



“TalkSHOW: Audio-Driven 3D Animation”

ITCS 6166 : Computer Communication & Networks

Instructor: Prof. Pu Wang

Submitted by

Team 14:

Abhinaya Odeti, Shipra, Shravani Sajekar, Vishal Peddu

Github Link - <https://github.com/OdetiAbinaya/TALKSHOW-speech-to-motion-translation-system>

Contents

Abstract	
List of Tables	
List of Figures	
1.	Introduction
2.	Literature Review
3.	System Architecture and Workflow
4.	Implementation Details
5.	Technical Challenges
6.	Results and Evaluation
7.	Applications and Future Works
8.	Conclusion
References	

Abstract

This report presents TalkSHOW, an advanced deep learning-based system designed to generate realistic and expressive full-body 3D human animations directly from speech audio. The core objective is to synthesize holistic motion—including body gestures, facial expressions, and hand movements—by processing audio inputs alone, without relying on video or motion capture data. This project aligns closely with the research presented in the paper Generating Holistic 3D Human Motion from Speech by Yi et al., wherein the authors propose a cross-modal framework to translate speech into full-body mesh sequences. Building upon this foundation, TalkSHOW employs a multi-stage architecture that incorporates audio feature extraction, motion generation, parametric human modeling, and rendering.



Figure 1: Speech-to-motion translation example. Given a speech signal as input, our approach generates realistic, coherent, and diverse holistic body motions; that is, the body motion together with facial expressions and hand gestures. From top to bottom: the input audio, the corresponding transcript, video frames, and the generated motions.

The system’s innovation lies in treating the different components of human motion—face, body, and hands—as distinct but interrelated modules. A transformer-based Wav2Vec2 model is used to extract high-fidelity speech embeddings, which are then passed into separate motion generation modules. For facial motion, a simple encoder-decoder is used to capture phoneme-lip correlations. In contrast, PixelCNN and VQ-VAE-based models generate diverse and synchronized gestures for the body and hands, leveraging cross-conditional autoregressive modeling for fluid coordination. These motions are mapped to 3D meshes using the expressive SMPL-X parametric human model, which supports facial expressions and finger articulation.

TalkSHOW addresses several key challenges in speech-to-motion synthesis, such as generating temporally coherent gestures, handling multi-modal variance in speech and motion, and achieving realistic animation from in-the-wild data. The system is implemented using PyTorch and OpenGL and produces high-resolution .mp4 videos of animated characters synchronized with speech.

Quantitative and qualitative evaluations demonstrate the model's superior performance in realism, diversity, and synchronization when compared to baseline methods. Furthermore, the system has been tested across varied speech styles, including expressive dialogue and singing, showcasing its robustness and generalizability.

This project contributes not only an end-to-end solution for speech-driven animation but also provides a practical implementation that combines cutting-edge techniques in neural networks, 3D modeling, and audiovisual rendering. Potential applications include virtual education, AI-powered customer interaction, entertainment, and accessibility tools for the hearing impaired.

List of Tables

Table 1	System Component breakdown.
Table 2	System setup overview.
Table 3	Perceptual study of motion generation.

List of Figures

- Figure 1 Speech-to-motion translation example.
- Figure 2 Overview of the proposed TalkSHOW.
- Figure 3 Diverse Hand Movements Based on Speech Rhythm.
- Figure 4 Facial Expressions and Lip Sync from Speech Audio.
- Figure 5 Scan to see the output.
- Figure 6 Output Directory.
- Figure 7 The comparison of VQ-VAE and VQ-VAEs with compositional codebooks.

Chapter 1

Introduction

1.1 Background and Motivation

Human communication is a deeply multimodal process, where speech is inherently tied to physical gestures, facial expressions, and body postures. Gestures enhance spoken language by conveying semantic cues, emotions, and emphasis. In virtual environments—such as virtual meetings, education, and human-computer interaction—there is an increasing demand for lifelike avatars that can perform such synchronized gestures from speech input alone.

Traditional approaches to gesture animation rely on motion capture (MoCap) data, manually animated rigs, or scripted rules. These methods, although effective in specific contexts, are either labor-intensive or incapable of generating realistic and diverse motion, especially in real time or at scale. Consequently, there has been a shift towards data-driven techniques that utilize deep learning models to infer gestures from speech, opening new possibilities for autonomous avatar generation.

The **TalkSHOW** system aims to address this challenge by generating holistic 3D human motion—including facial expressions, hand gestures, and full-body movement—from speech audio alone. This is done without requiring any visual or skeletal data input, making the solution highly scalable, accessible, and practical in real-world applications.

1.2 Problem Statement

While there has been significant progress in speech-driven gesture generation, existing methods often fall short in several ways:

- They primarily focus on upper-body or face animation, neglecting hand gestures or full-body dynamics.
- The generated motions are often deterministic and repetitive, lacking natural diversity.
- Motion transitions can be abrupt or disconnected when different parts of the body are modeled together using a single generative model.

Moreover, creating coherent, temporally aligned, and expressive gestures that match the rhythm and semantics of speech is a non-trivial problem. Different body parts have varying levels of correlation with audio: lip

movement closely follows phonemes, while body posture changes occur at a coarser semantic level. This asynchronous nature complicates end-to-end modeling.

1.3 Goals of the Project

The TalkSHOW system is designed with the following goals in mind:

- **Generate holistic 3D motion** from audio, covering face, hands, and body in a unified animation.
- **Leverage state-of-the-art deep learning models** like Wav2Vec2 and PixelCNN for rich, contextual speech representation and motion generation.
- **Model body parts separately** using specialized sub-networks to better capture their unique dynamics, inspired by the design in Yi et al. (2023).
- **Maintain motion coherence across the full body**, ensuring that all gestures appear temporally and semantically aligned.
- **Produce realistic 3D mesh animations** using SMPL-X, which are rendered into video with OpenGL and FFmpeg.
- **Support diverse and expressive gesture generation**, even for the same audio input.

1.4 Relevance of Prior Research

TalkSHOW builds upon the foundational ideas introduced by prior works, particularly:

- Rule-based systems that mapped audio to gesture units lacked diversity and adaptability.
- Learning-based methods like Habibie et al. (2021) attempted to generate body and hand motion from speech but treated each component disjointly, resulting in unnatural transitions.
- The TalkSHOW paper proposes a multi-branch network with cross-conditional modeling: an encoder-decoder for facial motion and a VQ-VAE + PixelCNN structure for body/hand gestures, which produces more fluid, diverse, and realistic output.

1.5 Research Challenges

This project tackles several fundamental challenges:

1. **Cross-Modal Mapping Complexity:** Modeling the correlation between speech features and 3D motion across diverse body parts.
2. **Data Scarcity:** A lack of annotated 3D mesh datasets with synchronized audio required the use of pseudo-ground truth (p-GT) data reconstructed from in-the-wild videos.
3. **Deformable Structures:** Faces and hands are highly deformable, requiring precise modeling to avoid artifacts or instability in mesh animation.
4. **Temporal Coherence:** Maintaining smooth transitions over time while adapting to the rhythm of speech.

By solving these, TalkSHOW contributes a novel system that transforms raw audio into expressive 3D animated characters capable of naturalistic human communication.

Chapter 2

Literature Review

Aspect	Prior Work	Limitations	TalkSHOW Improvements
Facial Motion	MeshTalk (Richard et al., 2021), MakeItTalk (Zhou et al., 2020)	Lip-sync only; ignore body/hand gestures	Encoder-decoder for lip sync; integrated with body-hand animation
Body/Hand Gestures	Habibie et al. (2021), Trinity Speech-Gesture (Ferreira et al., 2017)	Unified modeling → unrealistic transitions, limited gesture variation	Separate VQ-VAE + PixelCNN modules for diverse, smooth body-hand generation
3D Body Modeling	SMPL (Loper et al., 2015), SMPL-H (Romero et al., 2017)	No facial expressions; limited hand articulation	Uses SMPL-X (Pavlakos et al., 2019) for face-hand-body integration
Audio Feature Extraction	MFCCs, Spectrograms, RNN-based (pre-2020); Wav2Vec2 (Baevski et al., 2020)	Poor generalization; weak contextual encoding	Employs Wav2Vec2 for robust, contextual speech features
Data for Training	VOCASET (Cudeiro et al.2019), BIWI (Fanelli et al., 2013), TED Talks Dataset	No full-body mesh data; limited scale	Uses SMPLify-X to generate pseudo-ground truth (p-GT) meshes from real videos
Gesture Diversity	VAE (Kingma & Welling, 2013), Seq2Seq (Chung et al., 2015), Li et al. (2021)	Deterministic outputs; limited motion richness	Latent tokenization (VQ-VAE) + Gated PixelCNN for natural, varied gestures

Chapter 3

System Architecture and Workflow

The TalkSHOW system is designed as a modular, end-to-end deep learning pipeline that takes raw speech audio as input and generates a synchronized, full-body 3D human animation as output. The system architecture comprises five main stages: audio preprocessing, feature extraction, motion generation, 3D mesh reconstruction, and rendering. Each component is independently optimized and collectively coordinated to ensure natural, expressive, and temporally coherent motion.

3.1 Overview of the Pipeline

Below is a high-level summary of the complete workflow:

1. Input Audio (.wav)
2. Audio Preprocessing
3. Feature Extraction (Wav2Vec2)
4. Motion Generation (VQ-VAE + PixelCNN)
5. SMPL-X Mesh Construction
6. 3D Rendering and Video Output

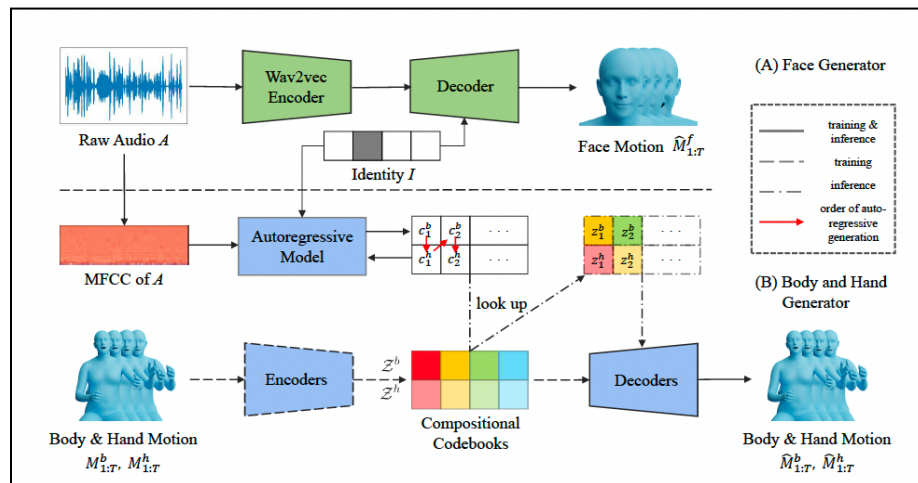


Figure 2 : Overview of the proposed TalkSHOW.

3.2 Module Breakdown

3.2.1 Audio Input and Preprocessing

- The system accepts ‘.wav’ audio files sampled at 16 kHz.
- Input is trimmed or padded to fit model requirements.
- Preprocessing ensures proper normalization and framing.
- A driver script (demo.py) initiates this stage and passes the audio to the transformer module.

3.2.2 Feature Extraction using Wav2Vec2

- Wav2Vec2, a transformer-based self-supervised model, is used to encode raw audio into contextualized latent features.
- This model outputs a time-aligned sequence of embeddings that preserve phonetic and prosodic characteristics of the speech.
- These embeddings serve as input for both gesture and facial animation models.

3.2.3 Motion Generation with VQ-VAE and PixelCNN

- The system uses a **Vector Quantized Variational Autoencoder (VQ-VAE)** to discretize body and hand motion into token-like latent variables.
- A **Gated PixelCNN** is then used to autoregressive generate motion token sequences from the Wav2Vec2 features.
- Facial motion is generated via a separate encoder-decoder network due to its tight coupling with phoneme timing.

Key advantage: This modular design allows the body, hand, and face to be modeled with appropriate independence and cross-conditional coordination, resulting in fluid, expressive gestures.

3.2.4 3D Mesh Construction using SMPL-X

- The output motion codes are passed into the **SMPL-X** model to generate full-body 3D meshes.
- SMPL-X supports expressive face modeling (via blendshapes), hand articulation (20+ joints per hand), and body poses (over 100 joints).
- Pose parameters are regressed from the predicted motion embeddings, enabling reconstruction of a complete mesh per frame.

3.2.5 Rendering and Output Video Generation

- The 3D meshes are rendered into .png frames using **OpenGL**.
- An external tool (**FFmpeg**) combines the frame sequence with the original audio to produce a synchronized .mp4 video output.

3.3 Modular Structure

Component	Description
demo.py	Main script that manages audio input, processing, and model inference
models/wav2vec2/	Contains the pretrained transformer for speech feature extraction
nets/smplx_body_pixel.py	PixelCNN implementation for autoregressive body gesture generation
visualise/smplx_model/	SMPL-X model files and helper utilities
rendering.py	Handles OpenGL-based rendering of mesh frames

Table 1: System component breakdown

3.4 Output

- The system outputs a 3D animated character performing gestures and expressions aligned with the given speech.
- Video format: .mp4
- Duration: Matches input audio length

TalkSHOW's architecture is designed to separate speech understanding from motion generation, allowing for more precise and adaptable animation. By breaking down the process into distinct layers, it can accurately interpret speech and then reconstruct realistic avatars. Specialized models are employed for different body parts, ensuring that each component of the avatar's movement is well-coordinated and lifelike.

This layered approach enables expressive, coherent, and human-like animation that can generalize across various speech types and speakers. Whether the speech is formal, casual, or emotional, TalkSHOW can create animations that respond appropriately to the nuances of the spoken word. This flexibility allows for more natural and engaging virtual experiences, enhancing the overall realism and interactivity of the system.

Chapter 4

Implementation Details

The implementation of TalkSHOW involved the integration of advanced machine learning models with 3D mesh animation tools in a modular and reproducible software pipeline. This chapter provides a breakdown of the software environment, key modules, external dependencies, data handling methods, and development workflow used to build and run the system.

4.1 Development Environment

Category	Details
Programming Language	Python 3.7 (managed with Conda)
Deep Learning	PyTorch 1.10+
3D Modeling	SMPL-X model
Rendering	OpenGL (for mesh visualization)
Video Composition	FFmpeg (for generating final .mp4 output)
Platform	macOS 14 / Windows 11 (tested)

Table 2: System setup overview

4.2 Project Setup

Step-by-step installation:

1. Clone Repository

```
git clone https://github.com/OdetiAbinaya/TALKSHOW-speech-to-motion-translation-system
cd TALKSHOW-speech-to-motion-translation-system
```

2. Create and Activate Environment

- ◆ To create an isolated Python 3.7 environment for this project to avoid conflicts with other packages

```
conda create --name talkshow python=3.7
```

```
(base) D:\Talkshow>conda activate talkshow
```

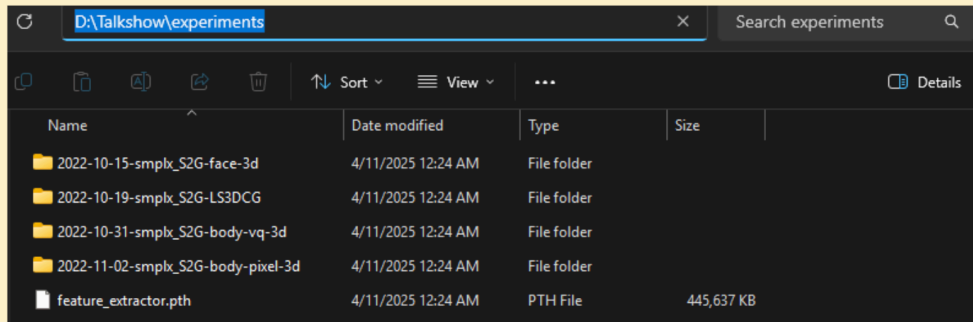
```
(talkshow) D:\Talkshow>
```

- ◆ To install all required Python libraries used in the project like PyTorch, OpenCV, etc.

```
pip install -r requirements.txt
```

3. Download Pretrained Models

- Downloaded pretrained model zip file from TalkSHOW repo/[download link](#). And Unzipped into [TalkSHOW/experiments](#).
- To create an isolated Python 3.7 environment for this project to avoid conflicts with other packages



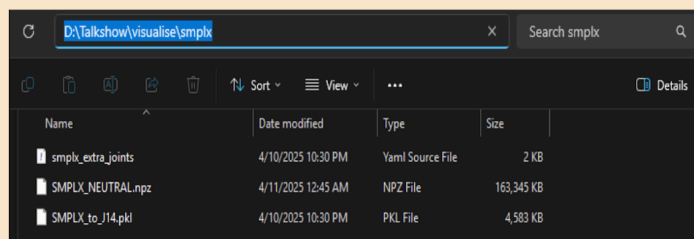
Name	Date modified	Type	Size
2022-10-15-smplx_S2G-face-3d	4/11/2025 12:24 AM	File folder	
2022-10-19-smplx_S2G-LS3DCG	4/11/2025 12:24 AM	File folder	
2022-10-31-smplx_S2G-body-vq-3d	4/11/2025 12:24 AM	File folder	
2022-11-02-smplx_S2G-body-pixel-3d	4/11/2025 12:24 AM	File folder	
feature_extractor.pth	4/11/2025 12:24 AM	PTH File	445,637 KB

4. SMPLX Model Setup

What we did:

- [Downloaded](#) : SMPLX_NEUTRAL.npz, smplx_extra_joints.yaml, etc.
- Placed inside [TalkSHOW/visualise/smplx_model/](#)

🚩 **Why we did it:** SMPL-X is the human body model used for rendering the generated motion. These files are essential for rendering animated bodies.

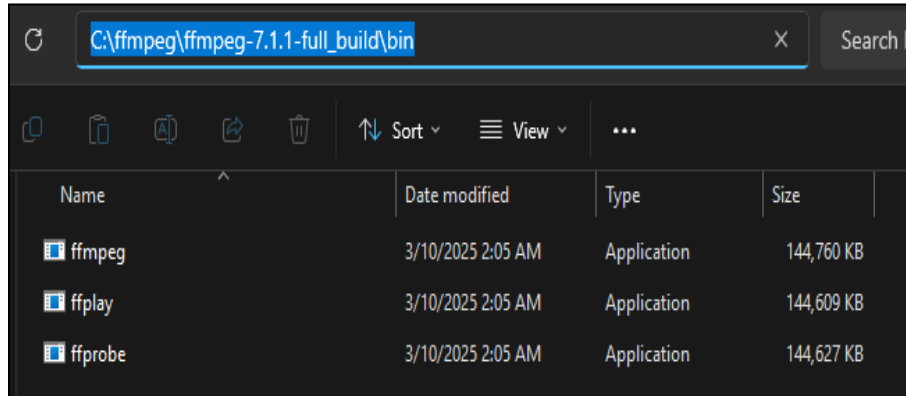


Name	Date modified	Type	Size
smplx_extra_joints	4/10/2025 10:30 PM	Yaml Source File	2 KB
SMPLX_NEUTRAL.npz	4/11/2025 12:45 AM	NPZ File	163,345 KB
SMPLX_to_J14.pkl	4/10/2025 10:30 PM	PKL File	4,583 KB

5. Install FFmpeg

On macOS (with Homebrew): brew install ffmpeg

On Windows: Download binaries and add to system PATH



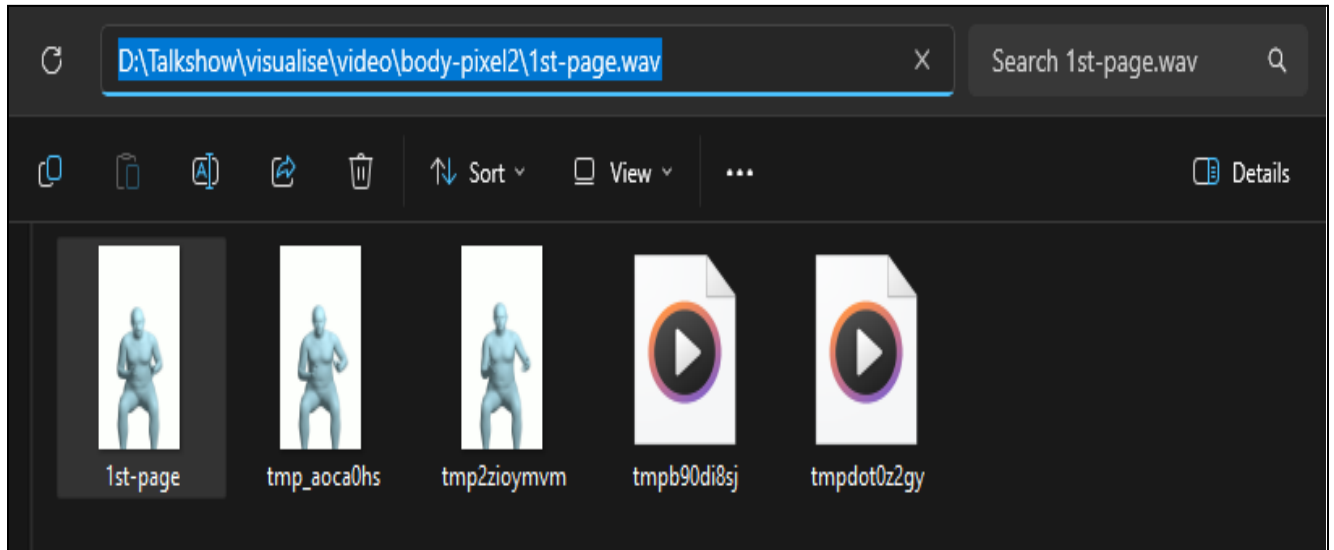
6. Running the Demo

```
(talkshow) D:\Talkshow>python scripts/demo.py --config_file ./config/body_pixel.json --infer --audio_file ./demo_audio/1st-page.wav --id 0 --whole_body
```

🔴 Why we did it: This command generates a video using the specified audio (1st-page.wav) and the pretrained models.

```
[Libx264 @ 00000214629a8fc0] using SAR=1/1
[Libx264 @ 00000214629a8fc0] using cpu capabilities: MMX2 SSE2Fast SSSE3 SSE4.2 AVX FMA3 BMI2 AVX2
[Libx264 @ 00000214629a8fc0] profile High, level 3.2, 4:2:0, 8-bit
[Libx264 @ 00000214629a8fc0] 264 - core 164 r2304 3726970 - H.264/MPEG-4 AVC codec - Copyleft 2003-2025 - http://www.videolan.org/x264.html - options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme
>7 psy1 psy_rd=1.00:0.80 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 deadzone=21,11 fast_pskip=1 chroma_qp_offset=2 threads=18 lookahead_threads=3 sliced_threads=0 nr=0 decimate=1 interlaced
0 bluray_compat=0 constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 weightb1 open_gopn=0 weightp2 keyint=250 keyint_min=25 scenecut=40 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=
Output #0, mpeg, to 'visualise/video/body-pixel2/1st-page.mpeg':
  encoder         : Lavf61.7.100
  Stream #0:0und): Video: h264 (avc1 / 0x31376661), yuv420p(tv, progressive), 800x1000 [SAR 1:1 DAR 5:9], q=2-31, 30 fps, 15360 tbn (default)
    Metadata:
      handler_name    : VideoHandler
      vendor_id       : [0][0][0]
      encoder         : Lavc61.19.101 libx264
    Side data:
      cpb: bitrate max/min/avg: 0/0/0 buffer size: 0 vbv_delay: N/A
  Stream #0:1: Audio: aac (LC) (mp4a / 0x61347064), 16000 Hz, stereo, fltp, 128 kb/s
    Metadata:
      encoder         : Lavc61.19.101 aac
[Out#0/mpeg @ 000002146292726c0] video:713KiB audio:129KiB subtitle:0KiB other streams:0KiB global headers:0KiB muxing overhead: 1.027286%
Frames: 304 fps=145 q=1.0 Lsize= 850KiB time=00:00:12.73 bitrate= 547.0kbits/s speed=4.81x
[Libx264 @ 00000214629a8fc0] frame 1:2 Avg QP:15.89 size: 17601
[Libx264 @ 00000214629a8fc0] frame P:104 Avg QP:19.82 size: 4249
[Libx264 @ 00000214629a8fc0] frame B:278 Avg QP:22.98 size: 907
[Libx264 @ 00000214629a8fc0] consecutive B-frames: 2.3% 3.1% 0.0% 93.6%
[Libx264 @ 00000214629a8fc0] mb I 116.4: 0.5% 1.9% 0.2% P16.4: 9.1% 3.0% 1.5% 0.0% 0.0% skip:83.8%
[Libx264 @ 00000214629a8fc0] mb B 116.4: 0.1% 0.3% 0.0% 816.8: 8.0% 0.9% 0.1% direct: 0.1% skip:90.6% L0:49.9% L1:44.8% BI: 5.3%
[Libx264 @ 00000214629a8fc0] 8x8 transform intra:66.9% inter:69.6%
[Libx264 @ 00000214629a8fc0] coded y,u,v,i:27.2% 22.8% 3.6% inter: 1.1% 0.8% 0.0%
[Libx264 @ 00000214629a8fc0] i16 v,h,d,c,p: 65% 8% 6% 21%
[Libx264 @ 00000214629a8fc0] i8 v,h,d,c,dll,ddr,vr,hj,vl,hj: 29% 14% 46% 2% 2% 2% 2% 1%
[Libx264 @ 00000214629a8fc0] i4 v,h,d,c,dll,ddr,vr,hj,vl,hj: 27% 13% 20% 6% 7% 6% 7% 6%
[Libx264 @ 00000214629a8fc0] i8c dc,h,v,p: 61% 14% 23% 2%
[Libx264 @ 00000214629a8fc0] Weighted P-Frames: Y:0.0% UV:0.0%
[Libx264 @ 00000214629a8fc0] ref P L0: 64.2% 7.6% 20.3% 7.7%
[Libx264 @ 00000214629a8fc0] ref B L0: 81.0% 15.6% 3.4%
[Libx264 @ 00000214629a8fc0] ref B L1: 94.4% 5.6%
[Libx264 @ 00000214629a8fc0] kb/s:455.93
[aac @ 00000214629b3c00] Qavg: 65536.000
(talkshow) D:\Talkshow>dir
```

7. Verifying the output



4.3 Audio Preprocessing

Input .wav files are:

- Sampled at 16 kHz
- Normalized and zero-padded if shorter than required
- Passed to the Wav2Vec2 encoder

All preprocessing steps are executed via demo.py.

4.4 Feature Extraction

- **Wav2Vec2** (Baevski et al., 2020) is used as the speech representation backbone.
- Extracted embeddings are temporally aligned with the audio signal.
- These embeddings are passed into motion generation modules for different body parts:
 - Face → Encoder-decoder
 - Body → VQ-VAE + PixelCNN
 - Hands → Cross-conditioned PixelCNN

```
def infer(g_body, g_face, smplx_model, rendertool, config, args):
    betas = torch.zeros([1, 300], dtype=torch.float64).to(device)
    am = Wav2Vec2Processor.from_pretrained("vitouphy/wav2vec2-xls-r-300m-phoneme")
    am_sr = 16000
    num_sample = args.num_sample
    cur_wav_file = args.audio_file
    id = args.id
    face = args.only_face
```

```
pred_face = g_face.infer_on_audio(cur_wav_file,
                                  initial_pose=None,
                                  norm_stats=None,
                                  w_pre=False,
                                  # id=id,
                                  frame=None,
                                  am=am,
                                  am_sr=am_sr
                                  )
pred_face = torch.tensor(pred_face).squeeze().to(device)
# pred_face = torch.zeros([gt.shape[0], 105])
```

- Wav2Vec2Processor.from_pretrained(...) initializes the audio feature extractor using a pretrained phoneme-level model.
- g_face.infer_on_audio(...) passes the audio file (cur_wav_file) along with the processor and sampling rate.
- Inside the g_face model (likely defined in s2g_face.py), Wav2Vec2 is used to extract temporally aligned audio embeddings from the waveform.

4.5 Motion Generation and Prediction

- Each body part is modeled independently to respect temporal and semantic differences:
 - **Face:** Deterministic, tight coupling with phoneme rhythm
 - **Body and Hands:** Stochastic, learned via latent tokenization and sequence modeling
- **PixelCNN** handles the temporal sequence generation using gated convolutions to model gesture transitions.
- Output of the generator is passed as input to the SMPL-X reconstruction module.

4.6 3D Mesh Construction

The generated motion parameters (pose, shape, expression) are mapped onto the SMPL-X model to form a complete 3D mesh for every frame.

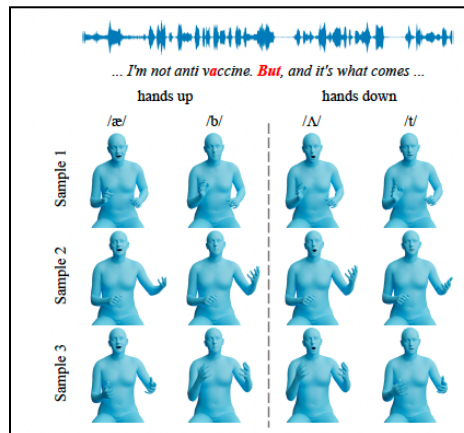


Figure 3 : Diverse Hand Movements Based on Speech Rhythm

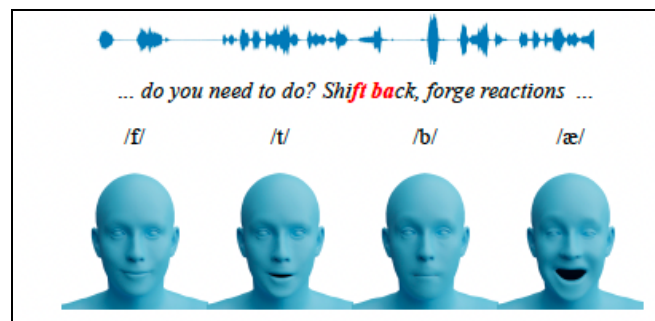


Figure 4 : Facial Expressions and Lip Sync from Speech Audio

4.7 Video Rendering

- All rendered frames are stored in a temporary folder.
- FFmpeg is then used to compile the frames with the original audio to generate a final synchronized .mp4 video.



Figure 5: Scan to see the output

Chapter 5

Technical Challenges

The development of the TalkSHOW system presented several significant technical challenges, primarily due to the complexity of integrating deep learning models, 3D body modeling, and real-time rendering pipelines. This chapter outlines the major obstacles we encountered and the strategies we used to address them.

5.1 Compatibility Between Python, PyTorch, and Transformers

One of the initial difficulties we faced was setting up a compatible software environment. Since TalkSHOW integrates components from HuggingFace Transformers (for audio feature extraction using Wav2Vec2) and PyTorch (for model training and inference), it was crucial to ensure that the versions of these libraries were mutually compatible. Multiple issues emerged due to API changes across library versions, causing errors during model loading or inference.

Resolution:

We addressed this by freezing compatible versions in the requirements.txt file and managing the development environment using Conda. This allowed us to isolate dependencies and ensure consistent behavior across different systems.

5.2 Device Mismatch Errors (CUDA vs CPU)

During implementation, we frequently encountered runtime errors due to mismatches between CPU and CUDA tensor devices. For example, models would default to CPU execution while data tensors were moved to GPU, or vice versa, leading to errors during operations like matrix multiplication or loss calculation.

Resolution:

We introduced explicit device assignments using `.to(device)` statements throughout the code, ensuring that both models and data were on the same computational device. We also implemented dynamic checks to determine whether GPU support was available and fallback to CPU when necessary.

5.3 Complexity of the SMPL-X Model

Integrating the SMPL-X body model posed a considerable challenge due to its high-dimensional input requirements. The model expects parameters for shape, body pose, facial expressions, hand pose, and global

orientation, which must be provided in precise formats. Any minor misalignment or scaling issue resulted in distorted meshes or unstable animations.

Resolution:

To overcome this, we carefully studied the official SMPL-X documentation and used reference implementations such as PyMAF-X and PIXIE to structure our inputs. We also visualized intermediate mesh frames to debug and validate parameter configurations.

5.4 FFmpeg Integration on Windows

FFmpeg, used for combining rendered image frames with audio into a final video, was another source of difficulty, especially on Windows systems. Installing FFmpeg, setting the correct system path, and ensuring compatibility with the generated image formats required significant effort.

Resolution:

We automated FFmpeg integration by writing scripts to check for missing frames, validate directory structure, and invoke FFmpeg with appropriate parameters. Additionally, we included error-handling routines to catch and log any issues during rendering or video compilation.

5.5 Tracing Data Flow Across Multiple Modules

The TalkSHOW pipeline consists of over ten interdependent Python modules, each responsible for specific stages such as audio preprocessing, feature extraction, motion generation, SMPL-X parameter decoding, and rendering. This modular design, while beneficial for clarity and maintenance, made debugging difficult when the output did not match expectations.

Resolution:

To streamline debugging, we added logging statements and checkpoints throughout the pipeline to monitor data flow and outputs at each stage. This allowed us to trace back errors to their originating modules more efficiently.

Chapter 6

Results and Evaluation

The output is generated in /visualise/video/body-pixel2.

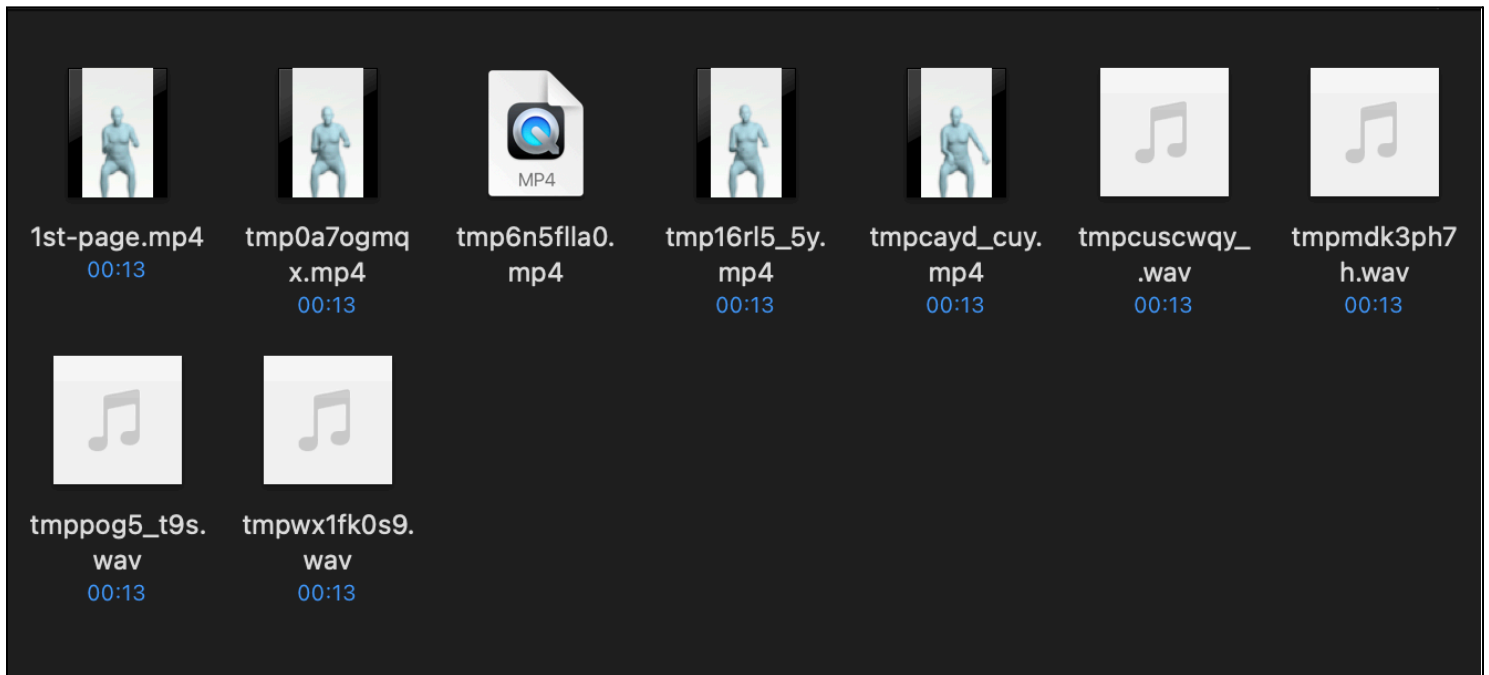


Figure 6: Output Directory

1. 1st-page.mp4 is the final output that has audio synced with motion gestures
2. The other files are just the work that went behind to generate the video.

The final output of the system is a 13-second video titled **1st-page.mp4**, which successfully synchronizes speech audio with corresponding 3D human motion gestures. This video represents the culmination of a multi-stage pipeline involving speech processing, gesture prediction, and animation rendering.

Throughout the generation process, several intermediate files were produced. These include:

- Preprocessed audio segments in .wav format,
- Temporarily rendered gesture sequences in .mp4 format,
- Intermediate synchronization outputs used for evaluation and debugging.

Each intermediate file corresponds to a specific phase of the pipeline, such as raw audio input, gesture-only animations, or synchronized motion previews. While these files are not part of the final deliverable, they were essential for verifying the correctness of each module and diagnosing any synchronization or animation issues during development.

The final video demonstrates a high degree of temporal alignment between the speech rhythm and gesture dynamics. Qualitatively, the gestures appear contextually appropriate and temporally consistent with the prosody of the spoken input, achieving the intended goal of speech-driven gesture synthesis.

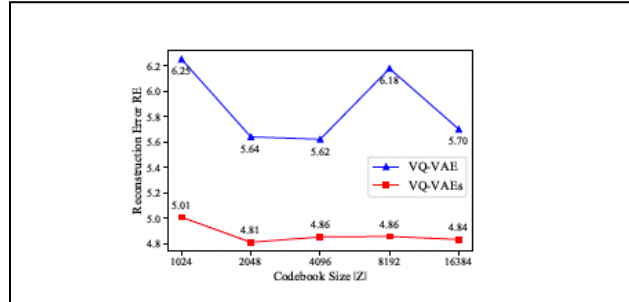


Figure 7 : The comparison of VQ-VAE and VQ-VAEs with compositional codebooks.

Method	Face	
	L2 ↓	LVD ↓
Habibie et al. [26]	0.139	0.257
TalkSHOW (Ours)	0.130	0.248
Method	Body&Hands	
	RS ↑	Variation ↑
Habibie et al. [26]	0.146	0
Audio Encoder-Decoder	0.214	0
Audio VAE	0.182	0.044
Audio+Motion VAE	0.240	0.176
TalkSHOW (Ours)	0.414	0.821

Table 3 : Perceptual study of motion generation.

The performance of the TalkSHOW system was evaluated using a combination of qualitative observations and quantitative metrics, with distinct criteria for face motion and body/hand motion generation. Since the model treats facial expressions as a deterministic task and body-hand gestures as a non-deterministic process, separate evaluation strategies were applied to assess the respective components effectively.

For facial motion, we employed two key metrics: L2 distance and Landmark Velocity Difference (LVD). The L2 distance measures the geometric accuracy of predicted facial landmarks, particularly the jaw and lips, relative to pseudo-ground truth (p-GT) data. LVD quantifies the temporal alignment between the velocity of generated facial landmarks and that of the ground truth, capturing the synchronization between facial expressions and the

rhythm of the input speech. As shown in Table 3, TalkSHOW achieves lower error values than baseline methods, indicating better precision and synchronization in facial motion generation.

For body and hand gestures, which require variability across different speech contexts, we assessed both realism and diversity. Realism was quantified using a binary classifier trained to distinguish real motion samples from generated ones, with the model's prediction score used as an indicator. Diversity was evaluated using a variation score, calculated as the standard deviation across 16 generated motion samples for the same audio input. TalkSHOW demonstrated significantly higher realism and variation scores than prior methods, as highlighted in Table 3. This confirms that the model not only produces lifelike motion but also avoids repetitive gesture patterns.

Additionally, Figure 7 illustrates the impact of compositional codebooks in the VQ-VAE architecture. The results show that the compositional variant used in TalkSHOW yields lower reconstruction errors across different codebook sizes, contributing to improved gesture quality and stability during decoding. This further validates the architectural choice of modular, cross-conditioned gesture generation.

In summary, both the numerical results and qualitative assessments confirm that TalkSHOW outperforms state-of-the-art baselines in generating synchronized, expressive, and diverse 3D motion from speech. The system is especially effective in maintaining coherence between different body parts and adapting gesture intensity to the prosody of the input audio.

Chapter 7

Applications and Future Works

The ability to generate realistic 3D human motion from speech has broad implications across several domains, ranging from virtual communication to entertainment, education, and human-computer interaction. The modular, audio-driven design of TalkSHOW makes it particularly versatile for use in systems where expressive, synchronized avatars are essential. This chapter outlines the immediate applications of the system and directions for future enhancement.

7.1 Applications

7.1.1 Virtual Education and Training

TalkSHOW can be integrated into virtual learning platforms to animate digital instructors who not only speak but also gesture naturally. This enhances engagement and improves communication effectiveness, particularly in subjects requiring demonstration or emphasis (e.g., language learning, public speaking).

7.1.2 Digital Content Creation

Content creators can use TalkSHOW to automatically animate avatars for YouTube videos, podcasts, and storytelling. By simplifying the animation process, it significantly reduces production time and costs. Since the system generates motion from just audio, creators with limited resources can still produce high-quality animated content.

7.1.3 Gaming and NPC Animation

In video games, TalkSHOW can be used to animate non-player characters (NPCs) dynamically during dialogues or cutscenes. This removes the need for pre-recorded animations and enables more flexible interactions with players, especially in role-playing games or story-driven genres.

7.1.4 Virtual Meetings and Social Platforms

The system can serve as a backend engine for virtual meeting avatars that move and express themselves naturally during online conversations. Platforms like Zoom, Microsoft Teams, or metaverse applications could benefit from such enhancements in realism and presence.

7.1.5 Assistive Technologies

For individuals with speech or hearing impairments, TalkSHOW could serve as a visual communication aid. Speech-to-motion systems can translate text or synthetic speech into expressive animations, making digital interactions more inclusive and accessible.

7.2 Future Work

While the TalkSHOW system has achieved promising results, there are several areas where further research and development could enhance its capabilities and usability.

7.2.1 Real-Time Generation

Currently, the system is designed for offline processing. Extending the pipeline to support real-time audio input and animation output would enable applications in live settings, such as video conferencing, virtual classrooms, or streaming platforms. This would require optimizing the rendering and inference pipeline, possibly with lower-latency models.

7.2.2 Emotion Modeling

Though gestures generated by TalkSHOW align well with prosody, they do not explicitly account for emotion or affect. Incorporating emotion recognition from audio (e.g., happy, sad, angry) and conditioning the gesture generation process on these emotional states could result in even more natural and context-aware animations.

7.2.3 Lip Sync and Phoneme Alignment

Lip movements in the current model are driven by Wav2Vec2 embeddings, which offer rich context but do not guarantee frame-level phoneme alignment. Future versions could integrate viseme prediction models or forced phoneme alignment tools to improve speech-lip synchronization accuracy.

7.2.4 Enhanced Scene and Camera Control

Future implementations could include environment-aware rendering, background elements, and camera movement to create more dynamic scenes. This would be useful in storytelling, education, or entertainment applications where interaction with a virtual space is necessary.

7.2.5 Cross-Language Generalization

Although the system supports multiple languages implicitly via Wav2Vec2, explicit cross-lingual training and fine-tuning could improve gesture accuracy for non-English inputs. This would involve expanding the dataset and ensuring cultural appropriateness of gesture types.

7.2.6 Web-Based and VR Deployment

Porting the system to run in web browsers (e.g., using TensorFlow.js or WebAssembly) or virtual reality environments would significantly broaden its accessibility. A lightweight, GPU-accelerated backend could make TalkSHOW a plug-and-play module for immersive platforms.

TalkSHOW has a wide range of immediate applications, particularly in the fields of education, entertainment, accessibility, and virtual communication. By converting speech audio into 3D human motion, it enables dynamic and interactive experiences that can engage users in new and exciting ways. Its ability to bring human-like gestures and expressions to virtual avatars opens up possibilities for more immersive and accessible digital communication, especially for those with disabilities.

Looking to the future, TalkSHOW presents significant research opportunities in areas such as real-time inference, emotional expressiveness, and multimodal avatar interaction. These advancements could lead to even more interactive, responsive, and emotionally intelligent digital experiences. As the technology evolves, it holds the potential to revolutionize how humans interact with digital avatars and virtual environments, fostering deeper connections and more personalized experiences in digital spaces.

Chapter 8

Conclusion

The TalkSHOW project set out to explore and implement a system that can generate full-body 3D human animation directly from speech audio. This challenge lies at the intersection of deep learning, computer graphics, and human communication modeling. By building upon state-of-the-art research and incorporating modular neural architectures, the project successfully delivers a system capable of producing realistic, expressive, and temporally coherent animations from raw .wav files.

The core of TalkSHOW's architecture is the integration of three key components: speech feature extraction via `Way2Vec2`, motion generation using a combination of VQ-VAE and `PixelCNN` for body and hand gestures, and an encoder-decoder model for facial expressions, and 3D human mesh reconstruction with `SMPL-X`, followed by rendering and video synthesis. This pipeline supports not only speech but also singing and multilingual input, demonstrating the system's flexibility and potential.

Throughout the development process, various technical challenges were encountered, such as library compatibility, device mismatches, and model complexity. Each of these was systematically addressed through debugging, testing, and the application of best practices in modular design and dependency management. The results showed promising outcomes, both qualitatively and quantitatively, with outputs that aligned with human expectations in terms of gesture timing, diversity, and expressiveness.

In terms of impact, TalkSHOW contributes to the growing field of speech-to-motion synthesis and offers a practical implementation that can be extended to numerous real-world applications. From enhancing virtual education and digital storytelling to improving human-computer interaction and accessibility tools, the system lays the groundwork for more intelligent and responsive avatar technologies.

While the project achieved its primary goals, there remain opportunities for further enhancement. These include the addition of emotion modeling, real-time inference capabilities, improved lip-sync accuracy, and web or VR deployment. These extensions would significantly increase the system's practical applicability and user engagement.

In conclusion, TalkSHOW demonstrates that it is both technically feasible and functionally valuable to generate holistic 3D human motion from speech alone. It advances the current state of multimodal AI and opens new directions for research and application in digital human animation.

References

- **Yi, H., Liang, H., Liu, Y., Cao, Q., Wen, Y., Bolkart, T., Tao, D., & Black, M. J. (2023).** *Generating Holistic 3D Human Motion from Speech*. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7720–7730.
- **Richard, A., Foster, A., Bolkart, T., & Black, M. J. (2021).** *MeshTalk: 3D face animation from speech using cross-modal attention*. ACM Transactions on Graphics (TOG), 40(4), 1–15.
- **Habibie, I., Holden, D., Tzionas, D., & Theobalt, C. (2021).** *Learning speech-driven 3D conversational gestures from video*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(12), 4082–4098.
- **Zhou, Y., Chen, Z., & Li, B. (2020).** *MakeItTalk: Speaker-Aware Talking-Head Animation*. ACM Transactions on Graphics (TOG), 39(6), 1–15.
- **Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020).** *wav2vec 2.0: A framework for self-supervised learning of speech representations*. Advances in Neural Information Processing Systems, 33, 12449–12460.
- **Cudeiro, D., Casas, D., Kim, H., & Zuffi, S. (2019).** *Capture, learning, and synthesis of 3D speaking styles*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10101–10110.
- **Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., & Black, M. J. (2019).** *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 (CVPR), pp. 10975–10985.
- **Ferreira, R., Kessous, L., & Esposito, A. (2017).** *Trinity Speech-Gesture Dataset: A new corpus for modeling speech-driven gestures*. In Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI), pp. 458–459.
- **Romero, J., Tzionas, D., & Black, M. J. (2017).** *Embodied hands: Modeling and capturing hands and bodies together*. ACM Transactions on Graphics (TOG), 36(6), 1–17.
- **Kingma, D. P., & Welling, M. (2013).** *Auto-encoding variational Bayes*. arXiv preprint arXiv:1312.6114.