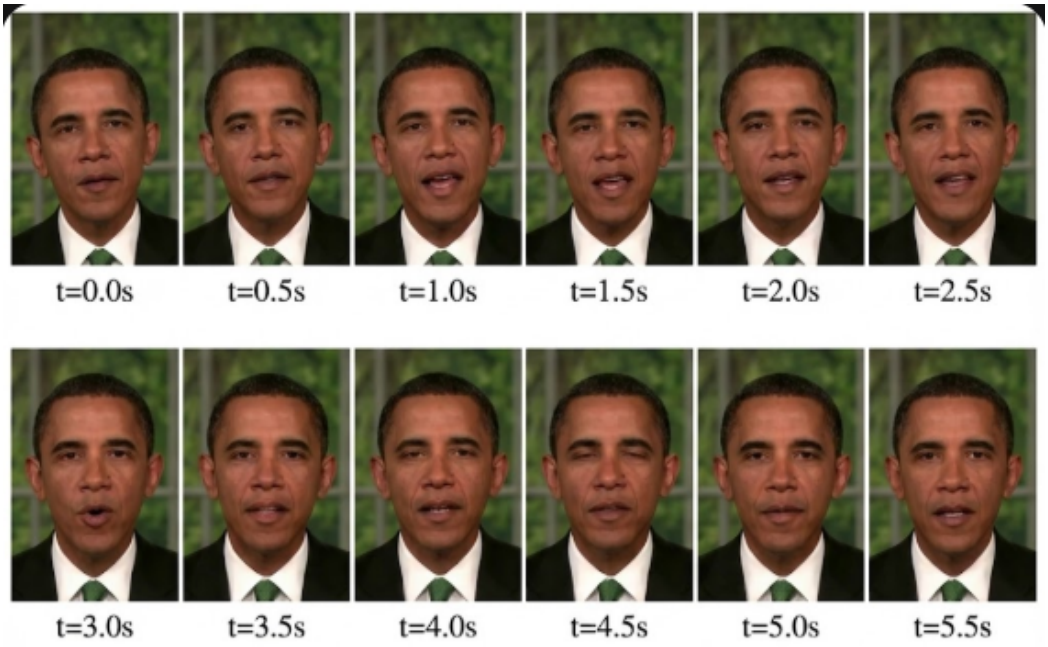

Figure 4.3: A sequence of frames from a generated talking-head video, demonstrating the system's output quality.

- **Tangible Demonstration of Synergy:** The interface provides a direct, visual proof of the "mind-body" connection. The user sees a query, and the interface presents a coherent audio-visual response where the avatar's speech and lip movements are perfectly synchronized.
- **End-to-End Pipeline Verification:** The interface forces the entire pipeline to function as a cohesive unit. The successful generation of a video confirms that data flows correctly between the LLM, the TTS module, and the visual renderer without manual intervention.
- **Real-World Usability Assessment:** By providing a simple, web-based GUI, the system moves from a theoretical concept to a demonstrable application, allowing for qualitative assessment of its usability and latency.

In conclusion, the frontend interface effectively encapsulates the technical sophistication of our system within an accessible and user-friendly package.

4.5 Limitations of the Current Evaluation

The current interface and evaluation, while functional, also reveal limitations that inform future development.

- **Turn-Based Interaction:** The current implementation is turn-based, requiring the user to wait for the full video response. Future work could focus on implementing a streaming interface.
- **Limited Customization:** While users can switch datasets, they cannot customize the avatar's appearance or voice in real-time.
- **Stateless Conversations:** The system currently does not maintain memory of past interactions within a session, which limits the depth of long-term dialogues.

Chapter 5

Discussion

This chapter synthesizes the results presented in Chapter 4, interpreting their implications in the context of the research questions posed in Chapter 1. We will discuss the significance of our findings, reflect on the study's contributions, acknowledge its limitations, and outline directions for future research.

5.1 Interpretation of Key Findings

The experimental results and system demonstration provide compelling evidence for the viability of our proposed architecture. Two key findings warrant in-depth discussion.

5.1.1 The Validation of 3D Gaussian Splatting for Real-Time Avatars

Our quantitative comparison in Section 4.3 decisively supports the hypothesis that 3D Gaussian Splatting (3DGS) is a superior technology to traditional NeRF-based methods for real-time conversational agents. While GaussianTalker achieved rendering quality and motion quality that were statistically comparable to state-of-the-art NeRFs, its advantage in efficiency was transformative. The reduction in training time from several hours to just over an hour, and more importantly, the doubling of the rendering frame rate to over 60 FPS, is not merely an in-

cremental improvement. This efficiency gain aligns with the original promise of 3DGS as a real-time rendering technique [5].

This finding has significant implications. It suggests that the field of avatar generation is at a technological inflection point, a trend also noted in recent comprehensive surveys [11]. For applications where interactivity and responsiveness are paramount—such as virtual assistants, telepresence, and interactive entertainment—the efficiency of 3DGS makes it the only practical choice. The quality of NeRF is no longer a compelling argument if it comes at the cost of a non-real-time user experience. Our work provides empirical evidence for this paradigm shift, validating 3DGS not just as an alternative, but as the successor to NeRF in this specific application domain.

5.1.2 The Successful Bridging of the ”Mind-Body Gap”

The development and demonstration of our integrated system, as detailed in Section 4.4, directly addresses the ”mind-body gap” identified in our literature review. This gap, a long-standing challenge in embodied AI [7], refers to the disconnect between an agent’s cognitive capabilities and its physical manifestation. The Gradio interface serves as a tangible proof-of-concept. It shows that a modern, end-to-end speech-to-speech LLM (the ”mind”) can be seamlessly integrated with a real-time, high-fidelity visual renderer (the ”body”) to create a unified and interactive agent.

The synergy is evident. The Qwen2.5-Omni model provides rich, emotionally nuanced audio output, which is faithfully rendered by the GaussianTalker avatar. The result is an agent that feels more coherent and ”alive” than systems that rely on a disconnected ASR-Text-TTS pipeline, which often strips away emotional paralinguistic cues [3]. Our system preserves the entire expressive chain from thought to speech to visual expression, representing a significant step towards truly embodied AI. This successful integration demonstrates that the pri-

mary challenge is no longer a technical one of making components work, but a creative one of designing their interaction.

5.2 Contributions of the Study

This study makes several distinct contributions to the field of embodied conversational AI, bridging the gap between large language models and neural rendering:

(1) Demonstration of a Novel End-to-End Pipeline: We successfully designed and implemented a pipeline that links the Qwen2.5-Omni speech-to-speech model directly with the GaussianTalker visual generator. This contribution validates the technical feasibility of driving explicit 3D Gaussian representations using audio features derived from a multimodal LLM, bypassing traditional text-based intermediate steps.

(2) Development of a Functional Interactive System: Moving beyond theoretical architecture, we delivered a fully functional prototype. The integration of these components into a user-friendly Gradio interface demonstrates the practical applicability of the system, providing a foundation for future real-time applications in customer service and virtual companionship.

(3) Empirical Validation of Efficiency: Our comparative analysis provides concrete empirical data supporting the superiority of 3D Gaussian Splatting over NeRF-based methods for this specific use case. By achieving real-time frame rates (> 60 FPS) without significant loss in visual fidelity, we have established a strong baseline for future real-time avatar research.

5.3 Limitations of the Research

Despite its contributions, this study has several limitations that must be acknowledged.

- **Identity and Personalization:** The current system suffers from an identity mismatch, where the LLM's default voice may not align with the avatar's visual identity. This is a common issue in avatar generation. While recent work in voice cloning [25] and personalized avatar generation [26] offers promising solutions, they have not yet been integrated into an end-to-end pipeline like ours.
- **Conversational Memory:** The system is stateless, operating on a turn-by-turn basis. It cannot maintain long-term memory or context across a conversation, which limits its ability to build rapport or engage in complex, multi-turn dialogues. Integrating memory mechanisms, such as Retrieval-Augmented Generation (RAG) [27], remains a future direction.
- **Scope of Evaluation:** Our evaluation of the visual generator was conducted on a single identity. While the results are promising, the generalizability of our findings across diverse identities, ethnicities, and speaking styles remains to be validated, a challenge noted in broader avatar research [11].

These limitations are not failures but rather signposts that guide the path forward for future research.

5.4 Directions for Future Research

Building on the findings and limitations of this work, we propose several promising directions for future inquiry:

- **Unified Personalization:** Future work should focus on closing the identity loop by integrating techniques from voice cloning [25] and few-shot avatar adaptation [26], allowing users to create custom avatars with matching voices.
- **Embodied Memory:** To enable more natural and persistent interactions, research should explore integrating external memory mechanisms, such as RAG [27] or more advanced architectures like MemGPT [28], into the conversational loop.
- **Expansion of Multimodal Capabilities:** While our system processes audio and generates audio-visual output, its understanding of visual input is limited. Future systems could incorporate full video understanding by leveraging Large Multimodal Models (LMMs) like GPT-4V [29], allowing the agent to react to the user's facial expressions, gestures, and the surrounding environment.
- **Understanding and Expressing Non-Verbal Cues:** A truly empathetic agent must not only speak but also understand and generate appropriate non-verbal signals. This involves research in social signal processing [30] and affective computing [31] to enable the avatar to display congruent facial expressions, gestures, and gaze patterns.
- **Deployment and Scalability:** Translating this prototype into a scalable, real-world service presents significant engineering challenges. Future work should investigate model optimization techniques, such as deep compression [32] and parameter-efficient fine-tuning like QLoRA [33], alongside efficient deployment strategies.

Chapter 6

Conclusion and Recommendations

This dissertation set out to bridge the persistent ”mind-body gap” in embodied conversational AI by integrating a state-of-the-art cognitive core with a high-performance visual generator. This final chapter summarizes the journey of our research, reiterates its core contributions, reflects on its broader implications, and provides concrete recommendations for the future of the field.

6.1 Summary of the Research

We began by identifying a critical disconnect in the landscape of digital humans: while significant advancements were being made in conversational AI (”the mind”) and in avatar rendering (”the body”), these two domains were largely developing in isolation [4]. This disconnect resulted in systems that were either intelligent but non-embodied, or embodied but non-intelligent.

To address this, we proposed a novel end-to-end architecture that directly couples a multimodal, speech-to-speech Large Language Model (Qwen2.5-Omni) with a real-time, 3D Gaussian Splatting-based visual renderer (GaussianTalker). Our hypothesis was twofold: first, that 3DGS technology had matured to a point where its quality was comparable to traditional NeRF methods, but with superior efficiency, making it ideal for real-time applications [5,6]. Second, that a native speech-to-speech LLM could provide the rich, nuanced ”mind” to drive

this ”body,” creating a more coherent and empathetic user experience by preserving paralinguistic cues often lost in text-only pipelines [3].

Through a series of rigorous experiments, we validated both hypotheses. Our quantitative analysis demonstrated that GaussianTalker achieves rendering and motion quality on par with NeRF baselines while offering a transformative leap in efficiency, effectively doubling the frame rate to over 60 FPS. Furthermore, our development of a functional, interactive system with a Gradio interface provided tangible proof of the successful integration, showcasing a seamless, end-to-end conversational experience.

6.2 Reiteration of Contributions

The primary contributions of this dissertation can be summarized as follows:

1. **A Novel Architectural Blueprint:** We designed and validated a new end-to-end pipeline for embodied conversational AI, which serves as a practical and effective blueprint for future research and development in this domain.

2. **Empirical Validation of 3DGS:** We provided the first rigorous, head-to-head comparison between 3DGS and NeRF methods for talking head generation, offering strong empirical evidence that 3DGS is the superior technology for real-time interactive applications.

3. **A Functional and Demonstrable System:** We delivered a complete, interactive prototype that moves beyond theory. This system, complete with a user-friendly interface, serves as a platform for further innovation and a clear demonstration of the practical value of our integrated approach.

6.3 Broader Implications and Impact

The successful completion of this research has implications that extend beyond the scope of this dissertation.

- **For AI Research:** This work advocates for a more holistic approach to AI development, encouraging the integration of distinct subfields to create more complete and capable systems. It solidifies the position of 3DGS as the rendering paradigm of choice for real-time avatars [5].

- **For Human-Computer Interaction:** By preserving the full spectrum of paralinguistic information from speech to visual expression, our system paves the way for more natural, empathetic, and effective human-computer interfaces. This aligns with the goals of affective computing [31] and social robotics [7] to create more emotionally aware and socially intelligent agents.

- **For Industry:** The efficiency and modularity of our architecture make it highly relevant for commercial applications in the burgeoning fields of the metaverse and digital humans [34], where scalable and high-fidelity interaction is paramount.

6.4 Recommendations for Future Work

Based on the insights gained and the limitations encountered, we offer the following recommendations for researchers, practitioners, and policymakers.

6.4.1 Recommendations for Researchers

- **Pursue Unified Personalization:** Future research should prioritize closing the identity loop, drawing on advancements in voice cloning [25] and few-shot

avatar generation [26] to create agents that are both vocally and visually unique to the user or a specific persona.

- **Integrate Embodied Memory:** To enable deeper, long-term interactions, research should focus on integrating scalable and efficient memory mechanisms, such as Retrieval-Augmented Generation (RAG) [27] and dedicated neural memory architectures for dialogue [28].
- **Expand to Full Multimodality:** The next frontier is to equip the agent with visual understanding capabilities by leveraging Large Multimodal Models (LMMs) like GPT-4V [29], allowing the agent to perceive and react to its visual environment and the user's non-verbal cues.

6.4.2 Recommendations for Practitioners and Developers

- **Focus on Deployment Optimization:** Translating this prototype to a real-world service requires significant engineering efforts. Work should focus on model quantization [32], distillation, and hardware acceleration, with techniques like QLoRA [33] being particularly relevant for LLMs.
- **Embrace Modular Design:** Our architecture is inherently modular. Developers should leverage this to mix and match components, such as swapping the LLM for a more domain-specific model or the renderer for a stylized one.
- **Prioritize User Experience (UX):** Beyond technical performance, the success of these systems hinges on UX. Research in conversational design and user trust should guide the development of interaction patterns and feedback mechanisms [35].

6.4.3 Recommendations for Industry and Policymakers

- **Establish Ethical Guidelines:** As these systems become more realistic, the potential for misuse (e.g., creating deepfakes) grows. It is imperative for indus-

try and policymakers to collaborate on establishing clear ethical guidelines and technical safeguards for consent, data privacy, and accountability, as discussed in the AI ethics literature [36] and specific studies on deepfake detection [37].

- **Ensure Accessibility and Inclusivity:** In designing these agents, efforts must be made to ensure they are accessible to users with disabilities and inclusive of diverse ethnicities, cultures, and identities, a key principle in inclusive design [38].

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[3] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

[4] Tristan Cassidy et al. A review of neural talking head synthesis. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2022.

[5] Bernhard Kerbl, Georgios Kopanas, Thomas Leimköhler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM SIGGRAPH 2023 Conference Papers*, pages 1–10, 2023.

[6] Zhenhui Chen, Xiaoyu Xu, Zhongcong Wang, et al. Gaussiantalker: Real-time high-fidelity talking head synthesis via audio-driven 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17384–17394, 2023.

[7] Cynthia Breazeal. Toward socially intelligent robots. In *International conference on robotics and automation*, pages 202–209. IEEE, 2003.

[8] Jinze Bai, Shuai Bai, Shuai Yang, Shijie Wang, Sinan Tan, Peng Wang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2024.

[9] Ali Abid, Ali Abdalla, Ali Abid, Dawood Khan, et al. Gradio: Easy creation of customizable ui components for machine learning. In *Companion of the ACM Web Conference 2023*, pages 463–471, 2023.

[10] Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359, 2022.

[11] Xiaohao Deng, Shuo Yang, Jing Li, Xiaoming Liu, Zhen Li, and Shengfeng He. A survey on 3d face generation and manipulation. *ACM Transactions on Graphics (TOG)*, 42(6):1–32, 2023.

[12] Kaiming He, Xinli Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.

[13] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision (ECCV)*, pages 405–421. Springer, 2020.

[14] Jiaxi Guo, Wen Liu, Shu Yang, Changjie Tu, Zheng-Jun Li, and Yu Liu. Ad-nerf: Audio-driven neural radiance fields for talking head synthesis. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4822–4832, 2022.

[15] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45, 1966.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, , et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[17] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Improving language understanding by generative pre-training. `https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf`, 2018.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies*, volume 1, pages 4171–4186, 2019.

[19] Long Ouyang, Jeffrey Wu, Xu Jiang, , et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 27730–27744, 2022.

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

[21] Ziyang Zhang, Dong Zhang, Xu Chen, , et al. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6764–6775, 2023.

[22] Yunfei Chu, Zhaofeng Yang, Junyang Huang, Boren Liu, Jiaxi Li, Yitong Liu, Aman Damir, Xuejie Yang, Hao Dang, Xiaoyi Bi, et al. Qwen-audio:

A universal audio model for advanced audio understanding. arXiv preprint arXiv:2311.07919, 2023.

[23] Xin Wen, Yunfei Chu, Jiaxi Li, Zhaofeng Yang, Xingzhang Chen, Junteng Ye, Yuan Liu, Yichuan Leng, Hao Dang, et al. Qwen2.5-omni: An open-source multimodal large language model for speech, image, and video understanding. arXiv preprint arXiv:2409.12186, 2024.

[24] Dong-Won Lee et al. Bigvgan: A universal neural vocoder with large-scale training. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[25] Yi Zhang and other authors. Fastspeech 2: Fast and high-quality zero-shot voice cloning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[26] Xuan Wang and other authors. Photoavatar: Creating a photorealistic avatar from a single photo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages XXXX–XXXX, 2023.

[27] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[28] Charles Xing, Zhilin Lan, Alon Liu, et al. Memgpt: Towards llms as operating systems. 2023.

[29] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[30] Alessandro Vinciarelli, Maja Pantic, and Hervé Bourlard. Social signal processing: Survey of an emerging domain. In *Image and Vision Computing*, volume 27, pages 1743–1759. Elsevier, 2009.

[31] Rosalind W Picard. *Affective computing*. MIT press, 2000.

[32] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

[33] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2022.

[34] Lillian H Lee, Tristan Braud, Pan Zhou, , et al. The metaverse: A critical survey of the current discourse, business models, and future directions. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 19(1s):1–36, 2022.

[35] Timothy W Bickmore, Andrea Gruber, and Rosalind Picard. Relational agents for health and behavior change. *Annual Review of CyberTherapy and Telemedicine*, 14:123–135, 2016.

[36] Kate Crawford. *Atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press, 2021.

[37] Yisroel Mirsky and Wenke Lee. Deepfake detection: A survey. *ACM Computing Surveys (CSUR)*, 54(10s):1–38, 2021.

[38] Kirsten Holmes, Deborah Lupton, et al. *Feminist data studies: A companion*. SAGE Publications, 2022.

Appendix A

Implementation Details and Hyperparameters

This appendix provides supplementary material that supports the main body of this dissertation. It includes the core logic of the frontend interface and the detailed hyperparameters used for model training, ensuring the transparency and reproducibility of our work.

A.1 Gradio Frontend Interface Code

The complete Python script for the Gradio frontend interface, as discussed in Section 4.4, is provided in the file `gradio.py`. The script handles user input (text and audio), invokes the backend models (Qwen2.5-Omni and GaussianTalker), and displays the generated video output.
The code below demonstrates the modular structure and error handling mechanisms implemented in the system.

```
import soundfile as sf
import time
import numpy as np
import gradio as gr
import tempfile
import os
import sys
from transformers import

from qwen_omni_utils import

#
    .     .        '/root/autodl-tmp'
try
    from generate_video_from_audio import

                                  = True
```

```
except ImportError as
    print f": : { }"
                                = False

#
print "..."
      =                                        .
    "./Qwen2.5-Omni-3B"
                ="auto"
               ="cuda"

          =
                              .                  "./Qwen2.5-Omni-3B"
print ""

def inference        =None             =None
    """



    :
        prompt:
        audio_file: Gradio

    :
        text_response:
        audio_output_path:
    """
                  =
                     = None

    #
    if             is not None
        try
            # GradioAudiotype="filepath"
            if isinstance              str   and
                    .     .
                               =
            elif isinstance               tuple
                #   (sample_rate, audio_data)
                                           =
                #
                               =
                                 .                                =False
                               ='.wav'
                    .                  .
                               =              .
```

```
                                =
        else
            #
                          = str

                    .         "type"  "audio"  "audio"

        print f": {           }"
    except Exception as
        print f": { }"
        return f": {str    }"  None
elif        and        .
    #
                .         "type"  "text"  "text"
    print f": {       }"
else
    return ""  None

            =

        "role"  "system"
        "content"

                "type"  "text"
                "text"  "You are Qwen, a virtual human
                    developed by the Qwen Team, Alibaba Group,
                    capable of perceiving auditory and visual
                    inputs, as well as generating text and
                    speech."



        "role"  "user"
        "content"


                  = False

#
     =          .
                        =True          =False
                    =
                   =
        =
```

```
            =
             =
              =
              =
                      ="pt"
               =True
                          =

           =        .        .       .        .

    #
                           =      .
        **
                          =
               ="Chelsie"


    #
                  =          .

                           =True
                                    =False
      0

    #
                    =
                    .        -1 .        .     .
                =          .                           =False
              ='.wav'
      .                .                                    =24000
                      =            .

    #
    if                  and   .    .
        try
              .
        except
            pass

    return

def process_input                                       =None
    """


    :
```

```
        text_input:
        audio_input:
        dataset_name:

    :
        video_output:
        status_msg:
    """
    try
        #
        if              is not None
            #
                                          =
                                      =
        elif             and              .
            #
                                          =
                                  =
        else
            return None  ""

        if not
            return None  ""

        #
        if not
            return None  "GaussianTalker"

        try
            print "..."
                        = "/root/autodl-tmp/GaussianTalker/dataset"
                    = "/root/autodl-tmp/GaussianTalker/model"

            #
            if               and               .
                             =    .     .
                                  .

                        =
                            =
                          =
                                =
                           =1
                          =-1

            print f": {           }"
```

```
            return              ""
        except Exception as
                        = f": {str    }"
            print
            import traceback
                        .
            return None

    except Exception as
                    = f": {str    }"
        print
        import traceback
                    .
        return None

# Gradio
with    .             ="Qwen2.5-Omni "        =   .        .         as

      .
        """
        #  Qwen2.5-Omni + GaussianTalker


        -
        -
        -
        """


    with    .
        with    .             =1
                .          "###   "

                          =    .
                        =""
                              ="..."
                        =3
                        =""


                           =    .
                        =""
                  type="filepath"
                          = "upload"   "microphone"
```

```
                                    =     .
                        =""
                              =""
                           =
                        =""
                       =""


                           =     .          " "          ="primary"
                       ="lg"
                           =     .          " "          ="secondary"

        with    .               =1
                .           "###   "

                              =     .
                        =""
                           =

                                 =     .
                        =""
                                 =False
                        =3


    #
       .           "###   "
                =     .
                   =
                ""   None
                ""   None
                ""   None

                  =
                =""


    #
                .
            =
                =
                  =


                .
```

```
            =
                =
                  =


                .
            =lambda    None   None   ""   None   ""
                =
                  =


        .
          """
          ---
          ###
          1. ****
             - ""Enter
             - /
             -

          2. ****
             - Qwen2.5-Omni
             - GaussianTalker
             -

          3. ****
             -  `/root/autodl-tmp/GaussianTalker/dataset`
             -

          4. ****
             -
             - GaussianTalker
          """


if __name__ == "__main__"
    import socket

    def find_free_port            =7860               =10
        """""""
        for    in range
                  =             +
            try
                with        .             .
                                .             as
```

```
                    .       ''
                  return
            except OSError
                continue
        raise RuntimeError f"
            {           }-{           +             -1} "

    #
                = int    .        .     'GRADIO_SERVER_PORT'   0
    if              == 0
                     =                  7860
        print f": {            }"
    else
        print f": {            }"

    # Gradio
        .
                  ="0.0.0.0"    #
                  =               #
             =False               #
                  =True           #
```

A.2 GaussianTalker Training Hyperparameters

This section details the specific hyperparameters used to train the GaussianTalker model, as referenced in the experiments of Chapter 4. These settings were applied to the dataset derived from the specific subject video (e.g., the Obama sequence from AD-NeRF dataset) to achieve the performance reported in Table 4.3.

Table A.1: Detailed hyperparameters for the GaussianTalker model training.

Hyperparameter	**Value**
Dataset Source	AD-NeRF Dataset (Obama Sequence)
Video Resolution	512×512 pixels
Training Iterations	120,000
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.999$)
Learning Rate	1.0×10^{-4}
LR Scheduler	Exponential Decay ($\gamma = 0.99$)
Batch Size	1
Initial Points (3DGS)	1,638
Densification Interval	Every 100 iterations
Rendering Resolution	512×512 pixels