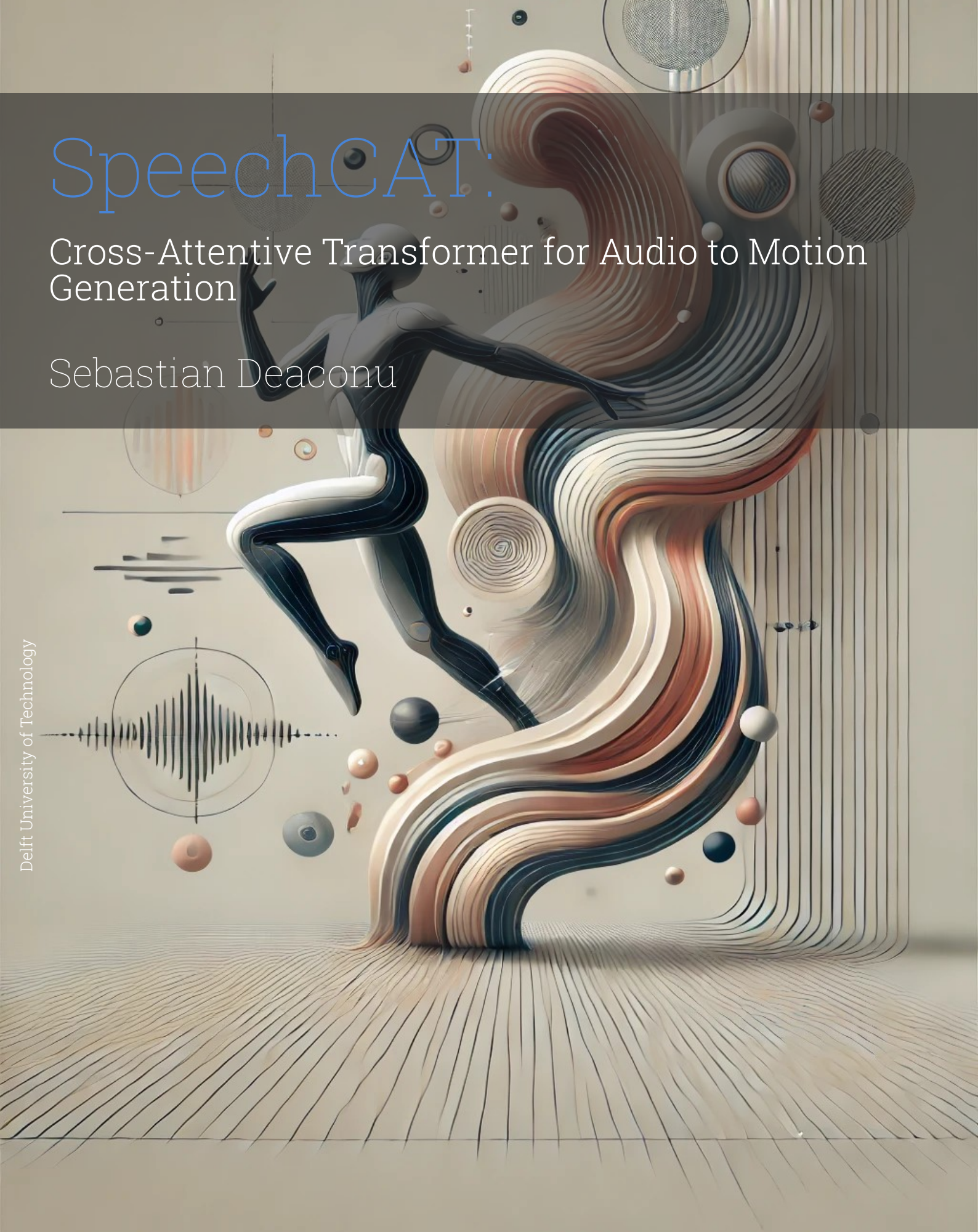
SpeechCAT:

Cross-Attentive Transformer for Audio to Motion Generation

Sebastian Deaconu



Delft University of Technology

TU Delft

SpeechCAT:

Cross-Attentive Transformer for Audio to Motion Generation

by

Sebastian Deaconu

to obtain the degree of Master of Science at the Delft University of Technology,
to be defended publicly on Thursday, 19 February 2025 at 13:00

Student Number:	5058368
Project Duration:	Nov 2024 - Feb 2025
Thesis Committee:	Dr. J. C. van Gemert, TU Delft, thesis advisor
	Dr. Xucong Zhang, TU Delft, daily supervisor
	Dr. Huijuan Wang, TU Delft, committee chair

TU Delft

Preface

My time at Delft University of Technology has come to an end with this. I am grateful of the wisdom, insights, and experiences I have gained along the way, as well as the bright individuals I have encountered who have enabled me to treasure these wonderful years. Despite being demanding and difficult, Delft played a significant role in shaping my life and my current self. I am hopeful about this work's potential in the field of motion generation and hope it reflects my commitment and growth.

I want to express my gratitude to everyone who has assisted me in getting here. The completion of this thesis would not have been feasible without you. First, I want to show my appreciation to Prof. Dr. Xucong Zhan, my supervisor, who has always offered insightful criticism and encouragement. Even when progress was sluggish or seemed impossible, his wisdom and demeanor enabled me to succeed. Your advice and the considerable amount of time you spent have been extremely valuable. Additionally, I would like to thank Prof. Dr. Huijuan Wang for showing interest in this project and for quickly agreeing to chair the committee. Even though we did not communicate much, I appreciated the fresh viewpoints and insights that emerged from our conversations. For the technical guidelines and writing, thank you to Dr. J.C. van Gemert.

I want to thank all of my lecturers and professors for motivating me to pursue this course. I hope more inquiring students hear what you have to say. I also want to thank my family for their unwavering love and support. They have always pushed me to do my best and inspired me to pursue my passions, for which I will always be thankful.

Lastly, I want to thank my partner, Andra, for her support and affection. When matters got difficult, they were there to bring warmth and happiness. I also want to express my gratitude to Radu, Alex, and Octav, my close friends and roommates, for their support and for helping to make the student experience one to remember. I will treasure every memory we were able to create together.

Sebastian Deaconu
Delft, February 2025

Contents

1

Introduction

Although it is a major challenge in the quickly developing field of artificial intelligence, producing motion from speech that is similar to that of a human has many uses. From virtual reality avatars and intelligent robots to online communication tools, the ability to produce synchronized and natural body, hand, and facial gestures is vital for creating immersive and intuitive digital experiences. Due to the limitations of monolithic architectures that are unable to capture complex interdependencies across various body parts, current methods, despite significant advancements, struggle to maintain the coherence and complexity of human motion.

With the help of Vector-Quantized Variational Autoencoders (VQ-VAEs) and a Cross-Attentive Transformer architecture, this thesis introduces a novel method called SpeechCAT. Our approach produces promising results for synchronization, diversity, and realism in motion generation by independently modeling the hands, face, and body and enabling cross-attention between them. Compared to our baselines, we achieve an accuracy gain of 2,55% and 6,17%. The diversity of SpeechCAT is specifically 34.38% and 16.21% higher. Furthermore, it improves temporal smoothness and stability by 0.84% and 0.71%, respectively. By explicitly modeling inter-body part correlations and maintaining computational efficiency, SpeechCAT addresses critical gaps in previous research and pushes state-of-the-art advances.

By bridging these gaps, SpeechCAT offers new possibilities for generating realistic and expressive motions from speech, with implications ranging from advanced robotics to next-generation communication platforms. Future advancements in artificial intelligence and human-computer interaction may be facilitated by this research, which offers not only a technically sound solution but can also accelerate the understanding and advancement of human motion synthesis.

This report is structured in three main parts. The primary objective and definition of the problem are covered in the introduction. The scientific paper represents the core of our report. It was aimed at experts in the computer vision field and was accepted at the IEEE/ACM International Conference on Human-Robot Interaction. The supplementary material chapter provides all the prerequisite information a master's student would need to comprehend the ideas covered in the paper.

2

Scientific Paper

SpeechCAT: Cross-Attentive Transformer for Audio to Motion Generation

Sebastian Deaconu
Tu Delft

Abstract

Audio-to-motion generation is an important task with applications in virtual avatar creation for XR systems and intelligent robot control in daily life scenarios. Most current motion generation methods depend on a single encoder-decoder architecture to simultaneously model all body parts, constraining their capacity to capture the diverse and complex motions exhibited by humans. In this paper, we propose a novel method, SpeechCAT, that employs three separate encoder-decoder modules to individually model the motions of the face, body, and hands. To capture the relationships and synchronization among these body parts, we introduce a cross-attention mechanism to effectively learn their correlations. SpeechCAT ensures sufficient capacity to model the unique characteristics of each body part while preserving the coherence between them. Our experimental results demonstrate the superiority of SpeechCAT over baseline methods, highlighting its effectiveness in generating diverse, realistic, and synchronized motions with face, body, and hand parts.

1. Introduction

Given a human speech audio input, motion generation aims to synthesize synchronized human motion corresponding to the audio similarly to other human movements. This foundational technique has broad applications, including film production [23], virtual digital avatar generation [53], and human-robot interaction [5]. Consequently, this area of research has garnered significant attention, with numerous studies contributing to advancements in the field [18, 21, 59].

Early works in audio-to-motion generation focused mainly on single motion domains, such as facial motion [32], hand gestures[30], or 3D body skeletons [14, 29, 38]. Although these studies achieved promising progress, they often lack the integration of multiple body parts, limiting their applicability to comprehensive human motion generation. Recent studies [33, 43, 50] have begun addressing this limitation by leveraging holistic information to generate full-body human motion from speech. These methods integrated multiple body parts, including face, body, and hands, leveraging human mesh representation [26, 35] as the intermediate or final outputs.

Human motion is a continuous and dynamic process, which makes it computationally expensive to model. To address this challenge, the Vector Quantized Variational Autoencoders (VQ-VAE) framework [45] has been applied in generating human facial [27, 31, 31, 47], body [55], and hand motions [50]. This approach effectively captures discrete representations of continuous motion, significantly reducing the computational burden while preserving motion quality. In terms of facial motions, these methods adopted a lip regressor[33] to synthesize synchronized lip movements from audio or trained an encoder-decoder model to produce synchronized mouth motions in a deterministic manner[50].

Although these motion generation methods have demonstrated promising results, they overlooked the correlations among the three body parts across face, body, and hands. For instance, hand gestures are closely correlated with wrist positions, which, in turn, should affect forearm movement. Similarly, head poses are strongly correlated with face motion and neck positions. Even though face motions, especially lip movements, are linked to the audio input directly, isolating the face domain from the body and hands hinders the generation of natural, synchronized human motions. This limitation shows the need to model across body parts correlations explicitly to enhance the realism and coherence of generated motion.

To address this limitation, we propose a novel method, SpeechCAT, for audio-to-motion generation. Different from previous works, SpeechCAT includes three VQ-VAE to encoder the whole body motions into three descreate codebooks corresponding to the face, body, and hands, respectively. Once the codebooks are fully trained, we concatenate the encoded audio features with body-hand-face triplet indices from the codebooks, which are then processed by a cross-attention transformer module. To model the correlation and synchronization across different body parts of face, body, and hands, we perform the cross-attention across the three body parts. In this way, Speech-

CAT explicitly captures and models the correlations and synchronization among the face, body, and hands, ensuring that the generated motions are coherent and realistic across all domains.

Our experimental results demonstrate that SpeechCAT significantly outperforms baselines that use a single VQ-VAE or without the cross-attention module, highlighting its effectiveness in producing synchronized, diverse and natural human motion. Especially, we find that the SpeechCAT can provide better synchronization in the generated motions.

2. Related Work

2.1. Speech Driven Motion Generation

Human motion generation can be guided by different modalities such as audio, text, or actions. Current attempts at generating motion from speech can be split into rule-based and learning-based methods. Rule-based [6, 22, 37, 41] methods map speech to pre-collected body motions based on pre-designed rules. Although easily explainable and controllable, creating complex, realistic, and coherent motion is timely and expensive due to the rules being manually made. Moreover, these methods suffer from a lack of diversity as they are deterministic. Early learning-based methods primarily focused on generating motions for isolated body parts, such as facial expressions, hand gestures, or 3D body poses [24, 57]. These approaches, while effective in their respective domains, often overlooked the interconnected dynamics of full-body motion. Recent methods have sought to address this limitation by generating holistic human motions, leveraging human mesh representations as intermediate outputs[17, 25, 44, 51, 52].

Recently, diffusion models have emerged as a powerful tool in generative tasks, especially image generation. In the domain of motion generation, they have demonstrated their capability to produce high-fidelity and temporally coherent results by iteratively refining noisy data. For instance, models like [48] and [54] employed a multi-modal framework for co-speech motion generation, integrating audio and text. [1] used diffusion for a denoising mechanism to aid temporal coherence, [3] utilized CLIP latent spaces for gesture synthesis, enabling expressive and semantically rich motion. Although these models are particularly effective when combined with expressive representations, enabling the synthesis of nuanced and realistic human motion and temporal coherence, they focus on single-domain motion or individual body parts and fail to integrate them cohesively. This leads to fragmented motions across face, body, and hands.

2.2. VQ-VAEs in Motion Generation

Vector Quantized Variational Autoencoders (VQ-VAEs), initially proposed in [34], are a VAE variant [20] that aims to learn reconstruction with discrete representations. VQ-VAEs have recently achieved promising results on generative tasks which include different modalities such as image synthesis [11, 49], text-to-image generation [40], speech gesture generation [46], music generation [8, 9] etc. VQ-VAEs offer a compact and discrete representation of continuous data, making them ideal for computationally intensive tasks such as motion synthesis. Models such as [2] proposed hierarchical neural embeddings to capture rhythmic elements in co-speech gestures while [17] effectively synthesized conversational gestures directly from video using different body parts. While rhythm and audio aware, these methods did not address the synchronization challenges of multi-body part motion. These models have been employed to encode different motions such as body, hand, and facial motions into separate codebooks, enabling more diverse and expressive motion generation. The use of distinct codebooks for different body parts has proven beneficial in capturing the unique characteristics of each domain, but it lacks mechanisms to model the correlations between these parts explicitly.

2.3. Transformers in Motion Generation

Transformers, with their ability to model long-range dependencies, have significantly advanced tasks such as image[10], text[15], and video[28] generation. By employing self-attention mechanisms, transformers can capture intricate temporal correlations within sequences. This has also been utilized in motion generation tasks.

In recent works, transformers have been utilized in autoregressive settings to predict motion trajectories from audio[12] or textual[13] input, demonstrating improved coherence and diversity. [7] introduced a novel transformer architecture coupling kinematics and dynamics for 3D human motion prediction, which contributes to more realistic and physically plausible human motion modeling. However, it focuses primarily on prediction tasks rather than generation, limiting its direct applicability to co-speech scenarios. [58] unified multiple perspectives of human motion representation, leveraging a pre-training framework that demonstrated superior generalization capabilities. [36] highlighted the potential of multi-task transformers for motion modeling. While effective in synthesizing specific motion types, its holistic application to conversational gestures remains unexplored. Despite their versatility, these models do not address the issue of synchronization across diverse body parts such as face, hands, and body.

2.4. Text Driven Motion Generation

Text-to-motion generation shares similarities with speech-to-motion tasks, with both requiring the alignment of sequential input (audio or text) to motion. Discrete representations, often facilitated by VQ-VAEs, have been pivotal in bridging this gap[50, 56]. Techniques integrating transformers and discrete codebooks have successfully generated natural and expressive motions from textual descriptions. [19] conceptualized motion as a "language", leveraging GPT-style architectures for generation. [4] demonstrated the effectiveness of transformers for generating emotive gestures. However, its focus on emotional expressiveness does not extend to full-body motion modeling.[46] advanced the field by using GPT for generating natural motions from text. Its application to speech-driven generation is however limited. Although this language-inspired approach provides flexibility, it does not address multi-body part dynamics, critical for conversational gestures.

Our approach builds upon VQ-VAE and transformer frameworks by employing separate VQ-VAEs for face, body, and hands, while also leveraging a cross-attentive transformer to explicitly model correlations between the different body parts (face, body, and hands). This enhances synchronization and coherence in generated motions while enabling finer-grained motion modeling and better alignment across modalities.

3. Method

We aim to generate highly correlated conversational body, hand, and facial gestures that match a given speech sequence. The framework comprises two modules: a VQ-VAE encoder-decoder and a Cross-Attentive Transformer. The former creates a mapping from motion sequences to discrete code sequences, while the latter generates said codes from input speech. The generated codes are then decoded into a sequence of motion vectors to recover the motion data. A visual representation is found in Figures 1 and 2

We aim to generate highly correlated conversational body, hand, and facial gestures that match a given speech sequence. The framework comprises two modules: a VQ-VAE encoder-decoder and a Cross-Attentive Transformer. While the latter generates code sequences from input speech, the former maps continuous motion to discrete code sequences. The generated codes are then decoded into a sequence of motion vectors to recover the motion data. A visual representation is found in Figures 1 and 2

3.1. Motion VQ-VAE

VQ-VAEs, such as the one in [2], can learn discrete representations of continuous data which is especially useful for generative models. Given a sequence of motions $M = \{m_i\}_{i=1}^{|M|}$ with $m_i \in \mathbb{R}^d$ where $|M|$ is the number of frames and d is the dimension of the motion, we aim to learn a codebook $Z = \{z_i\}_{i=1}^{|Z|}$ which contains discrete vectors $z_i \in \mathbb{R}^{d_z}$ that represent a quantized latent space for this input sequence. By splitting the motion data into representative body parts we can further extend the range of motion our codebook can represent. Thus we split our data into three compositional pieces, i.e., body, hands, and face. By doing this we map the pieces to three separate codebooks $Z^b = \left\{z_i^b\right\}_{i=1}^{|Z^b|}$, $Z^h = \left\{z_i^h\right\}_{i=1}^{|Z^h|}$ and $Z^f = \left\{z_i^f\right\}_{i=1}^{|Z^f|}$, where $z_i^b, z_i^h, z_i^f \in \mathbb{R}^{d_z}$. With this, we can achieve $\left|Z^b\right| \times \left|Z^h\right| \times \left|Z^f\right|$ different body-hand-face triplets $\left(z_i^b, z_i^h, z_i^f\right)$ and have a wider range of motion diversity. As shown in Figure 1, we encode our audio features using the encoder $E_{1:\tau} = (e_1, \cdots, e_\tau) \in \mathbb{R}^{64\times\tau}$ which can be mapped to our latent feature space $Z_{1:\tau} = (z_1, \cdots, z_\tau) \in \mathbb{R}^{64\times\tau}$ and later decoded by D for synthesis. Here τ is computed as $\tau = \frac{T}{w}$, with w the temporal downsampling of the encoder, i.e., w pose time frames correspond to a single time embedding. To achieve a balance between speed and quality at inference, w is set to 4. We can quantize an embedding by mapping it to the nearest code in each of the corresponding codebooks.

$$
\begin{aligned}
z_t^b &= \arg\min_{z_k^b \in \mathbb{Z}^b} \left\|e_t^b - z_k^b\right\| \in \mathbb{R}^{64}, \\
z_t^h &= \arg\min_{z_k^h \in \mathbb{Z}^h} \left\|e_t^h - z_k^h\right\| \in \mathbb{R}^{64}, \\
z_t^f &= \arg\min_{z_k^f \in \mathbb{Z}^f} \left\|e_t^f - z_k^f\right\| \in \mathbb{R}^{64}.
\end{aligned} \tag{1}
$$

Optimization goal. The VQ-VAE is optimized by the loss component $\mathcal{L}_{VQ}$[45] which is composed of a reconstruction loss $\mathcal{L}_{rec}$, an embedding loss $\mathcal{L}_{embed}$ and a commitment loss $\mathcal{L}_{commit}$.

$$
\begin{aligned}
\mathcal{L}_{VQ} = \mathcal{L}_{rec}(M_{1:T}, \hat{M}_{1:T}) + \underbrace{\|\text{sg}[E_{1:T}] - Z_{1:T}\|}_{\mathcal{L}_{embed}} \\
+ \beta \underbrace{\|E_{1:T} - \text{sg}[Z_{1:T}]\|}_{\mathcal{L}_{commit}}
\end{aligned} \tag{2}
$$

Here, $\mathcal{L}_{rec}$ is the MSE reconstruction loss, sq is the stop gradient operator for the codebook embedding loss and β is the trade-off hyperparameter for the commitment loss.

3.2. Cross-Attentive Transformer

With the trained VQ-VAE a motion sequence $M_{1:\tau} = (m_1, \cdots, m_\tau) \in \mathbb{R}^{256\times\tau}$ can be now mapped to sequences of indexes of the shape $S_{1:\tau} = (s_1, \cdots, s_\tau) \in \mathbb{R}^{64\times\tau}$. A sequence vector s_i is composed of a body-hand-face triplet $\left(z_i^b, z_i^h, z_i^f\right)$, which represent indexes from the codebooks

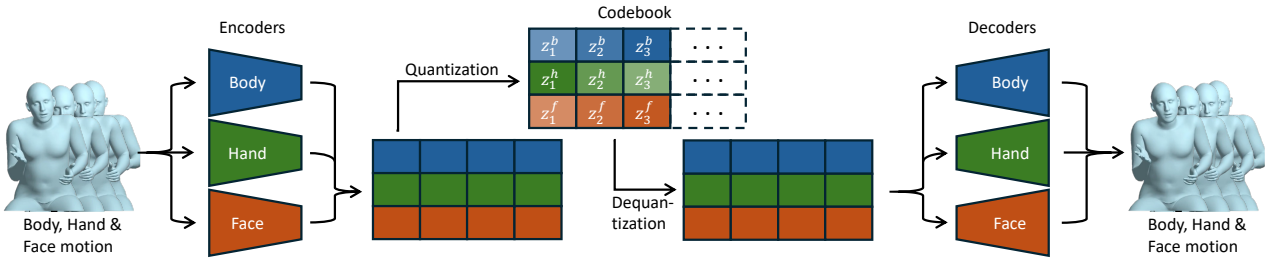


Figure 1. Overview of the VQ-VAE module that learns the discrete codebook indexes by reconstruction the input motion. Different with previous works usually only use one single codebook to model the whole body, we propose to separate the body motions to three codebooks for face $\mathcal{Z}^f$, body $\mathcal{Z}^b$, and hand $\mathcal{Z}^h$, respectively. Each body parts has it own encoder and decoder.

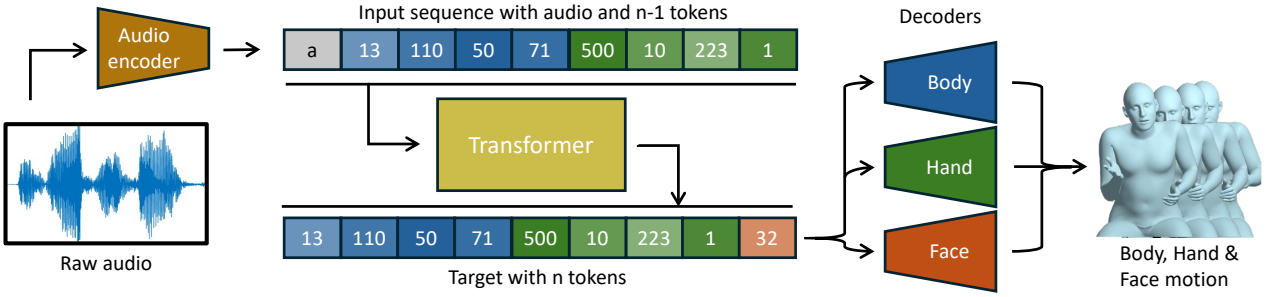


Figure 2. Overview of the motion generation module that takes the audio as input to generate the corresponding body motion. It first maps the audio feature to the motion indexes learned in the QV-VAE. The core of the motion generator is a transfer to prediction the motion index in an auto-regressive manner. The motion index is mapped into motion by the decoder of the VQ-VAE.

$Z^b = \{z_i^b\}_{i=1}^{|Z^b|}$, $Z^h = \{z_i^h\}_{i=1}^{|Z^h|}$ and $Z^f = \{z_i^f\}_{i=1}^{|Z^f|}$. S can then be decoded back to a motion M_{re} through the decoder D. Thus, we can now formulate text-to-motion generation as an autoregressive next-token prediction task[42]. Given a speech input a and the previous sequence $S_{<i}$ we use a Transformer to predict the distribution of the next possible indexes $p(S_i \mid a, S_{<i})$. To extract a speech embedding we use wav2vec which is then aggregated using a learned pooling strategy. This representation can also be seen in Figure 2.

Optimization goal. To optimize the transformer we maximize the log-likelihood of our data distribution. Given that our sequence data likelihood can be mapped as $p(S \mid c) = \prod_{i=1}^{|S|} p(S_i \mid a, S_{<i})$, we then have the following Transformer loss:

$$\mathcal{L}_{\text{trans}} = \mathbb{E}_{S\sim p(S)}\left[-\log p(S \mid a)\right] \tag{3}$$

Self-Attention. We apply the following causal self-attention strategy[39] in speechCAT:

$$\text{Attention} = \text{Softmax}\left(\frac{QK^T \times \text{mask}}{\sqrt{d_k}}\right) \tag{4}$$

where $Q \in \mathbb{R}^{t\times d}$ and $K \in \mathbb{R}^{t\times d}$ are the query and the key while mask represents the causal mask. The mask ensures that future information is not allowed to attend the current tokens. At inference, we start from the speech token and generate the next motion tokens in an auto-regressive manner, until the sequence is equal in size to the initial speech sequence.

Cross-Attention. In addition to causal self-attention, SpeechCAT employs a cross-attention module to enable communication and synchronization between different body parts i.e., body, hands, and face. The cross-attention mechanism is formulated as follows:

$$\text{CrossAttention} = \text{Softmax}\left(\frac{Q_{\text{part1}} K_{\text{part2}}^T}{\sqrt{d_k}}\right) \tag{5}$$

where $Q_{\text{part1}} \in \mathbb{R}^{t\times d}$ and $K_{\text{part2}} \in \mathbb{R}^{t\times d}$ represent the query and key matrices of two different body parts i.e., hands and body. This cross-attention mechanism allows each body part to attend to other parts, effectively sharing information to improve cohesiveness and synchronization in the generated motion. Each body part has its unique query, key, and value representations. Thus, the attention is directed across body parts rather than within a single part, ensuring motion that reflects coordinated gestures or body-language cues.

Method	Body			
	VSD ↓	ASD ↓	MSD ↓	BCS ↑
TalkSHOW	0.0359	0.0624	0.1580	0.851
SpeechCAT	0.0308	0.0529	**0.1519**	0.874
w/o cross-att.	0.0319	0.0548	0.1532	0.866
One-code	**0.0297**	**0.0513**	0.1530	**0.918**

Table 1. Comparison of stability and coherence metrics for body motion prediction between TalkSHOW, SpeechCAT, and two baselines.

Method	Body & Hand			Face	
	L2 ↓	Diversity ↑	MSD ↓	L2 ↓	LVD ↓
TalkSHOW	14.77	**1.07**	0.851	0.2049	**0.0303**
w/o cross-att.	12.47	0.37	0.1532	0.2039	0.0314
One-code	12.007	0.32	0.1530	**0.1892**	0.0321
SpeechCAT	**11.7**	0.43	**0.1519**	0.2048	0.0312

Table 2. Comparison of proposed SpeechCAT, two baselines, and TalkSHOW with multiple error metrics. We separate the body & hand and face since these joints are in different magnitudes. **Note:** Some of the original results from TalkSHOW could not be reproduced. TalkSHOW[50] notes a diversity of 0.821 for the body and for the face an L2 score of 0.130 and an LVD of 0.248

There is a discrepancy in index quality between inference and training. While in training all $i-1$ indexes are assumed to be correct, there is no guarantee that the indexes used for generation are relevant for the conditional probability. To address this we replace $\alpha \times 100\%$ of ground truth indexes, as explained in [55]. This combined with index sampling from the predicted distributions ensures diversity for our Transformer. For the trade-off of stability, α was set to 0.4.

4. Experiments

We evaluated the ability of our method to generate body movements (sequences of poses) effectively from speech in the TalkSHOW dataset[50] quantitatively and qualitatively. Specifically, a 80% / 10% / 10% train-val-test split is used, and the videos are between 3 and 4 seconds. Several metrics are used to measure coherence and stability as well as the diversity of the generated motions which can be split into facial and body poses.

4.1. Experimental setup

Evaluation metrics

Because both the face and the body are modeled in a single non-deterministic task, we asses the generated motion in terms of diversity and accuracy. Even though we use a generative, non-deterministic Transformer model, we test its ability to learn an informative and robust feature space, thus also using metrics such as coherence, synchronization, and stability. The full list of metrics used is the following:

- **L2 error**: L2 distance between GT and generated joints. This applies to both body(all body joints including hands) and face(facial expression joints including jaw position)
- **LVD**: Landmark Velocity Difference calculates the velocity difference between GT and generated body/face joints. It measures the synchronization between the speech and the generated motion.
- **Diversity**: Variance across 16 samples of body and hand motions for the same audio input.
- **VSD**: Velocity Standard Deviation computes the standard deviation of frame-to-frame velocities of generated joints.
- **ASD**: Acceleration Standard Deviation computes the standard deviation of accelerations of generated joints.
- **MSD**: Mean Squared Displacement measures the average frame-to-frame displacement of joints, thus quantifying temporal smoothness.
- **BCS**: Beat Consistency Score measures the alignment between audio and generated body motion.
- **BPSS**: Body Part Synchronization Score is a weighted average of the following motion metrics, calculated per body part(body, hands and face):
  - **Cross-Correlation**: Computes the correlation between pairs of body parts to measure general alignment.
  - **Mean Phase Coherence**: Measures phase alignment to capture synchronization in motion cycles.
  - **Velocity and Acceleration Alignment**: Uses cosine similarity to capture alignment in the direction and speed of movement between body parts.

Compared method

We compare SpeechCAT with two baselines and TalkSHOW[50], another VQ-VAE-based speech-to-motion method. The original results from TalkSHOW could not be reproduced, thus we show our comparison with the results we computed in Table 2. TalkSHOW maps facial generation as a deterministic task, while body and hands are kept as a non-deterministic task. Even though our method is fully nondeterministic, because TalkSHOW generates the face and body as two separate tasks, our results will also be compared separately for the face and body motions. Besides TalkSHOW our baselines, "w/o cross-att" and "one-code", are simplifications of SpeechCAT aimed to show the importance of cross-attention and body part VQ-VAEs. "W/o cross-att" baseline removes the cross-attention module from the transformer, while "one-code" uses a single VQ-VAE to encode the whole body collectively. These are further explained in Section 4.4.

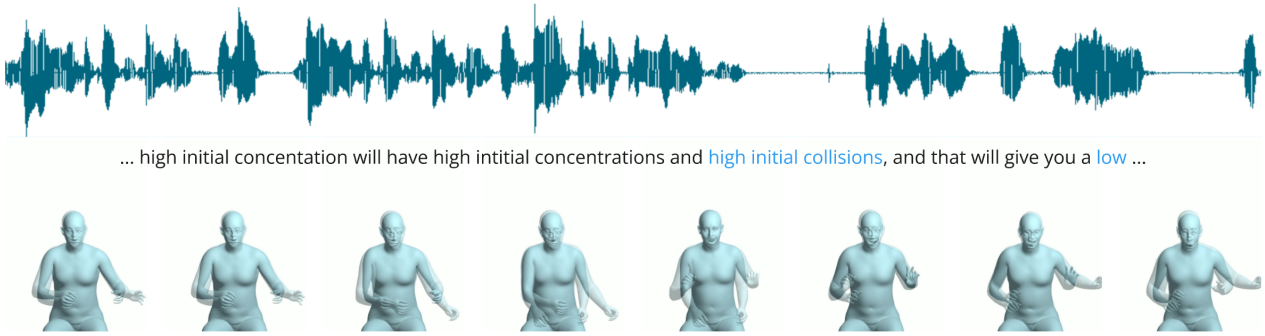


Figure 3. A demonstration of the synchronization between the input audio and motion generated by SpeechCAT. The method generates motion consistent with the rhythm and tone of input audio. The intonation of the strengthening words "high" and "low" is directly expressed through the hand motions.

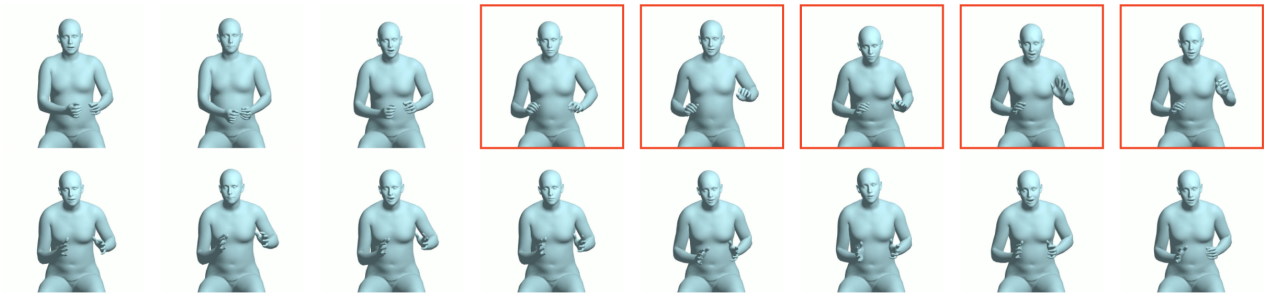


Figure 4. Visual comparison between SpeechCAT(top) and TalkSHOW(bottom). SpeechCAT has a gesticulative motion while TalkSHOW keeps the hands towards the middle with very few movements.

4.2. Quantitative Analysis

From Table 2, it is evident that SpeechCAT outperforms the two baselines across most error metrics, except the L2 error for the face part. Moreover, SpeechCAT achieves comparable results to TalkSHOW, which uses a specific encoder-decoder architecture for the facial features. Although differences are negligible, we can see a decrease in L2 error and a small increase in LVD which shows the model is as effective as TalkSHOW at generating motion. Moreover, the low LVD can be a product of exaggerated motions for speech/phonetic patterns in TalkSHOW. These results highlight the advantages of the proposed SpeechCAT with the cross-attention module and multiple encoder-decoder modules architecture.

When analyzing the body generation, however, Table 2 shows a clear decrease in error but also a decrease in diversity. This is expected due to the transformer's robust nature. Although losing diversity, the model ensures more motion stability and consistency based on the improvement in velocity and acceleration deviation as well as mean square displacement. Moreover, based on the improved beat consistency score, this robust nature generates more audio-consistent motions, as shown in Table 1.

Notably, SpeechCAT demonstrates significant improvements in diversity in body and hand parts, a critical metric for the audio-to-motion generation task. The diversity performance drops substantially for the one-code baseline, underscoring the limitation of using a single encoder-decoder architecture. Although improved, the diversity score is still lower than TalkSHOW. This is expected due to the transformer's robust nature. Although losing diversity, the model ensures more motion stability and consistency based on the improvement in velocity and acceleration deviation as well as mean square displacement. Moreover, this robust nature also generates more audio-consistent motions based on the improved beat consistency score. For the L2 error on the body and hand parts, the baseline without the cross-attention module exhibits the worst performance compared to our other methods. However, its L2 error was better than TalkSHOW. This emphasizes the necessity of the cross-attention mechanism in SpeechCAT for effectively modeling the correlations between body and hand motions.

Overall, we conclude that the proposed SpeechCAT effectively maps the three body parts such as face, body, and hands into separate latent spaces, while leveraging cross-attention to model their correlations. This design ensures

Model Ablation Study				
Model	**corr ↑**	**mpc ↑**	**vel_al ↑**	**acc_al ↑**
SpeechCAT	**0.618**	**0.732**	**0.881**	**2.761**
w/o cross-att.	0.429	0.623	0.853	2.674
One-code	0.349	0.577	0.844	2.554

Table 3. Ablation study results. Metrics include correlation (corr), mean phase coherence (mpc), velocity alignment (vel_al), and acceleration alignment (acc_al).

improved motion diversity, stability, and robustness, further enhanced by the transformer-based motion generator.

4.3. Qualitative Analysis

An example of generated full-body motion can be seen in Figure 3. Given words that are accentuated in a sentence, i.e. having a strengthening tone, the model can correctly express the given tone through coordinated body motion. For the expression "high initial collisions" which is followed by a pause in speech to attract the listener's attention, the model generates a body that lifts its hands in front with open palms. This is a plausible motion as it highlights and complements the speech pause.

A similar motion is generated for the word "low", which is also used as a strengthening tone. The hands are now lifted beforehand and slowly put down to accentuate the word. This is a natural expression that complements the intonation as well as the rhythm of the audio. In Figure 4 we can see a visual comparison between SpeechCAT and TalkSHOW. The TalkSHOW model has a very conservative approach with low movement by always keeping the hands close to the body. SpeechCAT generates a more expressive body motion with, more natural movement. The motions are not only correlated to each other but also with the speech itself. The correlation creates a more expressive motion which better facilitates audio.

4.4. Model Ablation

For the audio-to-motion generation task, aligning different body parts is crucial. To evaluate the synchronization among the face, body, and hand motions, we show the results in Table 3 using the BPSS metrics.

Effect of body part cross-attention We investigate the impact of the body part cross-attention mechanism on the quality of the generated motion. To understand how cross-attention affects the synchronization and cohesiveness of generated movements, we compare two versions of our model: one with the body part cross-attention module enabled and another without it. From Table 3 we can see that cross-attention significantly improves the correlation between body parts. There are also slight increases for all other metrics. Moreover, in Table 1 the improved metrics for cross-attention show better overall robustness. We conclude that adding cross-attention ensures a more reliable, cohesive, and better-synchronized model.

Effect of mapping body parts separately We investigate the impact of using separate VQ-VAEs for each body part i.e., body, hands, and face, compared to using a single VQ-VAE for the entire motion. The hypothesis is that modeling each body part independently might allow the model to capture finer, more nuanced motion details specific to each body part, potentially improving motion realism, synchronization, as well as diversity. From Table 3 we see a decrease in all metrics when using the one-code model, showing that separate VQ-VAEs increase the cohesiveness and synchronization between body parts. However, Table 2 shows that one-code body parts yield a more accurate and speech-aligned motion with the tradeoff of lower diversity. Although the facial expressions are also slightly improved, from Table 2, the higher LVD shows that they are not necessarily better aligned. Finally, Table 1 shows that there are slight improvements across some of the metrics for the one-code model which prompts higher stability. Overall, the one-code approach is slightly more performant in terms of stability than the use of separate VQ-VAEs. It does however fall short in terms of diversity as well as body part synchronization due to the lack of motion choices within the single VQ-VAE.

The results from Table 3 show that the proposed SpeechCAT significantly outperforms the two baselines across all error metrics. Notably, SpeechCAT achieves substantial improvements in cross-correlation and mean phase coherence, which are specifically designed to assess the alignment and correlation among different body parts. These improvements demonstrate the effectiveness of our SpeechCAT in modeling the relationships between the face, body, and hands. In contrast, the one-code baseline yields the poorest performance across all metrics, highlighting the limitations of using a single shared latent space for all body parts. This underscores the advantage of employing separate latent spaces tailored to each body part, as proposed in SpeechCAT. Overall, the combined approach of separate latent spaces integrated with cross-attention in SpeechCAT not only enhances robustness but also ensures better coherence and synchronization across the body parts compared to using standalone VQ-VAEs.

5. Conclusion

In this paper, we propose a novel method, SpeechCAT, for the audio-to-motion generation task. Our approach models the face, body, and hands separately using a three-encoder-decoder architecture, capturing the unique variations in motion for each body part by mapping them into distinct latent spaces. To account for the correlation between these body parts, we introduce a cross-attention mechanism that learns and integrates the correlations among the face, body, and hands. Experimental results demonstrate the effectiveness of the proposed archi-

tecture in generating diverse and coherent motions across all body parts. For future work, we aim to further optimize the model by fine-tuning its parameters, adding Convolution-Augmented Transformers(Conformers)[16], and expanding our comparisons to include additional state-of-the-art methods.

References

[1] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–20, 2023. 2

[2] Tenglong Ao, Qingzhe Gao, Yuke Lou, Baoquan Chen, and Libin Liu. Rhythmic gesticulator: Rhythm-aware co-speech gesture synthesis with hierarchical neural embeddings. *ACM Transactions on Graphics (TOG)*, 41(6):1–19, 2022. 2, 3

[3] Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *ACM Transactions on Graphics (TOG)*, 42(4):1–18, 2023. 2

[4] Uttaran Bhattacharya, Nicholas Rewkowski, Abhishek Banerjee, Pooja Guhan, Aniket Bera, and Dinesh Manocha. Text2gestures: A transformer-based network for generating emotive body gestures for virtual agents. In *2021 IEEE virtual reality and 3D user interfaces (VR)*, pages 1–10. IEEE, 2021. 3

[5] Judith Bütepage, Hedvig Kjellström, and Danica Kragic. Anticipating many futures: Online human motion prediction and generation for human-robot interaction. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4563–4570. IEEE, 2018. 1

[6] Justine Cassell, Hannes Houml;gni Vilhjaacute;lmsson, and Timothy Bickmore. Beat: The behavior expression animation toolkit, 1970. 2

[7] Ju Dai, Hao Li, Rui Zeng, Junxuan Bai, Feng Zhou, and Junjun Pan. Kd-former: Kinematic and dynamic coupled transformer network for 3d human motion prediction. *Pattern Recognition*, 143:109806, 2023. 2

[8] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music, 2020. 2

[9] Sander Dieleman, Aaron van den Oord, and Karen Simonyan. The challenge of realistic music generation: Modelling raw audio at scale, 1970. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and et al. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 2

[11] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2021. 2

[12] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. Faceformer: Speech-driven 3d facial animation with transformers, 2022. 2

[13] Zichen Geng, Caren Han, Zeeshan Hayder, Jian Liu, Mubarak Shah, and Ajmal Mian. Text-guided 3d human motion generation with keyframe-based parallel skip transformer, 2024. 2

[14] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning individual styles of conversational gesture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3497–3506, 2019. 1

[15] Li Gong, Josep Crego, and Jean Senellart. Enhanced transformer model for data-to-text generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 148–156, Hong Kong, 2019. Association for Computational Linguistics. 2

[16] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020. 8

[17] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Lingjie Liu, Hans-Peter Seidel, Gerard Pons-Moll, Mohamed Elgharib, and Christian Theobalt. Learning speech-driven 3d conversational gestures from video. In *Proceedings of the 21st ACM International Conference on Intelligent Virtual Agents*, pages 101–108, 2021. 2

[18] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 1

[19] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023. 3

[20] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes, 2013. 2

[21] Nikos Kolotouros, Thiemo Alldieck, Enric Corona, Eduard Gabriel Bazavan, and Cristian Sminchisescu. Instant 3d human avatar generation using image diffusion models. 2024. 1

[22] Stefan Kopp, Lars Gesellensetter, Nicole C. Krauml;mer, and Ipke Wachsmuth. A conversational agent as museum guide – design and evaluation of a real-world application, 1970. 2

[23] Jaebong Lee, Bohyung Han, and Seungmoon Choi. Motion effects synthesis for 4d films. *IEEE transactions on visualization and computer graphics*, 22(10):2300–2314, 2015. 1

[24] Boren Li, Hang Li, and Hangxin Liu. Driving animatronic robot facial expression from speech, 2024. 2

[25] Jing Li, Di Kang, Wenjie Pei, Xuefei Zhe, Ying Zhang, Zhenyu He, and Linchao Bao. Audio2gestures: Generating diverse gestures from speech audio with conditional variational autoencoders, 2021. 2

[26] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023. 1

[27] Yuanxun Lu, Jinxiang Chai, and Xun Cao. Live speech portraits: real-time photorealistic talking-head animation. *ACM Transactions on Graphics (ToG)*, 40(6):1–17, 2021. 1

[28] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation, 2024. 2

[29] Muhammad Hamza Mughal, Rishabh Dabral, Ikhsanul Habibie, Lucia Donatelli, Marc Habermann, and Christian Theobalt. Convofusion: Multi-modal conversational diffusion for co-speech gesture synthesis. In *CVPR*, pages 1388–1398, 2024. 1

[30] Evonne Ng, Shiry Ginosar, Trevor Darrell, and Hanbyul Joo. Body2hands: Learning to infer 3d hands from conversational gesture body dynamics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11865–11874, 2021. 1

[31] Evonne Ng, Hanbyul Joo, Liwen Hu, Hao Li, Trevor Darrell, Angjoo Kanazawa, and Shiry Ginosar. Learning to listen: Modeling non-deterministic dyadic facial motion. In *CVPR*, pages 20395–20405, 2022. 1

[32] Evonne Ng, Sanjay Subramanian, Dan Klein, Angjoo Kanazawa, Trevor Darrell, and Shiry Ginosar. Can language models learn to listen? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10083–10093, 2023. 1

[33] Evonne Ng, Javier Romero, Timur Bagautdinov, Shaojie Bai, Trevor Darrell, Angjoo Kanazawa, and Alexander Richard. From audio to photoreal embodiment: Synthesizing humans in conversations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1001–1010, 2024. 1

[34] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 2

[35] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *CVPR*, pages 10975–10985, 2019. 1

[36] Wenshuo Peng, Kaipeng Zhang, and Sai Qian Zhang. T3m: Text guided 3d human motion synthesis from speech. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1168–1177, 2024. 2

[37] I. Poggi, C. Pelachaud, F. de Rosis, V. Carofiglio, and B. De Carolis. Greta. a believable embodied conversational agent, 1970. 2

[38] Shenhan Qian, Zhi Tu, Yihao Zhi, Wen Liu, and Shenghua Gao. Speech drives templates: Co-speech gesture synthesis with learned templates. In *ICCV*, pages 11077–11086, 2021. 1

[39] Alec Radford. Improving language understanding by generative pre-training. 2018. 4

[40] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021. 2

[41] Philipp Krähenbühl Sergey Levine. Gesture controllers. 2

[42] Alex Stein, Samuel Sharpe, Doron Bergman, Senthil Kumar, C Bayan Bruss, John Dickerson, Tom Goldstein, and Micah Goldblum. A simple baseline for predicting events with auto-regressive tabular transformers. *arXiv preprint arXiv:2410.10648*, 2024. 4

[43] Mingze Sun, Chao Xu, Xinyu Jiang, Yang Liu, Baigui Sun, and Ruqi Huang. Beyond talking–generating holistic 3d human dyadic motion for communication. *arXiv preprint arXiv:2403.19467*, 2024. 1

[44] Kenta Takeuchi, Souichirou Kubota, Keisuke Suzuki, Dai Hasegawa, and Hiroshi Sakuta. Creating a gesture-speech dataset for speech-based automatic gesture generation, 1970. 2

[45] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 1, 3

[46] Congyi Wang. T2m-hifigpt: Generating high quality human motion from textual descriptions with residual discrete representations. *arXiv preprint arXiv:2312.10628*, 2023. 2, 3

[47] Jiayu Wang, Kang Zhao, Shiwei Zhang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Lipformer: High-fidelity and generalizable talking face generation with a pre-learned facial codebook. In *CVPR*, pages 13844–13853, 2023. 1

[48] Sen Wang, Jiangning Zhang, Weijian Cao, Xiaobin Hu, Moran Li, Xiaozhong Ji, Xin Tan, Mengtian Li, Zhifeng Xie, Chengjie Wang, et al. Mmofusion: Multi-modal co-speech motion generation with diffusion model. *arXiv preprint arXiv:2403.02905*, 2024. 2

[49] Will Williams, Sam Ringer, Tom Ash, John Hughes, David MacLeod, and Jamie Dougherty. Hierarchical quantized autoencoders, 2020. 2

[50] Hongwei Yi, Hualin Liang, Yifei Liu, Qiong Cao, Yandong Wen, Timo Bolkart, Dacheng Tao, and Michael J Black. Generating holistic 3d human motion from speech. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 469–480, 2023. 1, 3, 5

[51] Youngwoo Yoon, Woo-Ri Ko, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Robots learn social skills: End-to-end learning of co-speech gesture generation for humanoid robots, 2018. 2

[52] Youngwoo Yoon, Bok Cha, Joo-Haeng Lee, Minsu Jang, Jaeyeon Lee, Jaehong Kim, and Geehyuk Lee. Speech gesture generation from the trimodal context of text, audio, and speaker identity, 2020. 2

[53] Bowen Zhang, Yiji Cheng, Chunyu Wang, Ting Zhang, Jiaolong Yang, Yansong Tang, Feng Zhao, Dong Chen, and Baining Guo. Rodinhd: High-fidelity 3d avatar generation with diffusion models. In *ECCV*, 2024. 1

[54] Fan Zhang, Naye Ji, Fuxing Gao, and Yongping Li. Diffmotion: Speech-driven gesture synthesis using denoising diffusion model. In *International Conference on Multimedia Modeling*, pages 231–242. Springer, 2023. 2

[55] Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14730–14740, 2023. 1, 5

[56] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model, 2022. 3

[57] Qingcheng Zhao, Pengyu Long, Qixuan Zhang, Dafei Qin, Han Liang, Longwen Zhang, Yingliang Zhang, Jingyi Yu,

and Lan Xu. Media2face: Co-speech facial animation generation with multi-modality guidance, 2024. 2

[58] Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15085–15099, 2023. 2

[59] Wentao Zhu, Xiaoxuan Ma, Dongwoo Ro, Hai Ci, Jinlu Zhang, Jiaxin Shi, Feng Gao, Qi Tian, and Yizhou Wang. Human motion generation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

3

Supplementary material

The supplementary material provides technical explanations of the fundamental concepts and methodologies used in the scientific article in Chapter 2. By focusing on key topics like deep learning, generative models, motion generation techniques, and data models, this section aims to ensure that non-expert readers can fully comprehend the scientific article.

The topics covered include the fundamental ideas of Deep Learning [17] and Neural Networks, a principle of modern artificial intelligence, followed by an exploration of Generative Models [16], which are algorithms responsible for generating new data. We then explore the Vector-Quantized Variational Autoencoder (VQ-VAE) [25], a widely used generative framework for tasks that involve the creation of images and videos. Transformers [29], a crucial architecture that has revolutionized sequence modeling tasks, is then introduced in the following section. We go deeper into this architecture by showing how it can be used for Text Generation and Speech-to-motion tasks using Next-Token Prediction, the main mechanism behind sequence modeling tasks.

Lastly, we describe the datasets used in the scientific study as well as parametric and non-parametric 3D human mesh models, including the SMPL Model [20]. These offer the data-driven and structural basis for human motion modeling. We then explain how these techniques allow for the synthesis of synchronized, natural human motion from sequential inputs by using this data to link the concepts from the Text-to-Motion domain and then apply them to Speech-To-Motion generation. Together, these topics form a cohesive framework that equips readers with the necessary knowledge to engage with the scientific content in detail.

3.1. Deep Learning

3.1.1. Neural Networks

Deep Learning (DL) is a subset of machine learning inspired by the structure and function of the human brain. It enables machines to learn complex patterns and representations from data, making it pivotal in applications like computer vision, natural language processing, and speech recognition.

Neural networks are the foundation of deep learning. They are composed of layers of interconnected nodes called neurons. A Neuron receives an input vector $x = (x_1, x_2, \dots x_n)$ where its elements are each connected with a corresponding weight w_j. The neuron calculates a weighted sum of its inputs, adds a bias term b, and applies an activation function f to produce an output y:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{3.1}$$

From the input x the data flows through each layer of neurons to produce a prediction. This process is called **forward propagation** and can be seen in Figure 3.1. A neuron layer can then be defined as

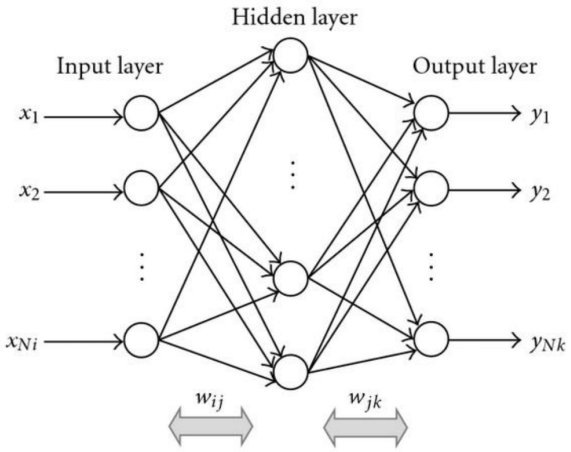


Figure 3.1: A simple neural network architecture, consisting of an input layer, hidden layer, and output layer [31].

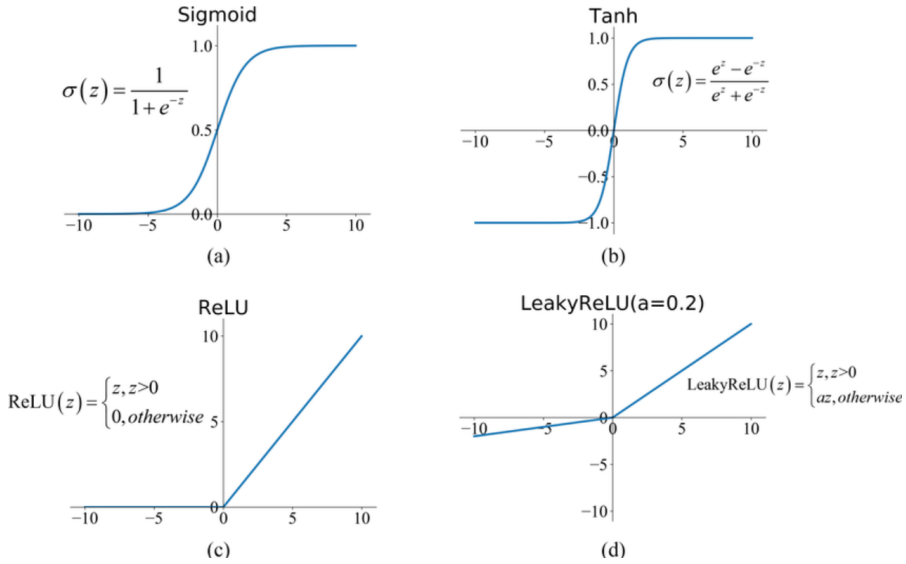


Figure 3.2: Different activation functions.

$a^{(l)} = f(W^{(l)}a^{(l-1)} + b^{(l)})$, where $a^{(l)}$ are the activations(outputs) of layer l, W is the weight matrix, b the bias vector and f the activation function.

The goal of training a neural network is to approximate a target function f^* by iteratively updating the network's parameters (θ, which include weights and biases). The network learns a mapping $y = \hat{f}(x, \theta)$, where the predicted output $\hat{y}$ should closely match the true output y. During training, the network minimizes a loss function that quantifies the error between $\hat{y}$ and y. Through forward propagation, the network computes predictions, while backpropagation adjusts the parameters using the gradients of the loss function. This optimization process ensures that $\hat{f}(x, \theta)$ converges towards f^*, improving the network's ability to generalize and make accurate predictions.

However, because of the weighted sum operation in Equation (3.1), each neuron layer in this format can only map linear relationships. On the other hand, most real-world events are non-linear. Simply altering the activation function of neurons in neural networks allows us to add non-linearities. Figure 3.2 contains common activation functions.

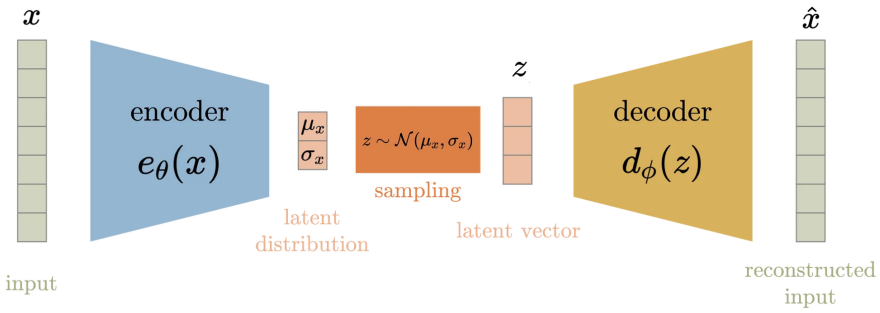


Figure 3.3: Example of a Variational Auto Encoder architecture

3.1.2. Generative models

A subclass of machine learning models known as generative models is made to produce fresh data samples that closely resemble the distribution of an existing dataset. Generative models seek to model the underlying data distribution $p(x)$. In contrast, discriminative models learn the boundary between various classes directly. Image synthesis, text generation, and audio-to-motion generation are just a few of the many uses of generative models.

Mathematical Foundation

Generative models work by approximating the true data distribution $p_{\text{data}}(x)$ using a model distribution $p_\theta(x)$, parameterized by θ. The goal is to learn θ such that $p_\theta(x)$ closely resembles $p_{\text{data}}(x)$. This can be achieved by explicitly modeling $p_\theta(x)$ and maximizing the likelihood of the data:

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p_\theta(x^{(i)}) \tag{3.2}$$

Or by generating samples directly without explicitly defining $p_\theta(x)$, as seen in Generative Adversarial Networks (GANs) [13].

In recent years, deep learning approaches have led to advanced explicit models like Autoregressive models (e.g., PixelRNN [21]) and Flow-based models (e.g., NICE [8], RealNVP [9]). These models offer more flexible and powerful ways to model data distributions.

Types of explicit Generative Models

Variational Autoencoders (VAEs)[19]

An autoencoder network is composed of two parts: an encoder and a decoder. The encoder maps the input data x to a compressed, dense representation, often called the latent space. The decoder then reconstructs the original input x from this latent representation. While traditional autoencoders can learn meaningful encodings, their latent space is not explicitly designed for generation. Thus, it may not be continuous, or allow easy interpolation. The lack of a structured latent space makes it difficult to generate new samples that closely follow the original data distribution.

Variational Autoencoders (VAEs) address this limitation by explicitly designing the latent space to be continuous and probabilistic. Instead of encoding x into fixed points in the latent space, VAEs encode x into a distribution over the latent space $z \sim \mathcal{N}(\mu, \sigma^2)$, where z is a latent vector sampled from the learned distribution. This enables smooth interpolation and generation of new samples from the latent space.

Training a VAE requires optimizing a specific objective function. The total loss for a VAE is the **Evidence Lower Bound (ELBO)**, which consists of the reconstruction loss and the Kullback–Leibler divergence(KL divergence). This can be seen in Equation (3.3). The reconstruction loss term ensures that the decoder reconstructs the input x accurately from the latent variable z. The KL divergence term ensures that the latent variable distribution $q_\phi(z|x)$ is close to a prior distribution $p(z)$, typically chosen as a standard Gaussian $\mathcal{N}(0, 1)$. This encourages the latent space to be smooth and continuous, allowing for meaningful sampling and interpolation.

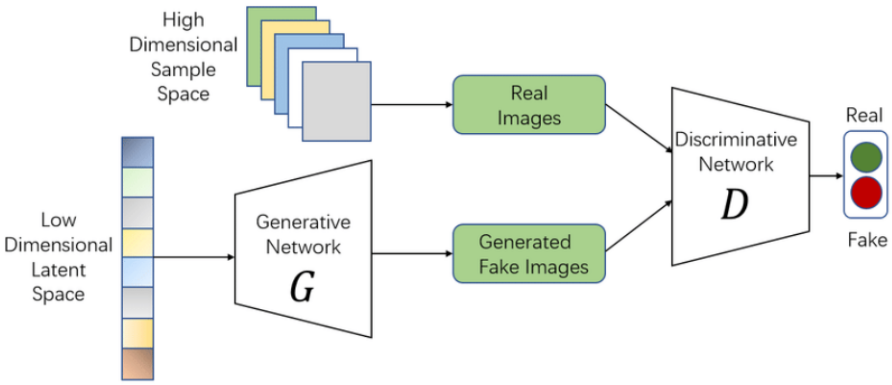


Figure 3.4: Example of a GAN architecture.

$$\mathcal{L}(\theta, \phi) = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{Reconstruction Loss}} - \underbrace{D_{KL}(q_\phi(z|x)\|p(z))}_{\text{KL Divergence}} \tag{3.3}$$

By introducing the KL divergence term, VAEs map the input data x into a structured latent space where points are distributed according to a standard Gaussian prior. This design enables the decoder to generate realistic new data by simply sampling random points from $\mathcal{N}(0, 1)$.

Generative Adversarial Networks (GANs)[13]

Generative Adversarial Networks (GANs) are a class of generative models designed to generate realistic data samples by framing the learning process as a game between two competing networks: the generator and the discriminator. GANs have revolutionized generative modeling and are widely used for tasks such as image synthesis, text-to-image generation, and data augmentation.

The key idea behind GANs is to train two neural networks simultaneously in an adversarial setting. The generator takes random noise (z) as input and generates synthetic samples ($G(z)$). Its goal is to produce data that is indistinguishable from the real data distribution. In comparison, the discriminator takes both real data (x) and synthetic data ($G(z)$) as input and outputs a probability indicating whether the input is real (1) or fake (0). This can be seen in Figure 3.4. Its goal is to correctly distinguish real data from fake data. They are both trained in a min-max game. The generator tries to fool the discriminator by making its outputs as realistic as possible and the discriminator tries to become better at identifying real data from fake data.

The training of GANs is formulated as a zero-sum game with the following objective:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.4}$$

Here $p_{\text{data}}(x)$ is the true data distribution and $p_z(z)$ is the prior distribution of the random noise input of the generator (e.g., a standard normal distribution $\mathcal{N}(0, 1)$). $D(x)$ is the probability assigned by the discriminator that x is real and $G(z)$ is the synthetic data generated by the generator. Here, The generator seeks to fool the discriminator by minimizing $\mathcal{L}_G$ (Equation (3.5)), while the discriminator seeks identify real data by minimizing $\mathcal{L}_D$ (Equation (3.6)).

$$\mathcal{L}_D = -\left(\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]\right) \tag{3.5}$$

$$\mathcal{L}_G = -\mathbb{E}_{z\sim p_z(z)}[\log D(G(z))] \tag{3.6}$$

Diffusion Models

Diffusion models are a class of generative models that use a forward and reverse diffusion process to generate data by gradually refining noisy samples into high-quality outputs. They have gained significant attention recently for their ability to produce high-quality images, audio, and other forms of data.

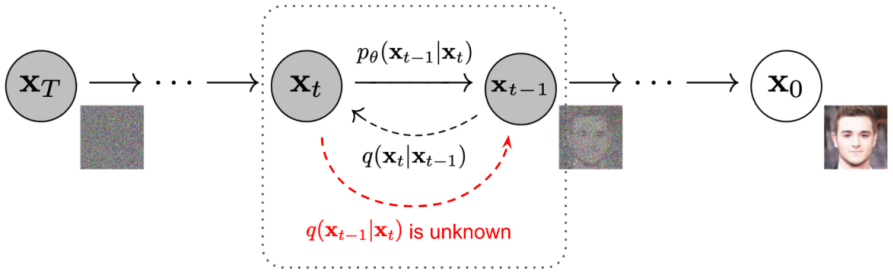


Figure 3.5: Example of diffusion model architecture[18].

They are inspired by physical processes like the diffusion of particles in gases, where a system evolves from an ordered state to a completely random state. These models reverse this process: they take random noise and iteratively transform it into meaningful data by learning the underlying data distribution.

As shown in Equation (3.7), diffusion models have two key components: forward diffusion and reverse diffusion. The forward diffusion process gradually adds noise to the data over several time steps $t = 1, \ldots, T$, transforming the original data x_0 into a pure Gaussian noise x_T. Mathematically, the forward process is defined as $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$, where β_t is a small variance that determines the amount of noise added at each step. Reverse Diffusion Process learns to denoise the noisy data x_t back to the original x_0. The reverse transition is parameterized by a neural network $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta^2(t)I)$. The training objective is to minimize the difference between the true forward process and the learned reverse process. This is typically formulated as a simplified loss function to predict the added noise ϵ as in Equation (3.7)

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] \tag{3.7}$$

Here, ϵ represents the true noise added during the forward process, and ϵ_θ is the model's prediction of that noise.

3.1.3. Vector-Quantized Variational Auto Encoder

Vector Quantized Variational Autoencoders (VQ-VAEs) are a variant of variational autoencoders that incorporate discrete latent representations instead of continuous ones. In many real-world scenarios, discrete representations align more naturally with data. For instance, many objects and concepts, such as "Cat," "Car," or "Tree," are inherently discrete. Interpolating between these categories in a continuous latent space often lacks semantic meaning. Moreover, discrete latent spaces are easier to model because each category corresponds to a single, fixed value. In contrast, continuous latent spaces require normalization of the density function and learning dependencies between variables, which can be computationally expensive and complex.

At a high level, VQ-VAEs retain the basic encoder-decoder structure of traditional autoencoders, with an added discrete codebook in the latent space. Thus, instead of encoding x into a distribution over the latent space $z \sim \mathcal{N}(\mu, \sigma^2)$, where z is a latent vector, VQ-VAEs use discrete latent variables. The distributions are now categorical, and the drawn samples return integral index values. These indexes retrieve learned embeddings from an index dictionary called codebook. Rather than directly using the embeddings, the indexed values are now passed to the decoder.

More specifically, the encoder maps the input x into a latent representation $z_e(x)$. Here, instead of directly using the continuous latent vector $z_e(x)$, VQ-VAEs map it to the nearest vector in a fixed set of discrete embeddings (the codebook). The quantized representation is denoted as $z_q(x)$. The decoder then reconstructs the input x from the quantized latent representation $z_q(x)$. The vector quantization step introduces the discrete latent space by selecting the closest embedding vector e_k from the codebook $\{e_1, e_2, ..., e_K\}$ for each encoded vector $z_e(x)$. This is done by using Equation (3.8) and is represented in Figure 3.6. This quantized vector $z_q(x)$ replaces the continuous latent vector $z_e(x)$ in the reconstruction process.

$$z_q(x) = \text{argmin}_{e_k \in \{e_1, e_2, \ldots, e_K\}} \|z_e(x) - e_k\|^2 \tag{3.8}$$

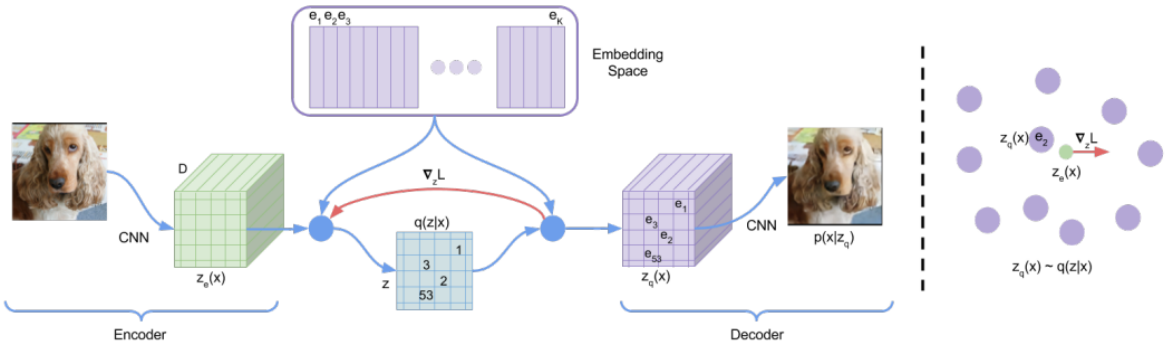


Figure 3.6: Left: a VQ-VAE architecture. Right: The encoder output $z(x)$ is mapped to the nearest point e_2 [22].

Unlike traditional VAEs, VQ-VAEs create a discrete latent space that is more suitable for tasks requiring symbolic or structured representations (e.g., text, audio, or discrete motion indexes). The discrete representations are compact and interpretable, making them ideal for tasks like clustering or generating diverse outputs. Thus, by using discrete codebook entries, VQ-VAEs eliminate the need for a probabilistic prior, avoiding the blurry reconstructions often seen in VAEs. Because of this, VQ-VAEs are especially used in generative tasks like image synthesis [27], audio synthesis [1], motion generation [35], and inpainting [25].

3.1.4. Transformers

Transformers are a class of deep learning models that revolutionized the field of machine learning, particularly in natural language processing (NLP), by introducing a mechanism called self-attention. First introduced in [30], transformers have since become the backbone of many state-of-the-art models like BERT[7], GPT[3], and their derivatives. Their ability to efficiently process sequential data makes them applicable to a wide range of tasks beyond NLP, including image and audio processing, and even generative tasks.

The transformer architecture is fundamentally built on the **self-attention** mechanism, which allows the model to focus on different parts of the input sequence when making predictions. This mechanism assigns a score to each element in the sequence, indicating its importance in understanding other elements. The input of a transformer is represented as a sequence of vectors, often derived from embedding layers (e.g., word embeddings for text or patch embeddings for images). For each input vector, self-attention is computed as shown in Equation (3.9). Where Q, K, V are the Query, Key, and Value matrices derived from the input and d_k is the dimensionality of the keys (used for scaling). Finally, The softmax function ensures that attention scores sum to 1. Here the Query Q represents what the model is looking for. The Key K shows what information is available in the sequence and V is the content retrieved based on the attention scores. This mechanism enables the transformer to capture dependencies between tokens in a sequence, regardless of their distance.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.9}$$

As shown in Figure 3.7, the transformer architecture is composed of two main components: the encoder and the decoder. While some applications, such as BERT, use only the encoder, others, like GPT, rely solely on the decoder. The encoder processes the input sequence and extracts contextual representations for each token. The decoder generates output sequences (e.g., translated sentences or predicted text). Both use self-attention layers alongside positional encodings to process sequences. Since the input embeddings do not contain information regarding their order in the sequence, positional encodings are used to incorporate it directly. These create a unique encoding representing the positions within a sequence. These encoded positions are then added to the input embeddings to inject the positional information directly within the attention mechanism. The positional encoding for a token at position pos in the sequence is given by Equation (3.10). Here, i represents the dimension index in

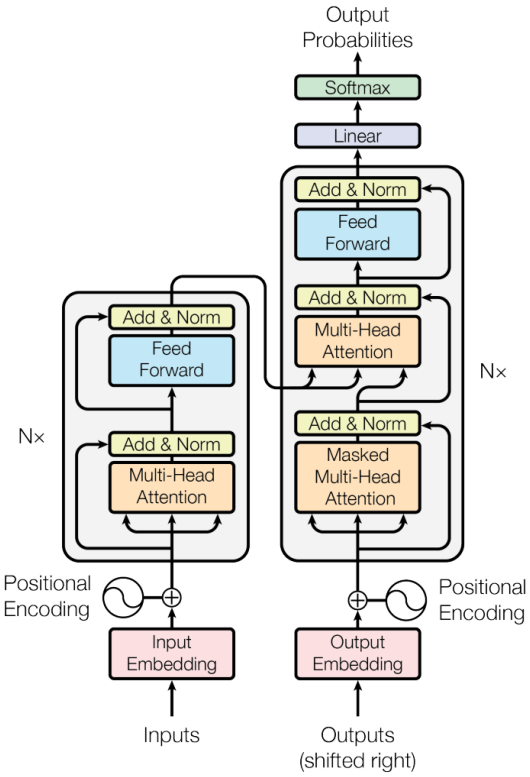


Figure 3.7: Example of a transformer architecture [30].

the embedding vector (half the dimensions are assigned to sine and the other half to cosine) and d_{model} is the dimensionality of the embedding space (e.g., 512 or 1024).

$$\begin{aligned} PE(pos, 2i) &= \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \\ PE(pos, 2i+1) &= \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \end{aligned} \tag{3.10}$$

Transformers are typically trained using a loss function specific to the task. For instance, in language modeling (Decoder-only models like GPT) where we aim for next token predictions, the aim is to maximize the likelihood of the next token x_i given the previous tokens, as shown in Equation (3.11).

$$\mathcal{L} = -\sum_{i=1}^{N} \log p(x_i | x_{<i}) \tag{3.11}$$

In comparison, masked language modeling (Encoder-only models like BERT) tasks aim to predict masked tokens based on the context. This is particularly used for question-answering models. Such an objective function is described in Equation (3.12).

$$\mathcal{L} = -\sum_{i \in \text{masked}} \log p(x_i | x_{\setminus i}) \tag{3.12}$$

Besides these tasks, transformers have multiple applications such as natural language processing: machine translation [30] (e.g., Google Translate), text generation [3] (e.g., GPT-3, ChatGPT), sentiment analysis [7] (e.g., BERT); Vision Transformers (ViTs) [10] for image recognition; Audio transformers [15] for tasks like speech recognition and synthesis and text-to-image generation [26] (e.g. DALL-E).

3.2. Computer Vision

Computer Vision is a field of artificial intelligence that enables machines to interpret and analyze visual data, such as images and videos. It has applications in object detection[36], image classification[34],

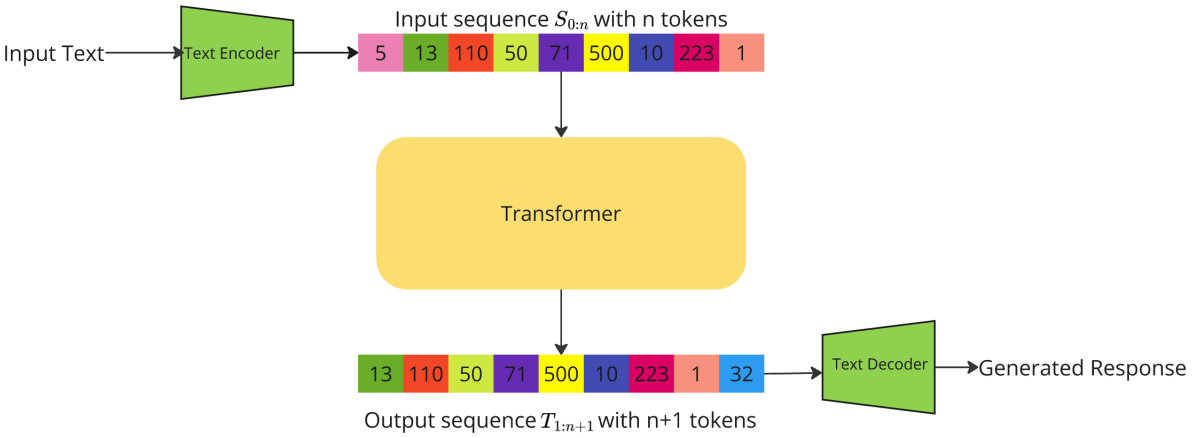


Figure 3.8: Transformer architecture on a next index prediction task. From the input $S_{0:n} = (s_0, \ldots, s_n)$ the transformer generates the next indexes as the sequence $S_{1:n+1} = (s_1, \ldots, s_{n+1})$.

etc. In this context, computer vision intersects with natural language processing and motion generation, enabling tasks like text-to-motion and speech-to-motion generation.

3.2.1. Next token prediction and text generation

Next-token prediction is a fundamental task in natural language processing (NLP) that has paved the way for token generation models. Given the preceding context, it involves predicting the next word or token in a sequence. This concept is widely applied in text completion, translation, dialogue systems, etc.

Autoregressive Models like GPT (Generative Pre-trained Transformers) generate responses by repeatedly predicting the sequence's most probable token. A sentence is represented as a sequence in which words are encoded into tokens, thus the task of next token prediction refers to predicting the next most probable word in a GPT response. For an input sequence of indexes $S_{0_\tau}^{true} = (s_0, \ldots, s_\tau)$, the model aims to generate the next index for each item in the sequence, thus having the output $S_{1_\tau+1}^{generated} = (s_1, \ldots, s_{\tau+1})$, as shown in Figure 3.8. The objective is to minimize the cross-entropy loss between the predicted and true token sequences, thus comparing $S_{1_\tau+1}^{true}$ with $S_{1_\tau+1}^{generated}$.

At inference, when generating responses, for each step the model outputs a probability distribution over the vocabulary for the next token, conditioned on the previous tokens $P(x_{t+1}|x_1, x_2, \ldots, x_t)$. The highest probability word is chosen, and the process is repeated iteratively. This is how responses are generated for large language models that use transformers, such as GPTs. Next-token prediction techniques are foundational for models in text-to-motion and speech-to-motion tasks, as these also rely on predicting sequences of tokes, such as encoded motion trajectories or poses.

3.2.2. Text-to-motion

Text-to-motion generation involves creating realistic human motion sequences directly from textual descriptions. This task lies at the intersection of natural language processing and computer vision.

To generate motion, we first take our text sequence and encode it using NLP models, such as Transformers, to capture semantic and contextual meaning. The result $S_{0_\tau}^{true} = (s_0, \ldots, s_\tau)$ is then mapped to the same dimension as the motion and collapsed(averaged) into a single token S_{avg}. The motion data is also encoded into tokens using a VQ-VAE as $M_{0_\tau}^{true} = (m_0, \ldots, m_\tau)$. Autoregressive models can now use next token prediction to generate the next motions within the sequence. This is done by generating the output probability distribution $P(m_{t+1}|S_{avg}, m_1, m_2, \ldots, m_t)$. Thus, we use the averaged speech information as the first token and then iteratively generate the motion tokens from it.

3.2.3. Speech-to-motion

Speech-to-motion generation focuses on synthesizing human-like motion sequences from speech input. This involves understanding the rhythm, tone, and semantics of the speech to generate synchronized

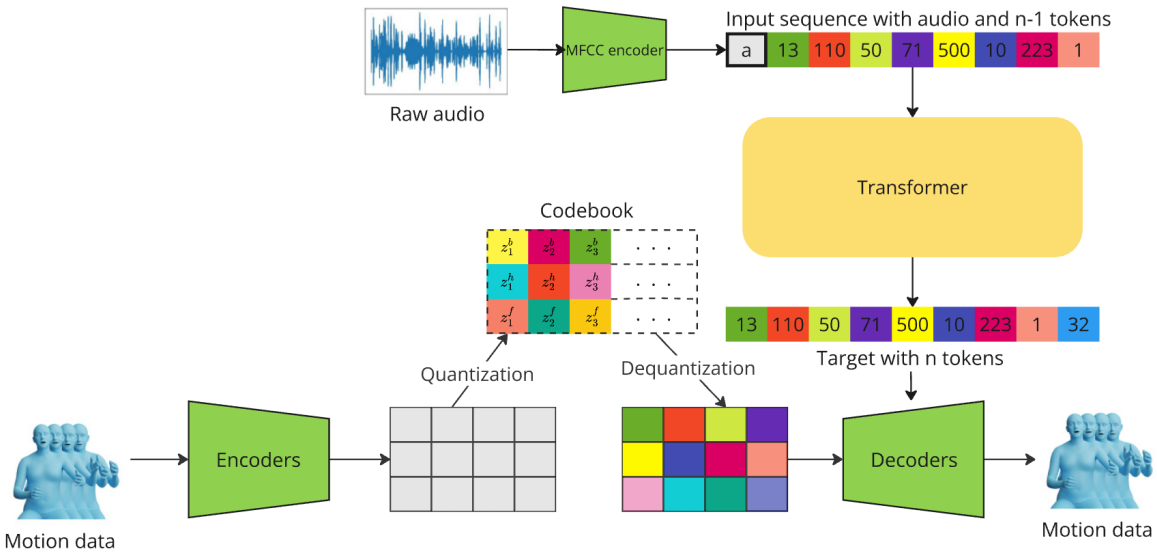


Figure 3.9: Simplified SpeechCAT architecture. The bottom represents the motion encoding using a VQ-VAE. The top part represents the motion generation where audio is used with the motion sequence to create the input $M_{0:n} = (S_{avg}, m_1, m_2, \ldots, m_n)$. The generated transformer sequence is $M_{1:n} = (m_1, m_2, \ldots, m_{n+1})$. The generated sequence is then decoded in motion data using the VQ-VAE decoder. A similar approach can be used for text-to-motion by changing the speech encoder.

and expressive motion.

As shown in Figure 3.9, speech-to-motion is a similar task to text to motion, the only difference being the input. From the input audio data, audio features are extracted using methods like MFCCs (Mel-frequency coefficients) or deep models like Wav2Vec [2]. These are then processed into tokens using VAEs or Audio transformers(Conformers) [15]. Tokens are then collapsed and the motion prediction process is now the same as in text-to-motion. This process reduces complex tasks such as speech or text into motion generation to already known concepts such as next index prediction.

Although straightforward, this process of translating to next index prediction poses its own challenges. First, you need to ensure the generated motion matches the rhythm and intonation of speech. This requires a very descriptive audio encoder that understands the rhythm, tone, and semantics of the speech. Second, the motion needs to be diverse and coherent. The VQ-VAE used to encode motion needs to be large enough to generate diverse motion indexes and keep the motions human-like. Finally, temporal coherence needs to be ensured across motion sequences by the autoregressive transformer model.

3.3. 3D human mesh models

3D human mesh models are mathematical representations of the human body used to capture its geometry and motion in three dimensions. They consist of a mesh, which is a collection of vertices and edges that form a 3D surface, and additional parameters to define the pose, shape, or motion of the body. These models are widely used in applications such as animation, motion capture, virtual reality, and human-computer interaction. 3D human mesh models can be broadly classified into two categories: non-parametric and parametric.

3.3.1. Non-Parametric models

Non-parametric models rely directly on raw data to represent the human body without introducing predefined constraints or parameters. These models are typically obtained through techniques like 3D scanning or photogrammetry, which capture detailed surface geometry from real-world humans. From input data, non-parametric models are usually defined as dense point clouds or meshes. This can be seen in Figure 3.10. A mesh is represented by a set of vertices $\{v_1, v_2, \ldots, v_n\}$ and faces $\{f_1, f_2, \ldots, f_m\}$, which form the surface of the human body.

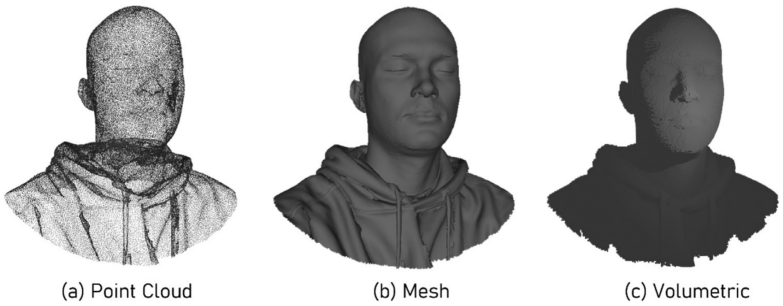


Figure 3.10: Different representations of a non-parametric mesh [5].

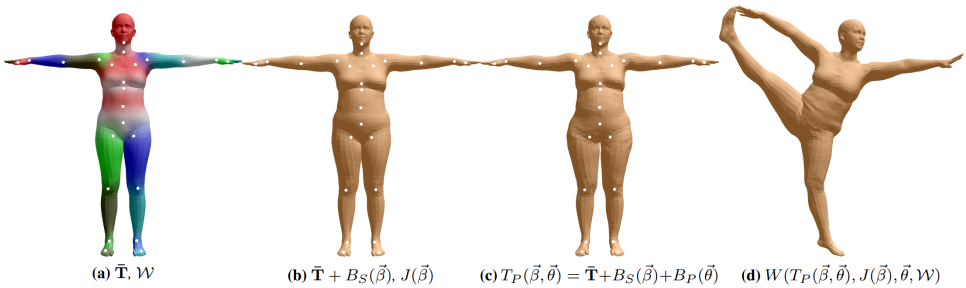


Figure 3.11: (a) Default template mesh. (b) Added identity-contribution to blend shape(ex. malefemale). (c) Addition of pose blend shapes; note the expansion of the hips. (d) Deformed vertices reposed by skinning for the split pose.[20]

Due to their extremely detailed and accurate representations, non-parametric meshes are used in high-precision applications such as medical imaging, 3D scanning, and photorealistic rendering. Although highly performant in capturing fine-grained surface details like wrinkles and clothing, non-parametric meshes have a high computational cost, lack of semantic control over pose and shape, and are difficult to generalize to new poses or body shapes.

3.3.2. Parametric models: SMPL

Parametric models introduce a structured and low-dimensional representation of the human body by leveraging statistical priors. SMPL (Skinned Multi-Person Linear model) is one of the most widely used parametric human mesh models [20].

SMPL represents the human body as a predefined mesh with $n = 6890$ vertices and is parameterized by shape and pose parameters. **Shape Parameters** (β) capture individual body shape variations (e.g., height, weight). **Pose Parameters** (θ) encode joint rotations for different poses.

Mathematically, the model generates a mesh $M(\beta, \theta)$ as:

$$M(\beta, \theta) = W(T(\beta, \theta), J(\beta), \theta, W) \tag{3.13}$$

Here, $T(\beta, \theta)$ is the template mesh deformed by β (shape) and θ (pose). $J(\beta)$ are joint locations determined by shape parameters and W are linear blend skinning weights. Linear blend skinning is a technique used to remove discontinuity by linearly blending vertices near the joint. This makes smooth transitions around the skin of rotating bone joints. This is also shown in Figure 3.11

Although body detail and expressiveness is reduced compared to the non-parametric models, SMPL has a compact and efficient representation that generalizes well across different poses and body shapes. SMPL can be easily integrated into computer vision pipelines as it can be defined parametrically, which reduces data volume and computation. Currently, SMPL is a popular model used for multiple tasks such as motion capture and retargeting, virtual avatars, and 3D pose estimation. Provides a balance between computational efficiency and realism. This makes it easy to train and use with standard datasets.

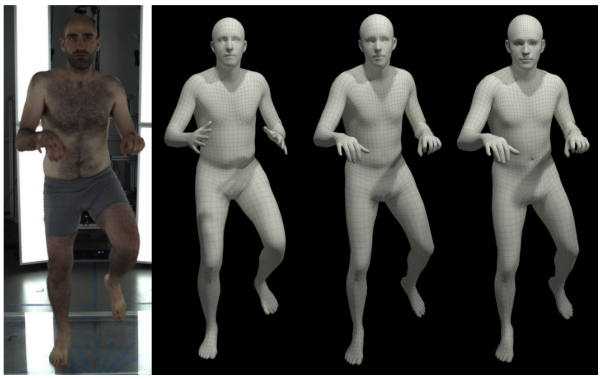


Figure 3.12: Comparison of SMPL (left), SMPL+H (middle) and SMPL-X (right). There is a clear increase in expressiveness from left to right. This is directly correlated with the model getting richer: from body-only (SMPL) to include hands (SMPL+H) or hands and face (SMPL-X) [24].

3.3.3. Other models

Several other 3D human mesh models exist, each designed for specific applications or with unique features. These include

1. **SMPL-X** [24]: As shown in Figure 3.12 it extends SMPL by adding parameters for hands and facial expressions. Suitable for full-body modeling, including gestures and facial expressions. SpeechCAT also uses this model.
2. **STAR (Sparse Trained Articulated Human Body Regressor)** [23]: A simplified alternative to SMPL with faster computation while maintaining comparable accuracy.
3. **Body-Region Specific Models**: Models like FaceMesh [14] and HandMesh [4] focus on specific body regions for applications requiring high precision.
4. **Neural Implicit Models**: Represent the human body using implicit functions, such as signed distance fields or neural radiance fields (NeRFs) [32]. These models are continuous and can achieve highly detailed reconstructions.
5. **HumanMesh++** [28]: Combines traditional parametric modeling with deep learning techniques to improve realism and flexibility.

3.3.4. Datased used

We utilized the TalkSHOW Dataset [33], a high-quality audiovisual dataset designed specifically for generating holistic 3D human body motions from speech. This dataset stands out for its inclusion of 3D body meshes, reconstructed from in-the-wild video data, along with synchronized audio. These features make it particularly valuable for training and evaluating speech-to-motion generation models.

The TalkSHOW dataset consists of 26.9 hours of annotated video data from four speakers with diverse speaking styles. Synchronized audio is recorded at a 22 kHz sample rate and frames are reconstructed at 30 FPS using the parametric SMPL-X model described in Section 3.3.3. The dataset is divided into short video clips, each less than 10 seconds, making it suitable for mini-batch processing during training. Reconstruction uses SMPL-X parameters to represent the pseudo-ground truth (p-GT), such as body shape parameters $\beta \in \mathbb{R}^{300}$, pose parameters $\theta \in \mathbb{R}^{156}$, facial expressions $\psi \in \mathbb{R}^{100}$ alongside camera poses and translations.

As shown in Table 3.1, TalkSHOW addresses limitations found in other speech-to-motion datasets. Unlike datasets such as VOCASET[6] or BIWI[11], which focus only on head or body motion, TalkSHOW provides a holistic representation of the face, body, and hands. Moreover, it surpasses Speech2Gesture[12] and similar datasets by offering connected 3D body meshes rather than disjoint representations.

Dataset	Head	Hands	Body	Holistic Body	In-the-Wild	Length
VOCASET	3D mesh	✗	✗	✗	✗	4D-scan
Speech2Gesture	✗	2D keypoint	2D keypoint	✗	✓	144 hours
Habibie et al.	3D mesh	3D keypoint	3D keypoint	✗	✓	33 hours
TalkSHOW	3D mesh	3D mesh	3D mesh	✓	✓	27 hours

Table 3.1: Comparison of speech-to-motion datasets.

References

[1] KTH Royal Institute of Technology, 2021. ISBN: 9789151955605. DOI: 10.30746/978-91-519-5560-5. URL: http://dx.doi.org/10.30746/978-91-519-5560-5.

[2] Alexei Baevski et al. “wav2vec 2.0: A framework for self-supervised learning of speech representations”. In: *Advances in neural information processing systems* 33 (2020), pp. 12449–12460.

[3] Tom B Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in Neural Information Processing Systems* 33 (2020). URL: https://arxiv.org/abs/2005.14165.

[4] Xingyu Chen et al. “Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13274–13283.

[5] Helena A Correia and Jose Henrique Brito. “3D reconstruction of human bodies from single-view and multi-view images: A systematic review”. In: *Computer Methods and Programs in Biomedicine* 239 (2023), p. 107620.

[6] Daniel Cudeiro et al. “Capture, learning, and synthesis of 3D speaking styles”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 10101–10111.

[7] Jacob Devlin et al. “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*. 2019. URL: https://arxiv.org/abs/1810.04805.

[8] Laurent Dinh, David Krueger, and Yoshua Bengio. *NICE: Non-linear Independent Components Estimation*. 2015. arXiv: 1410.8516 [cs.LG]. URL: https://arxiv.org/abs/1410.8516.

[9] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. *Density estimation using Real NVP*. 2017. arXiv: 1605.08803 [cs.LG]. URL: https://arxiv.org/abs/1605.08803.

[10] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations* (2021). URL: https://arxiv.org/abs/2010.11929.

[11] Gabriele Fanelli et al. “Random Forests for Real Time 3D Face Analysis”. In: *Int. J. Comput. Vision* 101.3 (Feb. 2013), pp. 437–458.

[12] S. Ginosar et al. “Learning Individual Styles of Conversational Gesture”. In: *Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019.

[13] Ian Goodfellow et al. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[14] Ivan Grishchenko et al. “Attention mesh: High-fidelity face mesh prediction in real-time”. In: *arXiv preprint arXiv:2006.10962* (2020).

[15] Anmol Gulati et al. “Conformer: Convolution-augmented Transformer for Speech Recognition”. In: *Proceedings of Interspeech 2020*. 2020. URL: https://arxiv.org/abs/2005.08100.

[16] GM Harshvardhan et al. “A comprehensive survey and analysis of generative models in machine learning”. In: *Computer Science Review* 38 (2020), p. 100285.

[17] Jeff Heaton. “Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618”. In: *Genetic programming and evolvable machines* 19.1 (2018), pp. 305–307.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[19] Diederik P Kingma, Max Welling, et al. “An introduction to variational autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[20] Matthew Loper et al. “SMPL: A skinned multi-person linear model”. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 851–866.

[21] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. *Pixel Recurrent Neural Networks*. 2016. arXiv: 1601.06759 [cs.CV]. URL: https://arxiv.org/abs/1601.06759.

[22] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu koray. “Neural Discrete Representation Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

[23] Ahmed AA Osman, Timo Bolkart, and Michael J Black. “Star: Sparse trained articulated human body regressor”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*. Springer. 2020, pp. 598–613.

[24] Georgios Pavlakos et al. “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*. 2019, pp. 10975–10985.

[25] Jialun Peng et al. “Generating Diverse Structure for Image Inpainting With Hierarchical VQ-VAE”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 10770–10779. DOI: 10.1109/CVPR46437.2021.01063.

[26] Aditya Ramesh et al. “Zero-Shot Text-to-Image Generation”. In: *Proceedings of the 38th International Conference on Machine Learning*. 2021. URL: https://arxiv.org/abs/2102.12092.

[27] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. *Generating Diverse High-Fidelity Images with VQ-VAE-2*. 2019. arXiv: 1906.00446 [cs.LG]. URL: https://arxiv.org/abs/1906.00446.

[28] Yating Tian et al. “Recovering 3d human mesh from monocular images: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* (2023).

[29] A Vaswani. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* (2017).

[30] Ashish Vaswani et al. “Attention is all you need”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964.

[31] Anjar Wanto et al. “Use of binary sigmoid function and linear identity in artificial neural networks for forecasting population density”. In: *IJISTECH (International Journal of Information System and Technology)* 1.1 (2017), pp. 43–54.

[32] Xinyue Wei et al. “Meshlrm: Large reconstruction model for high-quality mesh”. In: *arXiv preprint arXiv:2404.12385* (2024).

[33] Hongwei Yi et al. “Generating holistic 3d human motion from speech”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 469–480.

[34] Jiahui Yu et al. “Coca: Contrastive captioners are image-text foundation models”. In: *arXiv preprint arXiv:2205.01917* (2022).

[35] Jianrong Zhang et al. *T2M-GPT: Generating Human Motion from Textual Descriptions with Discrete Representations*. 2023. arXiv: 2301.06052 [cs.CV]. URL: https://arxiv.org/abs/2301.06052.

[36] Zhong-Qiu Zhao et al. “Object detection with deep learning: A review”. In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.