Chương 1. OVERVIEW

The gesture generation problem, like other machine learning tasks, has been studied and developed alongside both traditional and modern machine learning methods. These include rule-based and data-driven approaches. This thesis first presents the characteristics of gestures Section 1.1, serving as the foundation for learning the relationship between gestures, text, and speech. In section Section 1.3, the thesis introduces the common stages across different gesture generation methods.

In section Section 1.2, the thesis presents the methods that have been applied in gesture generation. A comparison of these methods is provided, leading to the motivation for adopting diffusion models for this task.

Section Section 1.4 explains how diffusion models have recently been applied to gesture generation.

1.1 Gesture Characteristics

According to linguistics, gestures can be categorized into six main groups: adaptors, emblems, deictics, iconics, metaphorics, and beats [?], [?]. Among them, beat gestures do not carry direct semantic meaning but play an important role in synchronizing rhythm between speech and gesture [?], [?]. However, the rhythm between speech and beat gestures is not fully synchronized, making the temporal alignment between them complex to model [?], [?], [?], [?].

Gestures interact with various levels of information in speech [?]. For instance, emblematic gestures like a thumbs-up are usually linked to high-level semantic content (e.g., "good" or "excellent"), while beat gestures often accompany low-level prosodic features such as emphasis. Previous studies typically extract features from the final layer of the speech encoder to synthesize gestures [?], [?], [?], [?], [?]. However, this approach may blend information from different levels, making it hard to disentangle rhythm and semantics.

As shown in linguistic studies [?], [?], [?], gestures in daily communication can be decomposed into a limited number of semantic units with various motion variations. Based on this assumption, speech features are divided into two types: high-level features representing semantic units and low-level features capturing motion variations. Their relationships are learned across different layers of the speech encoder. Experiments demonstrate that this mechanism can effectively disentangle features at different levels and generate gestures that are both semantically and stylistically appropriate.

1.2 Overview of Gesture Generation Methods

1.2.1 Rule-Based Methods

1.2.1.1 General Principle

These methods rely on clearly defined rules, which are manually crafted to determine how the system processes inputs to produce outputs.

1.2.1.2 Methods

Representative rule-based methods include the *Robot behavior toolkit* [?] and *Animated conversation* [?]. These approaches typically map speech to gesture units using handcrafted rules. Rule-based systems allow for straightforward control over model outputs and provide good interpretability. However, the cost of manually designing these rules is prohibitive for complex applications requiring the processing of large-scale data.

1.2.2 Statistical Methods

1.2.2.1 General Principle

These methods rely on data analysis, learning patterns from datasets, and using probabilistic models or mathematical functions for prediction. The approach involves optimizing model parameters to fit the data.

1.2.2.2 Methods

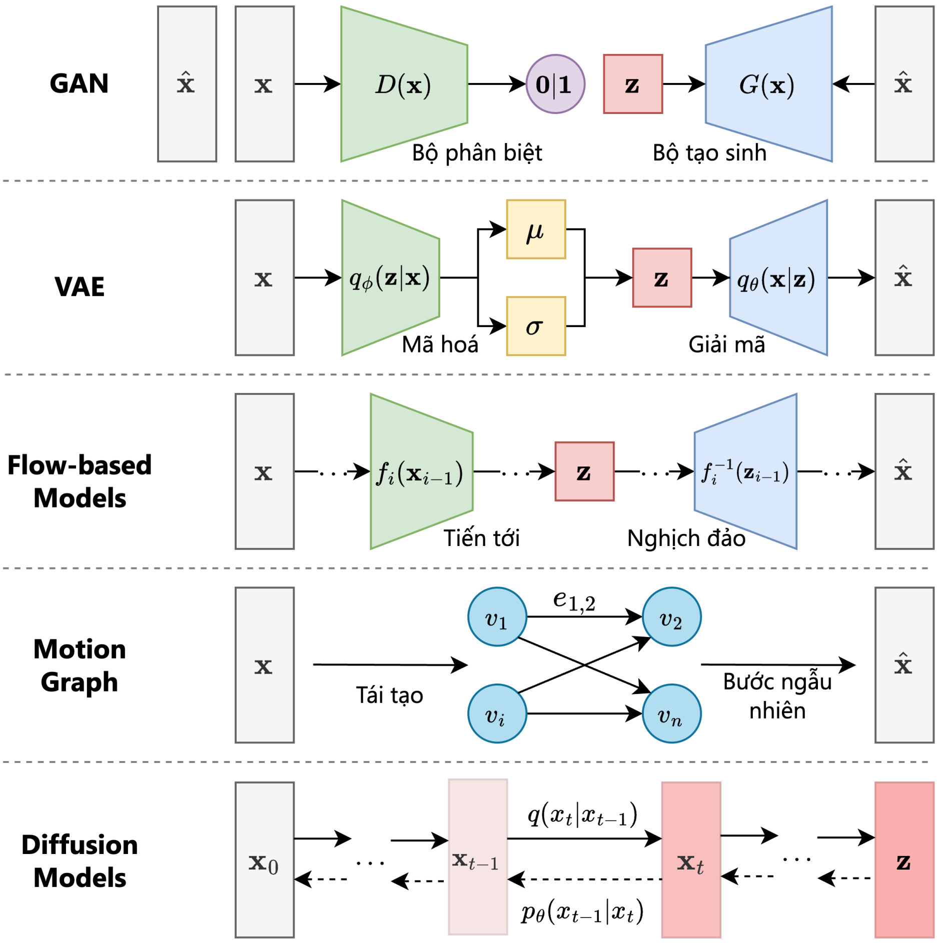
Like rule-based methods, data-driven methods also map speech features to corresponding gestures. However, instead of manual rules, they employ automatic learning based on statistical data analysis.

Representative statistical approaches include *Gesture controllers* [**?**], *Statistics-based* [**?**], which use probabilistic distributions to find similarities between speech and gesture features. *Gesture modeling* [**?**] constructs probabilistic models to learn individual speaker styles.

1.2.3 Deep Learning Methods

General Principle of Deep Learning Methods

These methods utilize multi-layer perceptrons (MLPs) to automatically extract features from raw data and learn complex data representations through parameter optimization.



Hình 1.1: Overview of different generative models.

As illustrated in Figure 1.1, deep learning-based gesture generation methods can be divided into two main groups: likelihood-based models and implicit generative models [?].

1.2.3.1 Likelihood-Based Models

General Principle of Likelihood-Based Models

These models work by maximizing the likelihood of observed data given model parameters θ. The objective is to find the optimal parameters θ' by modeling the probability $p(\mathbf{x})$ of the data, where $\mathbf{x}$ represents the gesture sequence.

Methods

The application of deep learning to gesture generation has evolved alongside the development of deep learning models, including RNNs, LSTMs, and Transformers. Representative likelihood-based methods include:

- *Gesticulator* [**?**], which uses a Multilayer Perceptron (MLP) to encode text and audio features, with BERT-derived vectors used as text features.

- *HA2G* [**?**] builds a hierarchical Transformer-based model to learn multi-level correlations between speech and gestures, from local to global features.

- *Gesture Generation from Trimodal Context* [**?**] uses an RNN architecture and treats gesture generation as a translation task in natural language processing.

- *DNN* [**?**] combines LSTM and GRU to build a classifier neural network that selects appropriate gesture units based on speech input.

- *Cascaded Motion Network (CaMN)* [**?**] introduces the BEAT dataset and a waterfall-like model. Speaker information, emotion labels, text, speech, and gesture features are processed through layers to extract latent vectors. In the fusion stage, CaMN combines features sequentially: speaker and emotion features are merged first, followed by integration with latent vectors of text, speech, and gestures.

- **Motion Graph**: In *Gesturemaster* [**?**], a semantic graph is constructed where words in a sentence are connected based on semantic relationships. The model then selects the most relevant nodes and edges in the graph to represent gestures.

1.2.3.2 Implicit Generative Models

Principle

Implicit generative models learn the data distribution without explicitly modeling the probability density function $p(\mathbf{x})$. Instead of directly computing $p(\mathbf{x})$, the model learns a mapping $G_\theta : \mathcal{Z} \rightarrow \mathcal{X}$ by matching the distributions between real and generated data $\mathbf{x} = G_\theta(\mathbf{z}), \quad \mathbf{z} \sim p_z(\mathbf{z})$. Here, $\mathcal{Z}$ denotes the input noise space, typically with a simple distribution p_z (e.g., Gaussian, Uniform), and $\mathcal{X}$ is the space of real data, which in this case is gesture sequences $\mathbf{x}$.

In gesture generation, to incorporate conditions such as speech or text labels, implicit generative models often introduce a condition $\mathbf{c}$ representing the context of the task, resulting in the conditional generation: $\mathbf{x} = G_\theta(\mathbf{z}, \mathbf{c})$. This context may include speech, text, speaker ID, style, emotion, or initial gestures.

A typical example is the generative adversarial network (GAN) and Diffusion model, where data is synthesized by transforming an initial simple distribution (e.g., Gaussian) into the target data distribution.

Methods

Representative implicit generative models include:

- *MoGlow* [?] uses Normalizing Flows to maintain motion consistency and compatibility, while providing control via input parameters. This allows users to generate new motions or modify existing ones by adjusting control parameters.

- **GAN**: *GRU-based WGAN* [?] utilizes Wasserstein GAN to evaluate and improve the quality of gesture synthesis. The model focuses on optimizing the Wasserstein loss, mitigating mode collapse typically seen in traditional GANs. GRUs process speech data and convert it into usable gesture features, which are then fed into the WGAN for evaluation and refinement.

- **VAE**: *FreeMo* [?] employs a VAE to decompose gestures into pose modes and rhythmic motions. Gesture poses are randomly sampled using conditional sampling in the VAE's latent space, while rhythmic motions are generated from...

- **VQ-VAE**: *Rhythmic Gesticulator* [?] preprocesses speech segments based on beats, dividing speech into smaller parts and representing them as blocks with normalized dimensions. Similar to the VQ-VAE approach, normalized gesture sequences are quantized into discrete gesture lexicons. The model learns the gesture vocabulary conditioned on the previous gesture, gesture style, and speech. It then reconstructs the gesture sequence via denormalization. Unlike generative models like GANs or Diffusion, VQ-VAE focuses more on compression rather than direct generation.

- **Diffusion**: Diffusion models focus on generating new data from noisy inputs by progressively denoising toward the original data. Diffusion-based approaches will be presented in section Section 1.4.

1.2.4 Comparison of Gesture Generation Methods

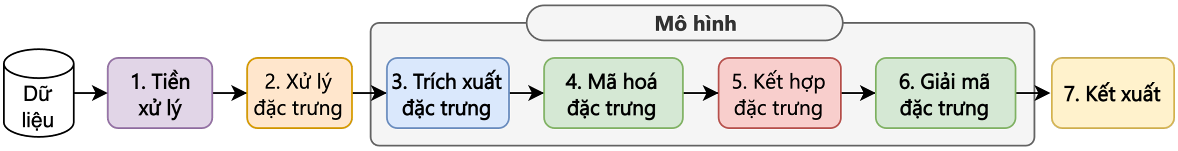
Representative Method	**Advantages**	**Disadvantages**	**Method Type**
Robot behavior toolkit [?]	- Easy to understand and implement. - Highly interpretable and controllable. - Effective for simple cases or small datasets.	- Does not generalize well to complex data. - Labor-intensive rule construction.	Rule-based model
MLP [?], RNN [?], [?], [?], [?], CNN [?], Transformer [?]	- Able to estimate data probability density. - Scalable and learns well from large datasets.	- Sensitive to noise. - Performs poorly on low-frequency data. - Low diversity in generation.	Likelihood-based model
DiffusionStyleGesture [?], MDM [?], Motiondiffuse [?]	- Generates high-quality data. - Flexible and diverse outputs. - Covers low-density regions of data space.	- Requires complex configuration for optimal performance. - Random noise leads to different outputs each time. - Slow sampling process.	Implicit generative model

Bảng 1.1: Comparison of advantages and disadvantages of different methods

1.3 Common Stages in Deep Learning Approaches for Gesture Generation

As presented in **??**, gestures consist of sequences of 3D point coordinates. For each dataset, the number of bones per frame may vary.

Deep learning approaches to gesture generation are implemented using various techniques. However, this thesis generalizes the process into the following main stages, illustrated in Figure 1.2:



Hình 1.2: Common stages in gesture generation models.

1. **Preprocessing**: In the preprocessing stage, data such as speech segments,

gesture sequences, and text are read and digitized into vectors or matrices that represent raw data information. Depending on the specific learning method, the selected initial data features may vary.

2. **Feature Processing**: In this stage, raw data such as speech and text are embedded into feature vectors. Different methods use different embedding models. The way gesture sequences are represented as feature vectors also varies across methods.

3. **Feature Extraction**: This stage uses linear transformation layers or CNN layers to extract features from the data. Text and speech features, after being processed, may be further passed through feature extraction layers to generate representation vectors corresponding to the input modalities.

4. **Feature Encoding**: In this stage, gesture, emotion, and speech vectors are encoded into a lower-dimensional latent space to facilitate learning the correlations among modalities in the feature fusion stage.

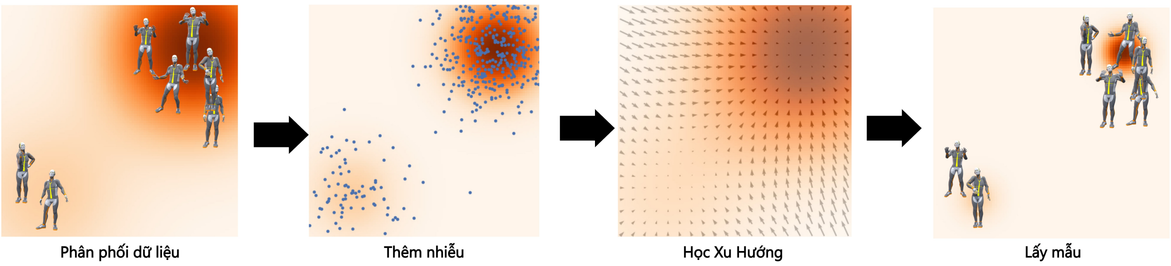
5. **Feature Fusion**: In this stage, features from speech, text, gestures, and other information are combined, typically using concatenation, fully connected layers, or operations such as vector addition or subtraction in the latent space.

6. **Feature Decoding**: In this stage, latent vectors are decoded or upsampled back to their original dimensionality.

7. **Rendering**: Once the output vectors are restored to their original size, they are converted back into BVH files and rendered using software such as Blender or Unity to visualize character motion.

1.4 Diffusion-based Model

1.4.1 General Principle of Diffusion-based Methods



Hình 1.3: Illustration of the general principle of the Diffusion model.

Similar to the methods in the class of Implicit Generative Models (Subsubsection 1.2.3.2), the core principle of Diffusion models is to learn a function f_θ that represents the drift process from a standard normal distribution $\mathcal{N}(0, \mathbf{I})$ to the data distribution $\mathbf{x}_0 \sim p(\mathbf{x})$. Sampling is performed step by step from this reverse process to generate new data, as illustrated in Figure 1.3.

This process is executed over T steps, with noise added at each step gradually reduced so that the model can navigate toward high-density regions of the data distribution.

1.4.2 Typical Methods

1.4.2.1 Typical Methods

- *MotionDiffuse* [?] employs a conditional Diffusion model, with conditions based solely on text, excluding audio. Additionally, the model predicts noise rather than directly predicting the original gesture sequence. MotionDiffuse utilizes Self-Attention and Cross-Attention layers to model the correlation between textual features and gesture features during *Stage 5. Feature Fusion* (Figure 1.2).

- *Flame* [?] applies a Diffusion model with a Transformer-based architecture. In *Stage 2. Feature Processing* (Figure 1.2), it uses the pre-trained RoBERTa model to embed the text into textual feature vectors, which serve as the conditioning input. During *Stage 5. Feature Fusion* (Figure 1.2), the text is used as the CLS token prepended to the gesture sequence before passing through the Transformer Decoder. Similar to other methods, the model predicts the added noise rather than the original gesture sequence.

- *DiffWave* [?] is a noise-predicting Diffusion model in which the time steps pass through multiple Fully Connected layers and a Swish activation function before feature fusion. It uses a dilated convolutional architecture inherited from WaveNet. DiffWave enables better representation of speech, improving the effectiveness of conditioning for the Diffusion model.

- *Listen, Denoise, Action* [?] builds upon DiffWave [?], replacing the dilated convolution layers with a Transformer, and integrating Conformer modules to enhance model performance.

- *DiffSHEG* [?] employs a Diffusion model; in *Stage 2. Feature Processing*, it uses HuBERT to encode the audio signal. The model treats facial expressions as a signal for gesture generation and achieves real-time fusion of both facial expressions and gestures.

- *GestureDiffuCLIP* [?] uses a Diffusion model conditioned on text, leveraging Contrastive Learning with CLIP to integrate text features and control gesture styles. Similar to other prompt-based approaches such as StableDiffusion or Midjourney, it treats text as prompts to learn gestures from descriptive sentences.

- *Freetalker* [?] trains a Diffusion model on multiple datasets to generate speaker-specific gestures conditioned on speech and text. Unlike Transformer-based methods, Freetalker employs an Attention-based Network to model the correlation between textual, auditory, and gesture features during *Stage 5. Feature Fusion* (Figure 1.2).

1.4.2.2 Inherited Method in This Thesis

- *MDM* [?] applies conditional Diffusion to gesture generation, using CLIP (Contrastive Language–Image Pre-training) embeddings of descriptive text as conditions. MDM adopts a Transformer-based architecture to reconstruct original gesture data. Like other text-based Diffusion approaches, in

Stage 3. Feature Extraction (Figure 1.2), the input text is randomly masked to hide certain segments, enabling the model to learn the importance of each part for various gestures.

In *Stage 5. Feature Fusion* (Figure 1.2), the text is prepended as a `CLS` token to the gesture sequence before passing through the Transformer Encoder, where self-attention models the relationship between the text and each gesture frame. MDM predicts the original gesture data rather than noise.

- **DiffuseStyleGesture** [?] extends *MDM* [?] by incorporating audio, initial gestures, and style as conditioning inputs. In *Stage 1. Preprocessing* (Figure 1.2), the model processes coordinate vectors to obtain feature vectors of dimension $D = 1141$ per frame. In *Stage 2. Feature Processing* (Figure 1.2), DiffuseStyleGesture uses WavLM for audio embedding. In *Stage 5. Feature Fusion* (Figure 1.2), it improves upon MDM by applying Cross-Local Attention prior to the Transformer Encoder.

1.4.3 Suitability of Diffusion Models for Gesture Generation

Given that gesture data consist of joint coordinates and rotational angles, they require high detail to ensure realistic character motion. However, data in extreme parameter cases are sparse. Thanks to their ability to capture fine-grained details and generalize to rare cases, Diffusion models are well-suited to address these challenges, as discussed in **??**. A comparative evaluation of Diffusion models and other approaches is shown in Table 1.1.

This thesis selects Diffusion as the core model for the gesture generation task. Specifically, it adopts the **DiffuseStyleGesture** [?] model and integrates text as a semantic feature during *Stage 5. Feature Fusion* (Figure 1.2) in the gesture generation pipeline to propose the novel model **OHGesture**.

DANH MỤC CÔNG TRÌNH CỦA HVCH

1. Towards a GENEA Leaderboard – an Extended, Living Benchmark for Evaluating and Advancing Conversational Motion Synthesis [**?**].