

ĐẠI HỌC QUỐC GIA TP. HCM  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

HOÀNG MINH THANH

OPENHUMAN: HỆ THỐNG TỔNG HỢP  
CỦ CHỈ HỘI THOẠI DỰA TRÊN CẢM  
XÚC VÀ NGỮ NGHĨA

LUẬN VĂN THẠC SĨ

TP. Hồ Chí Minh – Năm 2024

ĐẠI HỌC QUỐC GIA TP. HCM  
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

HOÀNG MINH THANH

**OPENHUMAN: HỆ THỐNG TỔNG HỢP CỦ CHỈ  
HỘI THOẠI DỰA TRÊN CẢM XÚC VÀ NGỮ NGHĨA**

Ngành: Khoa học máy tính

Mã số Ngành: 8480101

NGƯỜI HƯỚNG DẪN KHOA HỌC  
HDC: PSG. TS. LÝ QUỐC NGỌC

TP. Hồ Chí Minh – Năm 2024

## **LỜI CAM ĐOAN**

Tôi cam đoan luận văn thạc sĩ ngành Khoa học máy tính, với đề tài OpenHuman: Hệ thống tổng hợp cử chỉ hội thoại dựa trên cảm xúc và ngữ nghĩa là công trình khoa học do tôi thực hiện dưới sự hướng dẫn của TS. Lý Quốc Ngọc.

Những kết quả nghiên cứu của luận văn hoàn toàn trung thực và chính xác.

*TP. Hồ Chí Minh, ngày... tháng... năm...*

**HỌC VIÊN THỰC HIỆN**

*(Ký và ghi rõ họ tên)*

## **LỜI CẢM ƠN**

Chúng tôi xin chân thành cảm ơn thầy Lý Quốc Ngọc đã tận tình hướng dẫn, truyền đạt kiến thức và kinh nghiệm, và đưa ra các giải pháp cho chúng tôi trong suốt quá trình thực hiện đề tài luận văn tốt nghiệp này.

Xin gửi lời cảm ơn đến quý thầy cô Khoa Công Nghệ Thông Tin trường Đại Học Khoa Học Tự Nhiên - Đại Học Quốc Gia Thành Phố Hồ Chí Minh, những người đã truyền đạt kiến thức quý báu cho chúng tôi trong thời gian học tập vừa qua.

Đồng thời cảm ơn các nhà khoa học đã nghiên cứu về đề tài mà chúng tôi đã trích dẫn để có thể có những kiến thức hoàn thiện luận văn của chúng tôi.

Sau cùng chúng tôi xin gửi lời cảm ơn đến gia đình, bạn bè... những người luôn động viên, giúp đỡ chúng tôi trong quá trình làm luận văn.

Một lần nữa, xin chân thành cảm ơn !

# MỤC LỤC

Lời cam đoan . . . . .	i
Lời cảm ơn . . . . .	ii
Danh mục hình ảnh . . . . .	vi
Danh mục các bảng số liệu . . . . .	vii
Danh mục các ký hiệu, các chữ viết tắt . . . . .	viii
Bảng chú thích thuật ngữ . . . . .	ix
Trang thông tin luận văn tiếng Việt . . . . .	x
Trang thông tin luận văn tiếng Anh . . . . .	xii
<b>Chương 1 MỞ ĐẦU</b>	<b>1</b>
1.1 Bối cảnh chung . . . . .	1
1.2 Động lực nghiên cứu . . . . .	2
1.3 Phát biểu bài toán . . . . .	3
1.4 Các khó khăn cần giải quyết . . . . .	5
1.5 Đóng góp dự kiến . . . . .	5
<b>Chương 2 TỔNG QUAN</b>	<b>7</b>
2.1 Mối quan hệ giữa cử chỉ và lời nói . . . . .	7
2.2 Tổng quan các phương pháp cho bài toán sinh cử chỉ . . . . .	8
2.2.1 Phương pháp dựa trên luật . . . . .	8
2.2.2 Phương pháp dựa trên thống kê . . . . .	8
2.2.3 Phương pháp học sâu . . . . .	9
<b>Chương 3 PHƯƠNG PHÁP NGHIÊN CỨU</b>	<b>12</b>
3.1 Mô hình diffusion cơ bản . . . . .	12
3.1.1 Quá trình gây nhiễu (forward diffusion process) . . . . .	13
3.1.2 Quá trình khử nhiễu (denoising process) . . . . .	15
3.1.3 Quá trình huấn luyện trong mô hình diffusion cơ bản . . . . .	16
3.1.4 Quá trình lấy mẫu trong diffusion cơ bản . . . . .	17

3.2 Mô hình của đề xuất OHGesture . . . . .	20
3.2.1 Quá trình trích xuất đặc trưng . . . . .	21
3.2.2 Áp dụng cơ chế Attention cho bài toán sinh cử chỉ . . . . .	23
3.2.3 Điều khiển cảm xúc trong bài toán sinh cử chỉ . . . . .	25
3.2.4 Quá trình huấn luyện . . . . .	26
3.2.5 Quá trình lấy mẫu . . . . .	27
<b>Chương 4 THỰC NGHIỆM</b>	<b>29</b>
4.1 Tập dữ liệu . . . . .	29
4.2 Quá trình xử lý dữ liệu . . . . .	29
4.3 Quá trình huấn luyện . . . . .	30
4.4 Quá trình sử dụng Unity để kết xuất . . . . .	30
<b>Chương 5 KẾT QUẢ VÀ ĐÁNH GIÁ</b>	<b>31</b>
5.1 Phương pháp đánh giá . . . . .	31
5.1.1 Mean Opinion Scores (MOS) . . . . .	31
5.1.2 Fréchet Inception Distance (FID) . . . . .	31
5.2 So sánh với các phương pháp hiện tại . . . . .	32
5.3 Xây dựng và tiêu chuẩn hóa hệ thống đánh giá kết quả sinh cử chỉ . . . . .	33
<b>Chương 6 KẾT LUẬN</b>	<b>35</b>
6.1 Kết quả đạt được . . . . .	35
6.2 Ưu và nhược điểm của mô hình . . . . .	35
6.3 Phương hướng phát triển và nghiên cứu trong tương lai . . . . .	36
6.4 Đóng góp của luận văn . . . . .	37
6.5 Lời kết . . . . .	39
<b>TÀI LIỆU THAM KHẢO</b>	<b>40</b>
<b>Chương A Các tham số trong Diffusion</b>	<b>46</b>
A.1 Sự thay đổi của $\sqrt{\alpha}$ và $\sqrt{1 - \alpha}$ . . . . .	46
A.2 Sự thay đổi của $\sqrt{\bar{\alpha}}$ và $\sqrt{1 - \bar{\alpha}}$ . . . . .	47
A.3 So sánh $\epsilon$ objective và $x_0$ . . . . .	48
<b>Chương B Quá trình xử lý dữ liệu BVH</b>	<b>49</b>
B.1 Cấu trúc khung xương của một nhân vật . . . . .	49

B.2	Cấu trúc tệp dữ liệu BVH . . . . .	50
B.3	Quá trình chuyển góc quay Euler sang Quaternion . . . . .	51
<b>Chương C</b>	<b>Minh họa kết quả suy luận của cử chỉ</b>	<b>53</b>
C.1	So sánh giữa cử chỉ groundtruth và cử chỉ dự đoán . . . . .	53
C.2	Minh họa việc sinh cử chỉ với âm thanh nằm ngoài tập huấn luyện . . .	54
C.3	Minh họa chuyển động của nhân vật . . . . .	54

## DANH MỤC HÌNH ẢNH

Hình 1:	Minh họa kỹ thuật đồ họa máy tính trong việc xây dựng người kỹ thuật số . . . . .	1
Hình 3:	Minh họa một chuỗi cử chỉ, ta lấy $N$ frame đầu làm cử chỉ khởi tạo s (seed gesture) và $M$ khung hình còn lại làm cử chỉ để học .	4
Hình 4:	Tổng quan về các mô hình tạo sinh khác nhau. . . . .	9
Hình 5:	Minh họa quá trình gây nhiễu (Diffuse) và khử nhiễu (Denoise) cử chỉ chưa qua quá trình huấn luyện . . . . .	12
Hình 7:	Minh họa quá trình khử nhiễu (Denoise) . . . . .	15
Hình 8:	Mô hình diffusion cơ bản . . . . .	16
Hình 9:	Quá trình Training và Sampling trong mô hình Diffusion tiêu chuẩn . . . . .	18
Hình 10:	Quá trình Huấn luyện và lấy mẫu trong OHGesture . . . . .	20
Hình 11:	Mô hình trong OHGesture . . . . .	22
Hình 12:	Cơ chế Self-Attention trong Transformer Encoder và Cross- Local Attention . . . . .	25
Hình 13:	Minh họa về 6 cử chỉ Happy, Sad, Neutral, Old, Relaxed và Angry . . . . .	29
Hình 14:	Sự thay đổi của $\sqrt{\alpha}$ và $\sqrt{1 - \alpha}$ . . . . .	46
Hình 15:	Quá trình thay đổi của $\sqrt{\bar{\alpha}}$ và $\sqrt{1 - \bar{\alpha}}$ . . . . .	47
Hình 16:	<b>So sánh <math>\epsilon</math> objective (bên trên) và <math>x_0</math> (bên dưới).</b> . . . . .	48
Hình 1:	Khung xương của một nhân vật . . . . .	49
Hình 16:	Quá trình xử lý tệp tin BVH . . . . .	50

## DANH MỤC CÁC BẢNG SỐ LIỆU

## DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

Từ viết tắt	Nội dung đầy đủ
$\theta$	Trọng số huấn luyện cần để học
$\theta'$	Trọng số huấn luyện sau khi học xong dùng để lấy mẫu
$\epsilon$	Nhiều được thêm vào ảnh
$\hat{\epsilon}$	Nhiều dự đoán
$\epsilon_\theta$ hay $f_\theta$	Hàm dự đoán nhiễu DDPM với trọng số $\theta$ , $f_\theta$ và $\epsilon_\theta$ là một.
$N, M, D$	Các số thể hiện số chiều của ma trận hoặc vector
$t : 1 \rightarrow T$	Bước gây nhiễu $t$ từ bước $t = 1$ đến $t = T$
$n_{\text{joins}}$	Số khung xương
$\mathcal{N}(a\mathbf{x}, b^2)$	Một hàm $f(x) = a\mathbf{x} + b\epsilon$ với $\epsilon \in \mathcal{N}(0, 1)$ thì tương đương phân phối chuẩn $\mathcal{N}(a\mathbf{x}, b^2)$
$\mathcal{N}(0, \mathbf{I})$	Phân phối chuẩn với mean là 0 và phương sai là 1
$\mathcal{N}(\mathbf{x}_t; \mu_\theta, \Sigma_\theta)$	Phân phối chuẩn với $\mathbf{x}_t$ là output, $\mu$ trong bình của phân phối chuẩn đó với tham số $\theta$ , hàm dựa trên tham số $\theta$ có phương sai $\Sigma$
$\mathbf{x}_t^{1:M}$	Dữ liệu cử chỉ ở bước thứ $t$ , bắt đầu từ khung hình thứ 1 đến khung hình thứ $M$
$\hat{\mathbf{x}}_0$	Dữ liệu dự đoán của mô hình
$f(x y)$	Hàm xác xuất có điều kiện với $y$ có trước tìm xác xuất để có $x$
$q(\mathbf{x}_t \mathbf{x}_{t-1})$	Hàm xác xuất có điều kiện của hàm gây nhiễu, khi biết trước $\mathbf{x}_t$ và tìm $\mathbf{x}_{t-1}$
$p_\theta(\mathbf{x}_{t-1} \mathbf{x}_t)$	Hàm xác xuất của quá trình khử nhiễu khi biết trước $\mathbf{x}_t$ và tìm $\mathbf{x}_{t-1}$ . Quá trình học để cập trọng số $\theta$
$\mu_\theta(\mathbf{x}_t, t)$	Giá trung bình của $\mathbf{x}_t$ ở bước thứ $t$ sau khi đi qua mô hình với trọng số $\theta$
$\mathbb{E}$	Kỳ vọng
$\mathcal{L}_t$	Hàm mất mát ở bước thứ $t$
$G_\theta$	Mô hình OHGesture cần huấn luyện với trọng số $\theta$
$\prod_{t=1}^T$	Hàm tích số học, $\prod_{t=1}^T a_t = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_T$

## BẢNG CHÚ THÍCH THUẬT NGỮ

Từ viết tắt	Nội dung đầy đủ
DDPM	Denoising Diffusion Probabilistic Models
OHGesture	Mô hình đề xuất của luận văn
Sampling/Inference	Quá trình lấy mẫu hoặc suy luận của mô hình
KL	Kullback–Leibler Divergence
MAE	Mean Absolute Error
MSE	Mean Squared Error
Forward Diffusion Process	Quá trình gây nhiễu không có huấn luyện bằng trọng số

# **TRANG THÔNG TIN LUẬN VĂN**

Tên đề tài luận văn: OpenHuman: Hệ thống tổng hợp cử chỉ hội thoại dựa trên cảm xúc và ngữ nghĩa

Ngành: Khoa học máy tính

Mã số ngành: 8480101

Họ tên học viên cao học: Hoàng Minh Thanh

Khóa đào tạo: 31

Người hướng dẫn khoa học: PGS. TS. Lý Quốc Ngọc

Cơ sở đào tạo: Trường Đại học Khoa học Tự nhiên, ĐHQG.HCM

## **1. TÓM TẮT NỘI DUNG LUẬN VĂN:**

Cùng với sự bùng nổ của phần cứng, các mô hình ngôn ngữ lớn, sự bùng nổ của các hệ thống trí tuệ nhân tạo dựa trên văn bản như ChatGPT, CharacterAI, Gemini,.. và sự phát triển của đồ họa máy tính thì nút nghẽn cổ chai hiện nay để phát triển người kỹ thuật số (digital human) chính là khả năng tạo ra chuyển động của nhân vật tương ứng với văn bản hoặc âm thanh. Một trong những khó khăn trong quá trình sinh cử chỉ là dữ liệu cử chỉ không đủ nhiều và chất lượng, cũng như sự thiếu thông nhất về ngữ cảnh trong cử chỉ là một trong những khó khăn trong việc xây dựng các hệ thống. Trong các phương pháp hiện nay, mô hình diffusion đạt kết quả tốt nhất do có khả năng tổng quát hóa và phủ được ở những vùng thiếu cân xứng giữa các đặc trưng, và các vùng có mật độ dữ liệu thấp.

Để có thể sinh cử chỉ tương ứng với âm thanh, và thông tin về cảm xúc cần học, chúng tôi sử dụng mô hình diffusion có điều kiện. Với điều kiện ở đây chính là cử chỉ khởi tạo, cảm xúc, âm thanh và văn bản tương ứng. Chúng tôi kế thừa từ mô hình DiffuseStyleGesture để xây dựng mô hình đề xuất **OHGesture**, chúng tôi chuyển âm thanh thành văn bản, như một đặc trưng về ngữ nghĩa bổ sung trong quá trình học. Chúng tôi kế thừa mã nguồn Unity của mô hình DeepPhase để kết xuất, và trực quan hóa các chuyển động của nhân vật, cuối cùng các mã nguồn và mã nguồn chương trình được chúng tôi công khai để cộng đồng nghiên cứu về sinh cử chỉ tiếp tục phát triển các hệ thống sinh cử chỉ tối ưu hơn trong tương lai.

## **2. NHỮNG KẾT QUẢ MỚI CỦA LUẬN VĂN:**

Thứ nhất, qua kết quả sinh cử chỉ thực tế, chúng tôi có thể chứng minh được mô hình đề xuất OHGesture đạt kết quả sinh cử chỉ tốt và thể hiện sự độ đồng bộ giữa cử

chỉ, âm thanh, và cảm xúc. Thứ hai, bằng việc tích hợp văn bản để bổ sung đặc trưng ngữ nghĩa, chúng tôi cung cấp thêm một hình học giúp mô hình có thể hiểu được từng ngữ cảnh cụ thể trong quá trình sinh cử chỉ. Thứ ba, nghiên cứu đóng góp một phương pháp mới trong việc xây dựng hệ thống đánh giá chuẩn hóa cho các mô hình sinh cử chỉ. Phương pháp kết hợp dữ liệu từ nhiều nguồn ngôn ngữ và sử dụng đánh giá của con người, tạo nên một nền tảng so sánh khách quan và toàn diện. Cuối cùng, chúng tôi sử dụng cộng minh hoạ chuyển động của nhân vật, mã nguồn trên github, mô hình pretrain trên Huggingface. Cũng như kết quả sinh cử mô hình đạt kết quả tốt với đầu vào nằm ngoài dữ liệu của quá trình huấn luyện.

### **3. CÁC ỨNG DỤNG/ KHẢ NĂNG ỨNG DỤNG TRONG THỰC TIỄN HAY NHỮNG VẤN ĐỀ CÒN BỎ NGỎ CẦN TIẾP TỤC NGHIÊN CỨU:**

Mô hình OHGesture đã có thể sinh cử chỉ từ nhãn cảm xúc, âm thanh và văn bản tương ứng. Từ đây, ta có thể phát triển các hệ thống tương tác với người bằng cách kết hợp với các agent chuyên xử lý thông tin bằng văn bản khác như Character.AI, ChatGPT API,.. Ngoài ra, chúng tôi cũng dự định làm một store tương tự như App Store nhưng chứa người ảo, với mỗi người ảo như một app được xây dựng từ nhiều người khác nhau có thể ứng dụng từ bạn gái ảo, giáo dục, trợ lý khách hàng...

Mô hình hiện tại đang sử dụng các chuỗi cử chỉ như một bức ảnh để khử nhiễu, nên kết quả vẫn chưa tốt, chúng tôi dự định sẽ cải tiến bằng các sử dụng Fast Fourier Transform để trích xuất các đặc trưng về pha của chuyển động, từ đó xây dựng hệ thống sinh cử chỉ tốt hơn.

### **TẬP THỂ CÁN BỘ HƯỚNG DẪN HỌC VIÊN CAO HỌC**

(Ký tên, họ tên)

(Ký tên, họ tên)

### **XÁC NHẬN CỦA CƠ SỞ ĐÀO TẠO HIỆU TRƯỞNG**

## THESIS INFORMATION

Thesis title: OpenHuman: A conversational gesture synthesis system based on emotions and semantics

Speciality: Computer Science

Code: 8480104

Name of Master Student: Hoàng Minh Thanh

Academic year: 31

Supervisor: Associate Professor Ly Quoc Ngoc

At: VNUHCM - University of Science

### 1. SUMMARY:

Along with the explosion of hardware advancements, large language models, text-based AI systems such as ChatGPT, CharacterAI, Gemini, and the development of computer graphics, the current bottleneck in creating digital humans lies in generating character movements corresponding to text or audio inputs.

One major challenge in gesture generation is the lack of sufficient and high-quality gesture data, as well as the lack of contextual consistency in gestures, making system development more difficult. Among current approaches, diffusion models achieve the best results due to their ability to generalize and handle imbalanced feature distributions and low-data-density regions effectively.

To generate gestures corresponding to audio and emotional information, we employ a conditional diffusion model. The conditions include initial gestures, emotions, audio, and corresponding text. We extend the DiffuseStyleGesture model to propose the **OHGesture** model. Our method converts audio into text as an additional semantic feature for training. We leverage the Unity codebase of the DeepPhase model for rendering and visualizing character movements. Lastly, we make all our source code publicly available to encourage further development of gesture generation systems by the research community.

### 2. NOVELTY OF THESIS:

First, through actual gesture generation results, we demonstrate that the proposed OHGesture model achieves high-quality gesture generation with synchronization between gestures, audio, and emotions.

Second, by integrating text to supplement semantic features, we provide an additional mechanism to help the model understand specific contexts during gesture

generation.

Third, this research contributes a novel method for constructing standardized evaluation systems for gesture generation models. By combining data from multiple linguistic sources and utilizing human evaluations, we establish an objective and comprehensive benchmarking platform.

Finally, we utilize animated character visualization, source code hosted on GitHub, and pretrained models on Huggingface. The proposed model achieves favorable results even with inputs outside the training dataset.

### **3. APPLICATIONS/ APPLICABILITY/ PERSPECTIVE:**

The OHGesture model is capable of generating gestures from emotion labels, audio, and corresponding text. This enables the development of human-interactive systems by integrating it with other text-based processing agents like Character.AI or the ChatGPT API. Additionally, we envision creating a store similar to the App Store but dedicated to virtual humans. Each virtual human, acting as an app, could be developed by various contributors and serve diverse applications such as virtual girlfriends, education, and customer assistance.

Currently, the model treats gesture sequences as images for denoising, which limits its performance. We plan to improve it by employing Fast Fourier Transform (FFT) to extract phase features of movements, aiming to develop a better gesture generation system.

**SUPERVISOR                  Master STUDENT**

(Signature, full name) (Signature, full name)

**CERTIFICATION**

**CERTIFICATION UNIVERSITY OF SCIENCE**

**PRESIDENT**

# Chương 1. MỞ ĐẦU

## 1.1 Bối cảnh chung



(a) Công nghệ CGI với người kỹ thuật số siêu thật [17]



(b) Minh họa công trình tái tạo khuôn mặt tổng thống Obama [25]

Hình 1: Minh họa kỹ thuật đồ họa máy tính trong việc xây dựng người kỹ thuật số

Mỗi ngày, trên thế giới có hàng tỷ người nhìn vào màn hình RGB, kết quả hiển thị trên màn hình là đầu ra của mọi hệ thống phần mềm. Nên việc hiển thị từng pixel trên màn hình và cách để mô phỏng lại hình ảnh trên một cách chân thực nhất được các nhà khoa học về đồ họa máy tính (Computer Graphic) nghiên cứu từ những năm 1960s và đặc biệt là việc mô hình phỏng lại con người. Từ 2014, các họa sĩ 3D đã có thể tạo nên một nhân vật người siêu thật như Hình 1a trong khi các phần cứng máy tính còn chưa phát triển như hiện nay.

Ngày nay, công nghệ đồ họa máy tính đã hoàn toàn có thể mô phỏng nhiều vật giống đến mức siêu thực (realistic) các vật phức tạp như nước, đường xá, bánh mì,... và thậm chí là cả cơ thể và khuôn mặt con người với độ chi tiết đến từng lông tơ, nốt mụn và vân mắt. Vào năm 2015, bằng kỹ thuật quét 3 chiều ghi lại toàn bộ các góc của khuôn mặt, sự phản chiếu ánh sáng, trong công trình [25] các nhà khoa học đồ

họa máy tính đã có thể tái tạo toàn bộ khuôn mặt của tổng thống Obama trên máy tính với độ chính xác cao và gần như không thể phân biệt Hình 1b.

Trí tuệ nhân tạo thể hiện kết quả vượt bậc những năm gần đây không chỉ trong nghiên cứu mà còn trong ứng dụng thực tế, tiêu biểu như ứng dụng ChatGPT, Mid-Journey và sự phát triển cả theo chiều dọc và chiều ngang trong nhiều lĩnh vực ứng dụng trí tuệ nhân tạo. Mặc dù đồ họa máy tính đã có thể xây dựng khuôn mặt người siêu thật, việc sinh cử chỉ lại phụ thuộc vào việc chụp chuyển động (Motion Capture) từ các cảm biến và gặp khó khăn khi xây dựng hệ thống trí tuệ nhân tạo học từ dữ liệu.

Các hệ thống trí tuệ nhân tạo hiện nay đã có thể tạo văn bản và âm thanh tiêm cận như con người, nhưng một trong những trở ngại lớn nhất để xây dựng con người kỹ thuật số hiện nay chính là việc sinh cử chỉ. Chính vì vậy, mục tiêu của luận văn là xây dựng một hệ thống sinh cử chỉ hội thoại dựa trên cảm xúc và ngữ nghĩa với dữ liệu đầu vào gồm cả văn bản và giọng nói.

## 1.2 Động lực nghiên cứu

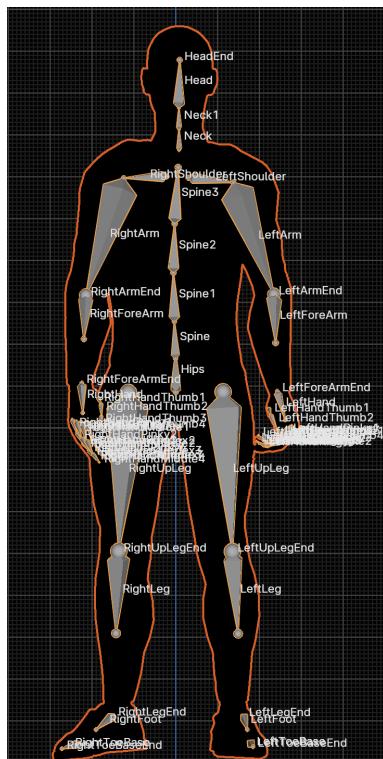
Tổng hợp cử chỉ hội thoại giúp ích cho rất nhiều lĩnh vực như hoạt ảnh, dựng phim, trò chơi điện tử, giáo dục và những ứng dụng thực tế ảo. Việc tổng hợp cử chỉ chuyển động được thực hiện theo cách truyền thống là thuê các diễn viên sử dụng thiết bị theo dõi chuyển động và bố trí các hệ thống cảm biến xung quanh thu nhận chuyển động để đạt được độ chính xác chân thực nhất. Tuy nhiên, các chuyển động thu được sau đó chỉ được phát lại và không có sự biến đổi linh hoạt giữa các hành động. Do đó, việc áp dụng trí tuệ nhân tạo để có thể học các chuyển động từ dữ liệu thu nhận và sau đó có thể sinh ra dữ liệu mới sẽ là một cuộc cách mạng trong ngành công nghiệp chụp chuyển động.

Vào năm 2011, một nhóm tác giả [3] đã chứng minh rằng có sự liên hệ giữa giọng nói và cử chỉ con người, đây chính là tiền đề cho thấy chúng ta có thể dùng dữ liệu âm thanh để có thể dùng để học và biểu diễn được cử chỉ con người. Với sự thành công của các mô hình ngôn ngữ tự nhiên trong xử lý văn bản và độ chính xác siêu thật trong việc mô phỏng gương mặt con người, ngành đồ họa máy tính đã đạt được những tiến bộ vượt bậc. Cùng với đó là sự dễ dàng và chính xác trong việc tổng hợp giọng nói con người. Việc ứng dụng trí tuệ nhân tạo để sinh cử chỉ con người là một trong những điểm nghẽn chính trong phát triển trợ lý ảo giao tiếp và tương tác với con

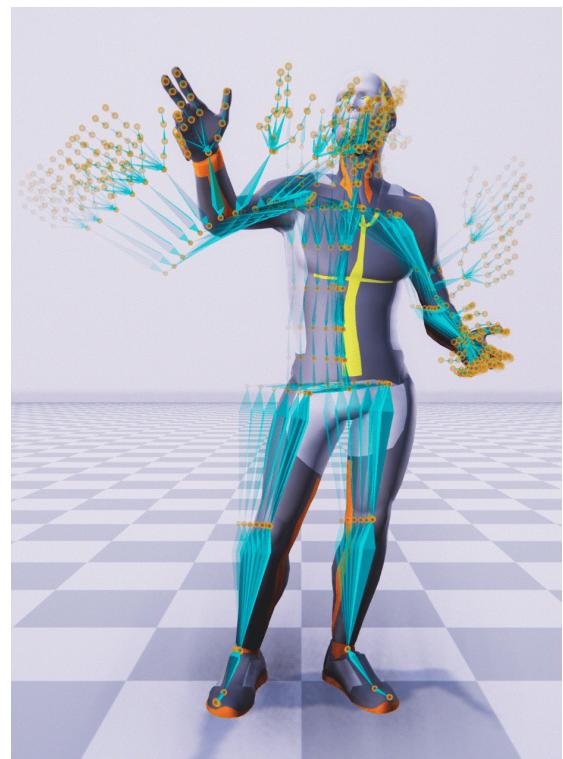
người.

### 1.3 Phát biểu bài toán

Mục tiêu của việc sinh cử chỉ (gesture generation) bằng phương pháp học máy là tạo ra các cử chỉ tự nhiên, giống con người (human-likeness) và phù hợp với ngữ cảnh. Sinh cử chỉ là bài toán hồi quy (regression), với đầu vào là một chuỗi cử chỉ cho trước và đầu ra là chuỗi cử chỉ tiếp theo.



(a) Khung xương và tên của các khớp của một khung xương trong mỗi khung hình.



(b) Chuỗi chuyển động của cử chỉ bao gồm 6 cử chỉ quá khứ và 6 cử chỉ tương lai.

Mỗi khung hình chuyển động của một nhân vật hay khung xương (skeleton) bao gồm dữ liệu về toạ độ vị trí và vận tốc theo thời gian. Ở đây dữ liệu của chúng tôi của một khung xương với mỗi khung hình (frame) bao gồm:

$$\mathbf{g} = \left[ \mathbf{p}_{\text{root}}, \mathbf{r}_{\text{root}}, \mathbf{p}'_{\text{root}}, \mathbf{r}'_{\text{root}}, \mathbf{p}_{\text{joins}}, \mathbf{r}_{\text{joins}}, \mathbf{p}'_{\text{joins}}, \mathbf{r}'_{\text{joins}}, \mathbf{d}_{\text{gaze}} \right] \quad (1)$$

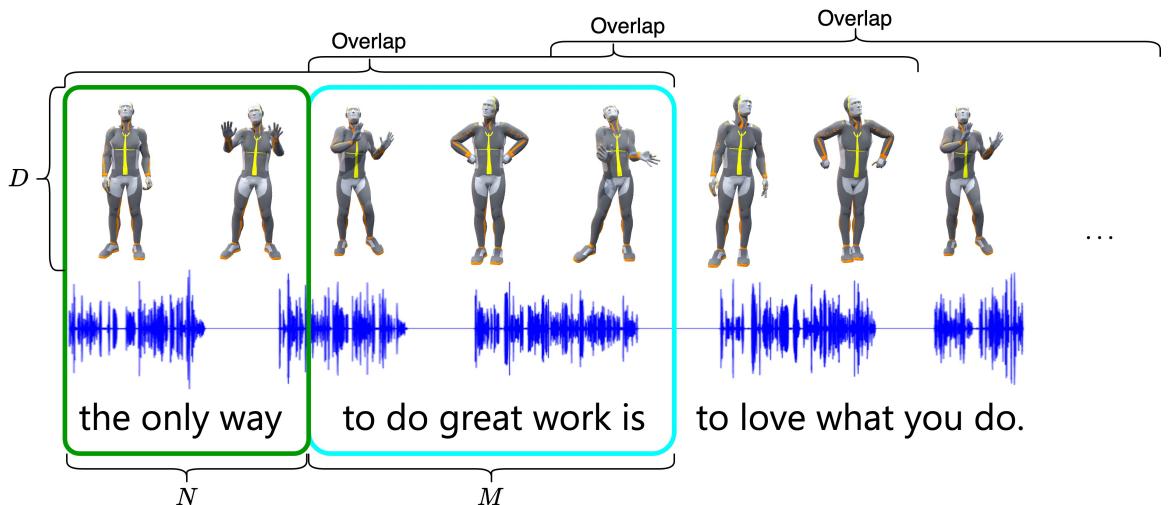
Trong đó với mỗi  $\mathbf{g} \in \mathbb{R}^{1141}$  bao gồm:

- $\mathbf{p}_{\text{root}} \in \mathbb{R}^3$ : Toạ độ của điểm gốc

- $\mathbf{r}_{\text{root}} \in \mathbb{R}^4$ : Góc quay của điểm gốc
- $\mathbf{p}'_{\text{root}} \in \mathbb{R}^3$ : Vận tốc thay đổi của tọa độ gốc
- $\mathbf{r}'_{\text{root}} \in \mathbb{R}^3$ : Vận tốc thay đổi của góc quay gốc
- $\mathbf{p}_{\text{joins}} \in \mathbb{R}^{3n_{\text{join}}}$ : Tọa độ của các khung xương
- $\mathbf{r}_{\text{joins}} \in \mathbb{R}^{6n_{\text{join}}}$ : Góc quay của các khung xương theo mặc phẳng X và Y
- $\mathbf{p}'_{\text{joins}} \in \mathbb{R}^{3n_{\text{join}}}$ : Vận tốc thay đổi của tọa độ các khung xương
- $\mathbf{r}'_{\text{joins}} \in \mathbb{R}^{3n_{\text{join}}}$ : Vận tốc thay đổi của góc quay các khung xương
- $\mathbf{d}_{\text{gaze}} \in \mathbb{R}^3$ : Là hướng nhìn

Tổng cộng có 75 khớp (joins) hay  $n_{\text{join}} = 75$ , với mỗi khung hình (frame) ta sẽ có một vector gồm 1141 chiều. Tập dữ liệu là tập nhiều chuỗi cử chỉ với độ dài tùy ý, từ mỗi cử chỉ độ dài tùy ý ta sẽ cắt thành các đoạn  $N + M$  khung hình,  $g \in \mathbb{R}^{(N+M) \times 1141}$ , trong đó cử chỉ  $s \in \mathbb{R}^{N \times 1141}$  đầu tiên là cử chỉ khởi tạo (seed gesture),  $x \in \mathbb{R}^{M \times 1141}$  cử chỉ tiếp theo cho việc dự đoán.

Dữ liệu âm thanh  $a_{\text{raw}} \in \mathbb{R}^{\text{length}}$  là chuỗi âm thanh thô được đọc ở sample rate 16000, sau đó được cắt thành  $a \in \mathbb{R}^{64000}$  tương ứng với 4 giây.



Hình 3: Minh họa một chuỗi cử chỉ, ta lấy  $N$  frame đầu làm cử chỉ khởi tạo  $s$  (seed gesture) và  $M$  khung hình còn lại làm cử chỉ để học

Quá trình xử lý dữ liệu được trình bày đầy đủ ở phụ lục B.

## 1.4 Các khó khăn cần giải quyết

Có rất nhiều khó khăn trong việc xây dựng một mô hình có thể học được các đặc trưng cử chỉ hội thoại như con người.

Thứ nhất, *dữ liệu không đủ nhiều và chất lượng*, chi phí để tạo ra một bộ dữ liệu trong ngành công nghiệp chụp chuyển động có chất lượng và quy mô lớn để ứng dụng thực tế là rất cao.

Thứ hai, *sự thiếu đồng nhất về ngữ cảnh của các loại dữ liệu*, các bộ dữ liệu về văn bản thường nhiều hơn so với giọng nói, và cũng không rõ văn bản đó được tạo ra bởi ai. Sự đồng bộ giữa giọng nói và cảm xúc khi nói cũng thường thiếu trong tập dữ liệu. Ngoài ra, dữ liệu văn bản trong tập dữ liệu huấn luyện lại thuộc nhiều chủ đề đa dạng.

Thứ ba, *sự phân bố không cân xứng về dữ liệu giữa các loại đăng trưng cần học*. Các dữ liệu dùng cho nghiên cứu cử chỉ hiện nay thường tập trung vào ngôn ngữ Tiếng Anh, các cử chỉ có sự phân bố không cân xứng giữa các trạng thái như nói, hỏi, hoặc im lặng.

Thứ tư, *chi phí tính toán với nhiều loại dữ liệu của mô hình là một thách thức lớn*. Với đầu vào của mô hình gồm nhiều loại dữ liệu như văn bản, tiếng nói và điểm 3D, cần nhiều lớp mã hóa cho từng loại dữ liệu, dẫn đến chi phí tính toán cao trong cả giai đoạn huấn luyện và suy luận. Nếu giảm thông tin dữ liệu đầu vào cũng sẽ giảm kết quả suy luận của mô hình khi sinh cử chỉ.

Cuối cùng, *các bước xử lý cần được thực hiện tuần tự*, cách hiệu quả nhất để con người tương tác với máy tính là thông qua giọng nói và nhập từ bàn phím, tuy nhiên việc xử lý được văn bản và giọng nói để làm đầu vào cho mô hình phải thực hiện tuần tự. Độ trễ trong quá trình suy luận của sản phẩm thực tế cũng là một vấn đề lớn, vì người dùng không thể chờ đợi quá lâu để nhận kết quả. Ngoài ra, việc hiển thị cử chỉ đó lên máy tính bằng kỹ thuật đồ họa cũng cần được tối ưu để giảm thời gian xử lý.

## 1.5 Đóng góp dự kiến

- Dựa trên tập dữ liệu có sẵn, chúng tôi chuyển âm thanh của dữ liệu thành văn bản, và dùng văn bản đó để làm dữ liệu huấn luyện mới.
- Dựa trên mô hình cơ bản DiffuseStyleGesture, chúng tôi mở rộng thêm đặc

trưng văn bản trong quá trình khử nhiễu có điều kiện.

- Chúng tôi sử dụng Unity để render, trích xuất dữ liệu và trực quan hóa kết quả sinh cử chỉ.
- Chúng tôi xây dựng hệ thống kết xuất, và minh họa chương trình bằng Unity

## **Chương 2. TỔNG QUAN**

Bài toán sinh cử chỉ cũng tương tự như các bài toán khác đều đã nghiên cứu và phát triển song hành với các phương pháp học máy truyền thống và hiện đại. Gồm các nhóm phương pháp dựa trên luật và các phương pháp dựa trên dữ liệu. Đầu tiên chúng tôi chứng minh mối quan hệ giữa cử chỉ và lời nói 2.1, từ đó là tiền đề thể hiện sự đồng bộ về dữ liệu giữa cử chỉ và lời nói. Ở phần 2.2 chúng tôi trình bày về các phương pháp đã được sử dụng trong quá trình sinh cử chỉ.

### **2.1 Mối quan hệ giữa cử chỉ và lời nói**

Cử chỉ được chia thành 6 nhóm chính theo ngôn ngữ học [10], [31] cử chỉ thích nghi (adaptors), cử chỉ biểu tượng (emblems), cử chỉ chỉ định (deictics), cử chỉ biểu trưng (iconics), cử chỉ ẩn dụ (metaphorics) và cử chỉ nhấn mạnh (beat).

Trong đó, cử chỉ nhấn mạnh không liên quan trực tiếp đến ngữ nghĩa lời nói [18] nhưng rất quan trọng để tạo sự hài hòa về nhịp điệu giữa lời nói và cử chỉ [31]. Tuy nhiên, lời nói và cử chỉ nhấn mạnh không đồng bộ hoàn toàn về mặt nhịp điệu [24], nên việc học mối liên hệ thời gian giữa chúng gặp khó khăn [4], [20], [41].

Cử chỉ liên quan đến các cấp độ khác nhau của thông tin lời nói [31]. Ví dụ, cử chỉ biểu tượng như cầm ngược ngón cái thường đi kèm với ngữ nghĩa cấp cao như tốt hay tuyệt vời, trong khi cử chỉ nhấn mạnh thường xuất hiện cùng với sự nhấn mạnh âm thanh cấp thấp. Nhiều nghiên cứu trước đây chỉ sử dụng các đặc trưng được trích xuất từ lớp cuối cùng của bộ mã hóa âm thanh để tổng hợp cử chỉ [1], [4], [21], [28], [42]. Tuy nhiên, cách thiết lập này có thể khuyến khích bộ mã hóa trộn lẫn thông tin lời nói ở nhiều cấp độ khác nhau vào cùng một đặc trưng.

Như được chỉ ra trong các tài liệu ngôn ngữ học [18] [27] [36], cử chỉ được sử dụng trong cuộc hội thoại hàng ngày có thể được chia thành một số lượng hạn chế các đơn vị ngữ nghĩa với các biến thể chuyển động khác nhau. Chúng tôi giả định rằng các đơn vị ngữ nghĩa này, thường được gọi là từ ngữ, liên quan đến các đặc trưng cấp

cao của âm thanh lời nói, trong khi các biến thể chuyển động được xác định bởi các đặc trưng âm thanh cấp thấp. Do đó, chúng tôi tách rời các đặc trưng cấp cao và cấp thấp từ các lớp khác nhau của bộ mã hóa âm thanh và học các ánh xạ giữa chúng và các từ ngữ cử chỉ và các biến thể chuyển động, tương ứng. Các thử nghiệm chứng minh cơ chế này thành công trong việc tách rời các đặc trưng ở nhiều cấp độ của cả lời nói và chuyển động và tổng hợp các cử chỉ phù hợp về mặt ngữ nghĩa và có phong cách.

## 2.2 Tổng quan các phương pháp cho bài toán sinh cử chỉ

### 2.2.1 Phương pháp dựa trên luật

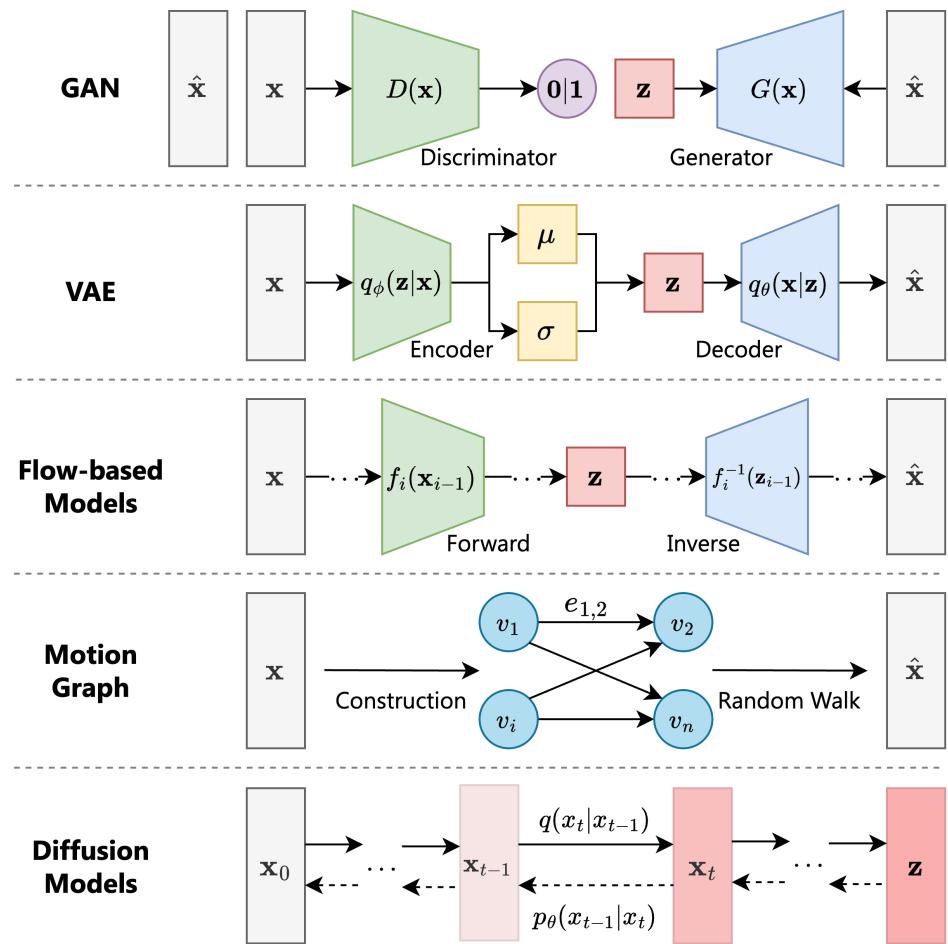
Các phương pháp dựa trên luật thường ánh xạ (mappings) từng âm thanh với từng đơn vị cử chỉ [16]. Và luật được tạo thủ công. Phương pháp dựa trên luật thì chúng ta có thể dễ dàng điều khiển kết quả của mô hình và có khả năng giải thích tốt kết quả dự đoán của mô hình. Tuy nhiên chi phí để tạo thủ công là không khả thi để xây dựng cho các ứng dụng phức tạp đòi hỏi phải xử lý một lượng dữ liệu rất lớn.

### 2.2.2 Phương pháp dựa trên thống kê

Tương tự như phương pháp dựa trên luật, phương pháp dựa trên dữ liệu cũng ánh xạ các đặc trưng của âm thanh tương ứng với cử chỉ nhưng thay vì làm thủ công thì được sử dụng học một cách tự động dựa trên dữ liệu. Trong đó có hai phương pháp chính là phương pháp thống kê và phương pháp dựa trên dữ liệu.

Phương pháp thống kê sử dụng phân phối xác suất để tìm sự tương đồng giữa các đặc trưng âm thanh và cử chỉ [22]. Tác giả [27] xây dựng mô hình để học từng phong cách của từng người nói.

### 2.2.3 Phương pháp học sâu



Hình 4: Tổng quan về các mô hình tạo sinh khác nhau.

Phương pháp sinh cử chỉ được chia thành hai nhóm chính. Bao gồm các mô hình ước lượng log likelihood (likelihood-based model) và phương pháp dựa vào các mô hình sinh ngầm định (implicit generative models) [32].

### 2.2.3 Likelihood-based Model

Phương pháp học ước lượng log likelihood là phương pháp học trực tiếp từ hàm mật độ xác suất (probability density) thông qua maximum likelihood. Các phương pháp điển hình là autoregressive models, normalizing flow models, energy-based models (EBMs), và variational auto-encoders (VAEs).

Mô hình được kết hợp với văn bản đầu vào được gắn thẻ với chủ đề, trọng tâm câu và thành ngữ để tạo ra các kịch bản cử chỉ, sau đó được ánh xạ sang một chuỗi các

củ chỉ được chọn từ một từ điển hoạt họa. [8] huấn luyện một model classifier neural network để chọn một đơn vị củ chỉ phù hợp dựa trên đầu vào là lời nói.

Củ chỉ có thể được tổng hợp bằng các mô hình xác định như perceptron đa tầng (MLP) [20], recurrent neural networks [4], [23], [13], [41], convolutional networks [12] và transformer [5]

### 2.2.3 Implicit Generative Models

Trong các phương pháp dựa vào mô hình sinh ngầm định, phân phối của dữ liệu được học một cách ngầm định thông qua việc quá trình lấy mẫu (sampling). Ví dụ tiêu biểu nhất là mô hình generative adversarial networks (GANs). Khi dữ liệu được tổng hợp bằng cách chuyển phân phối dữ liệu ban đầu ở dạng phân phối chuẩn về phân phối của dữ liệu.

Phương pháp tiêu biểu	Ưu điểm	Nhược điểm	Loại phương pháp
Robot behavior toolkit [16]	<ul style="list-style-type: none"> <li>- Dễ hiểu và dễ triển khai.</li> <li>- Dễ giải thích và có thể kiểm soát được</li> <li>- Hiệu quả trong các trường hợp đơn giản hoặc dữ liệu nhỏ.</li> </ul>	<ul style="list-style-type: none"> <li>- Không tổng quát hóa tốt với dữ liệu phức tạp.</li> <li>- Đòi hỏi nhiều công sức trong việc xây dựng quy tắc thủ công.</li> </ul>	Rule-Based
MLP [20], RNN [4], [23], [13], [41], CNN [12], Transformer [5]	<ul style="list-style-type: none"> <li>- Khả năng ước lượng mật độ xác suất của dữ liệu</li> <li>- Có khả năng mở rộng và học từ dữ liệu lớn</li> </ul>	<ul style="list-style-type: none"> <li>- Dễ bị ảnh hưởng bởi nhiễu</li> <li>- Kết quả thấp ở vùng dữ liệu hiếm</li> <li>- Khả năng sinh không đa dạng</li> </ul>	Likelihood-Based Models
<b>DiffusionStyle-Gesture</b> [40], MDM [34], Motiondiffuse [43]	<ul style="list-style-type: none"> <li>- Tạo ra dữ liệu chất lượng cao.</li> <li>- Linh hoạt và đa dạng</li> <li>- Phù được vùng có mật độ dữ liệu thấp</li> </ul>	<ul style="list-style-type: none"> <li>- Cần cấu hình phức tạp để đạt hiệu năng tốt.</li> <li>- Khó đánh giá do mỗi lần nhiễu là khác nhau.</li> <li>- Quá trình lấy mẫu chậm</li> </ul>	Implicit Generative Models

Bảng 2.1: Bảng so sánh ưu và nhược điểm của các phương pháp

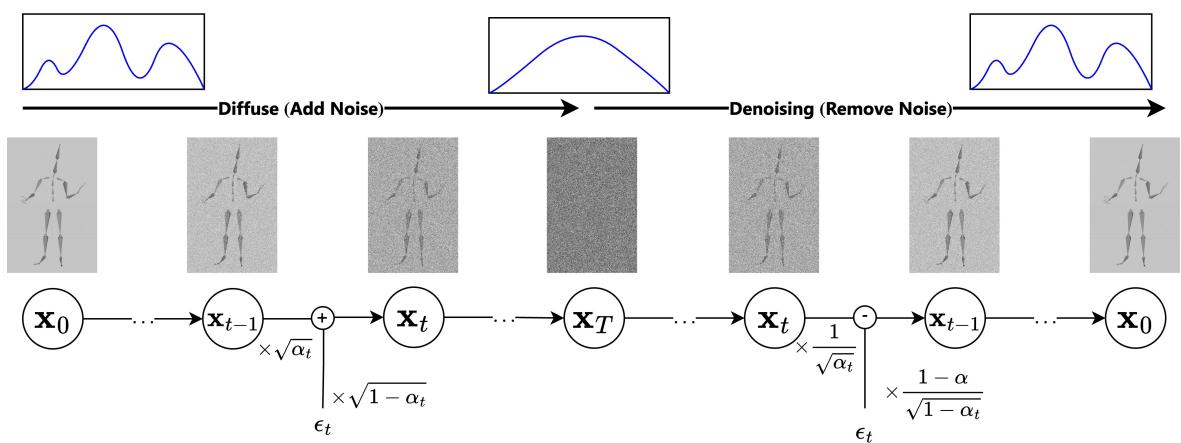
Với đặc điểm dữ liệu là giá trị của các góc quay, các toạ độ của điểm khớp, nên cần độ chi tiết cao để tạo ra sự chân thực trong các chuyển động của nhân vật. Ngoài ra dữ liệu sẽ thiếu và rất ít dữ liệu trong các trường hợp cực trị của tham số. Nên chúng tôi sử dụng mô hình Diffusion, với đặc điểm là có thể học được độ chi tiết cao hơn và có thể phủ được dữ liệu trong các trường hợp cực trị của tham số và độ phủ về mật độ

dữ liệu thấp.

## Chương 3. PHƯƠNG PHÁP NGHIÊN CỨU

Bản chất của các phương pháp dựa trên neuron network là ước lượng xác xuất của dữ liệu (probability density) nên cần chuẩn hoá (normalization), trong khi mô hình Diffusion sẽ học để có thể ước lượng đạo hàm của phân phối dữ liệu (estimating gradients of data distribution) và không phải chuẩn hoá trên toàn bộ dữ liệu [32], nên có kết quả tốt hơn so với các phương pháp neuron network không sử dụng diffusion. Mô hình diffusion có khả năng mô phỏng phân phối dữ liệu trong các trường hợp mật độ phân phối của dữ liệu thấp và có thể sinh được kết quả với độ chi tiết cao nên phù hợp với bài toán sinh cử chỉ. Mô hình của chúng tôi dựa trên mô hình DiffuseStyleGesture [40] với cải tiến để có thể học theo cảm xúc của nhân vật. Trước tiên chúng tôi trình bày về mô hình diffusion 3.1, và phần 3.2 sẽ trình bày về mô hình đề xuất OHGesture.

### 3.1 Mô hình diffusion cơ bản



Hình 5: Minh họa quá trình gây nhiễu (Diffuse) và khử nhiễu (Denoise) cử chỉ chưa qua quá trình huấn luyện

Với các phương pháp sử dụng neural network như ResNet, GAN,... mục tiêu là tìm được trọng số  $\theta$  của hàm  $f_\theta(x)$ , Hàm mà đầu vào là  $x$ , chúng ta sẽ cực tiểu hoá hàm lỗi  $\mathcal{L}_{\text{loss}}$  giữa nhãn  $y$  và kết quả dự đoán là  $\hat{y}$ . Sau khi kết thúc quá trình huấn luyện và học được trọng số  $\theta'$  ta dự đoán một mẫu dữ liệu mới  $x'$  bằng cách forward qua hàm  $f_{\theta'}$  để ra được kết quả dự đoán  $y'$ .

Tương tự phương pháp VAE, mã hoá (encode) ma trận đầu vào thành vector tiềm ẩn  $z$  và giải mã (decode) vector tiềm ẩn  $z$  ngược trở lại ma trận kích thước ban đầu, tuy nhiên mô hình diffusion chia quá trình học thành từng  $T$  bước, ở bước thứ  $t$  quá trình gây nhiễu (forward diffusion process)  $q(\mathbf{x})$  từ  $1 \rightarrow T$  được thực hiện bằng cách thêm nhiễu  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$  phân phối chuẩn Gaussian vào dữ liệu, với giá trị trung bình (mean) bằng  $\mathbb{E}[\epsilon] = 0$  và độ lệch chuẩn  $\text{Var}(\epsilon) = 1$ ,  $\mathbf{I}$  là phần tử đơn vị của phân phối chuẩn Gaussian. Trong hình minh họa 5 quá trình giảm nhiễu từ trái qua phải, còn quá trình khử nhiễu từ phải qua trái  $p_\theta(x)$ . Trong quá trình khử nhiễu (denoising process) từ  $T \rightarrow 1$ , mục tiêu là học được trọng số  $\theta$  của hàm dữ đoán nhiễu  $f_\theta$  hay còn được ký hiệu là hàm dự đoán lượng nhiễu ( $\epsilon_\theta$ ) đã được thêm vào. Sau khi kết thúc quá trình học và ta có trọng số  $\theta'$  của hàm dự đoán nhiễu  $\hat{\epsilon}$ , ta sẽ dùng hàm  $f_{\theta'}$  để dự đoán nhiễu. Sau khi có nhiều dự đoán  $\hat{\epsilon}$ , chúng ta trừ đi ảnh bị nhiễu  $\mathbf{x}_t$  để có được ảnh khử nhiễu  $\mathbf{x}_{t-1}$ , và cộng với nhiễu  $\mathbf{z} \in \mathcal{N}(0, \mathbf{I})$  để tạo ra sự đa dạng cử chỉ. Thực hiện lần lượt từ  $T \rightarrow 1$  để có được ảnh dự đoán  $\hat{x}_0$ . Như hình 8 là kiến trúc đầy đủ của mô hình diffusion tiêu chuẩn (Denoising Diffusion Probabilistic Models - DDPM).

### 3.1.1 Quá trình gây nhiễu (forward diffusion process)

Cho dữ liệu  $\mathbf{x}_0$  là được lấy từ dữ liệu thật  $\mathbf{x}_0 \sim q(x)$ , với mỗi bước ta sẽ thêm nhiễu vào đầu vào  $\mathbf{x}_0$  với tỷ lệ nhiễu và ảnh gốc được kiểm soát bằng hệ số  $\beta$ :

$$\mathbf{x}_t = \sqrt{1 - \beta_t} \mathbf{x}_{t-1} + \sqrt{\beta_t} \epsilon_{t-1} \quad (1)$$

Trong đó, quá trình gây nhiễu từ  $1 \rightarrow T$ , với mỗi bước  $t$  quá trình thêm nhiễu  $\epsilon$  được điều khiển bằng  $\beta_t$  theo phương sai  $\{\beta_t \in (0, 1)\}_{t=1}^T$ :

$$\begin{aligned} q(\mathbf{x}_t | \mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \\ q(\mathbf{x}_{1:T} | \mathbf{x}_0) &= \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}) \end{aligned} \quad (2)$$

Mục tiêu ở bước  $t$  của hệ số  $\sqrt{1 - \beta_t}$  và  $\beta_t$  là để lần lượt giảm tỷ lệ của ảnh gốc  $x_t$  và tăng dần nhiễu  $\epsilon_{t-1}$ , vì vậy  $\beta_1 < \beta_2 < \dots < \beta_T$ . Khi  $T \rightarrow \infty$  thì  $x_T$  sẽ hoàn toàn

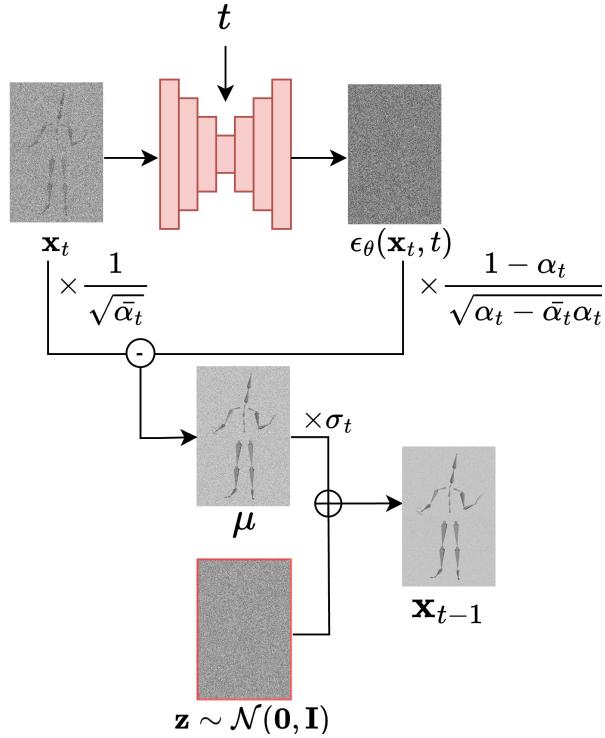
nhiều [37] (Isotropic Gaussian Distribution)  $q(T) = \mathcal{N}(0, \mathbf{I})$ .

Vì nhiều  $\epsilon_{t-1}, \epsilon_{t-2}, \dots \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  luôn luôn là phân phối chuẩn Gaussian, và cho trước trong mọi bước  $t$ , nên ta có thể dễ dàng truy ngược được  $\mathbf{x}_t$  từ  $\mathbf{x}_0$ . Bằng cách đặt  $\alpha_t = 1 - \beta_t$  và  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ , từ công thức 1, ta có hàm forward diffusion viết lại theo  $\alpha$  như sau:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t} \mathbf{x}_{t-1} + \sqrt{1 - \alpha_t} \epsilon_{t-1} \\
&= \sqrt{\alpha_t} \left( \sqrt{\alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}} \epsilon_{t-2} \right) + \sqrt{1 - \alpha_t} \epsilon_{t-1} \\
&= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{\alpha_t(1 - \alpha_{t-1})} \epsilon_{t-2} + \sqrt{1 - \alpha_t} \epsilon_{t-1} \\
&= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{\alpha_t(1 - \alpha_{t-1}) + (1 - \alpha_t)} \epsilon_{t-2} \\
&= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \epsilon_{t-2} \\
&= \dots \\
&= \sqrt{\prod_{i=1}^t \alpha_i} \mathbf{x}_0 + \sqrt{1 - \prod_{i=1}^t \alpha_i} \epsilon_0 \\
&= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_0 \\
&\sim \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})
\end{aligned} \tag{3}$$

Sự thay đổi của  $\sqrt{\alpha}$  và  $\sqrt{1 - \alpha}$  trong quá trình gây nhiễu được thể hiện ở phụ lục B

### 3.1.2 Quá trình khử nhiễu (denoising process)



Hình 7: Minh họa quá trình khử nhiễu (Denoise)

Quá trình khử nhiễu  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  ở bước thứ  $t$  từ  $T \rightarrow 1$  được bắt đầu từ  $\mathbf{x}_T$  là hoàn toàn nhiễu  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ . Ta sẽ dùng một neural network  $f_\theta(x_t, t)$  để dự đoán nhiễu  $\hat{\epsilon} = f_\theta(\mathbf{x}_t, t)$  đã được thêm vào để được  $\mathbf{x}_{t-1}$  từ  $\mathbf{x}_t$ .

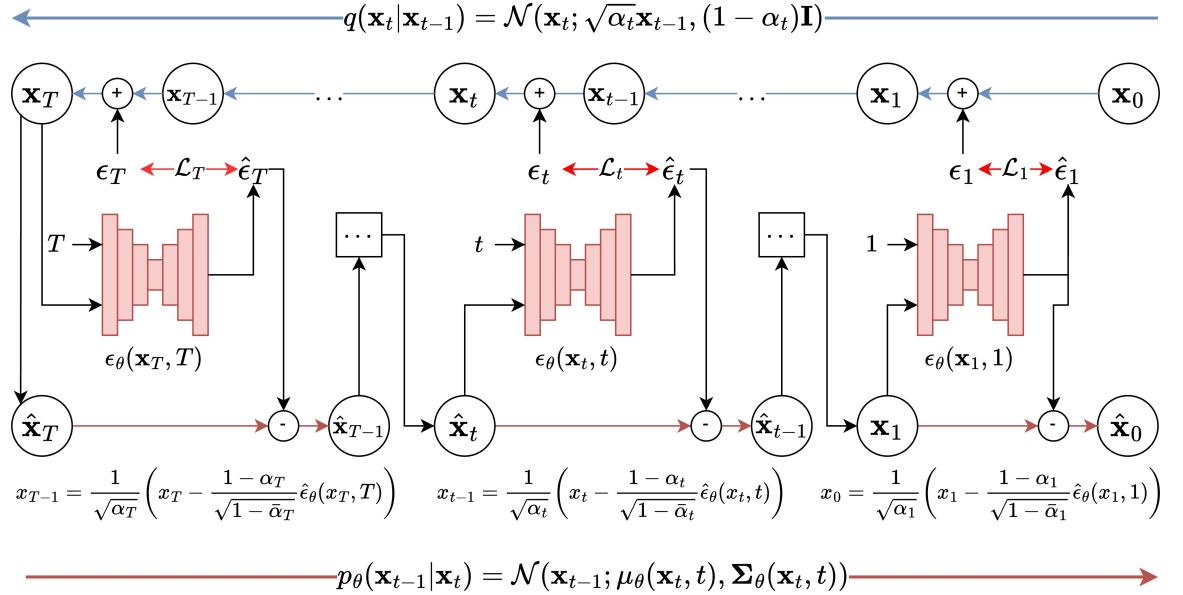
Quá trình khử nhiễu có trung bình  $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} f_\theta(\mathbf{x}_t, t) \right)$  và độ lệch chuẩn  $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$  như sau:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad (4)$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

Quá trình khử nhiễu là quá trình bắt đầu từ nhiễu hoàn toàn, dùng một neuron network  $f_\theta$  để học được quá trình khử nhiễu.

### 3.1.3 Quá trình huấn luyện trong mô hình diffusion cơ bản



Hình 8: Mô hình diffusion cơ bản

Mô hình diffusion sẽ học trọng số  $\theta$  của hàm dự đoán lỗi  $f_\theta(x_t, t)$  hay còn ký hiệu là  $\epsilon_\theta(x_t, t)$ . Trong quá trình denoising, ta sẽ tối ưu độ lỗi giữa nhiễu dự đoán  $\epsilon_\theta(x_t, t)$  và nhiễu thực tế  $\epsilon_t$ . Với mỗi bước thứ  $t$  ta sẽ tối ưu hàm loss  $\mathcal{L}_t$  để thu được trọng số  $\theta$ .

$$\begin{aligned} \mathcal{L}_t &= \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \epsilon_t} \left[ \|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\|^2 \right] \\ &= \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \epsilon_t} \left[ \|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t, t)\|^2 \right] \end{aligned} \quad (5)$$

Với  $f_\theta(x_{t-1}, t)$  hay  $\epsilon_\theta$  là một mô hình Unet dùng để mã hoá và giải mã dữ liệu để dự đoán nhiễu đã thêm vào dữ liệu. Quá trình tính toán hàm lỗi được trình bày minh họa trong hình 8.

---

**Algorithm 1** Thuật toán training trong DDPM

---

1. Tính sẵn các giá trị  $\sqrt{\alpha_t}$ ,  $\sqrt{1 - \alpha_t}$  và  $\sqrt{\bar{\alpha}_t}$  cho mỗi bước  $t : 1 \rightarrow T$ . Xác định lịch trình nhiễu  $\{\alpha_t \in (0, 1)\}_{t=1}^T$ , với  $\alpha_1 < \alpha_2 < \dots < \alpha_T$ .
2. Lấy nhãn  $\mathbf{x}_0$  từ phân phối của dữ liệu đã chuẩn hóa.
3. Tạo nhiễu ngẫu nhiên  $\epsilon_t$  cho mỗi bước  $t : 1 \rightarrow T$ , với  $\forall t : \epsilon_t \sim \mathcal{N}(0, \mathbf{I})$ .
4. Gây nhiễu (forward)  $\mathbf{x}_0$  để thu được  $\mathbf{x}_t$  ở mỗi bước  $t : 1 \rightarrow T$ :

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t$$

5. Với mỗi  $t$ , lấy  $t$  **ngẫu nhiên** từ  $[1, T]$ .
6. Cho  $\mathbf{x}_t$  và  $t$  vào mô hình để dự đoán nhiễu:  $\hat{\epsilon} = \epsilon_\theta(\mathbf{x}_t, t)$ .
7. Tính đạo hàm để cập nhật trọng số:

$$\nabla_{\theta_t} \|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\|^2$$

Tính loss:

$$\mathcal{L}_t = \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \epsilon_t} [\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t, t)\|^2]$$

8. Quay lại bước 6 cho đến khi hội tụ để thu được trọng số tối ưu  $\theta'$ .
- 

### 3.1.4 Quá trình lấy mẫu trong diffusion cơ bản

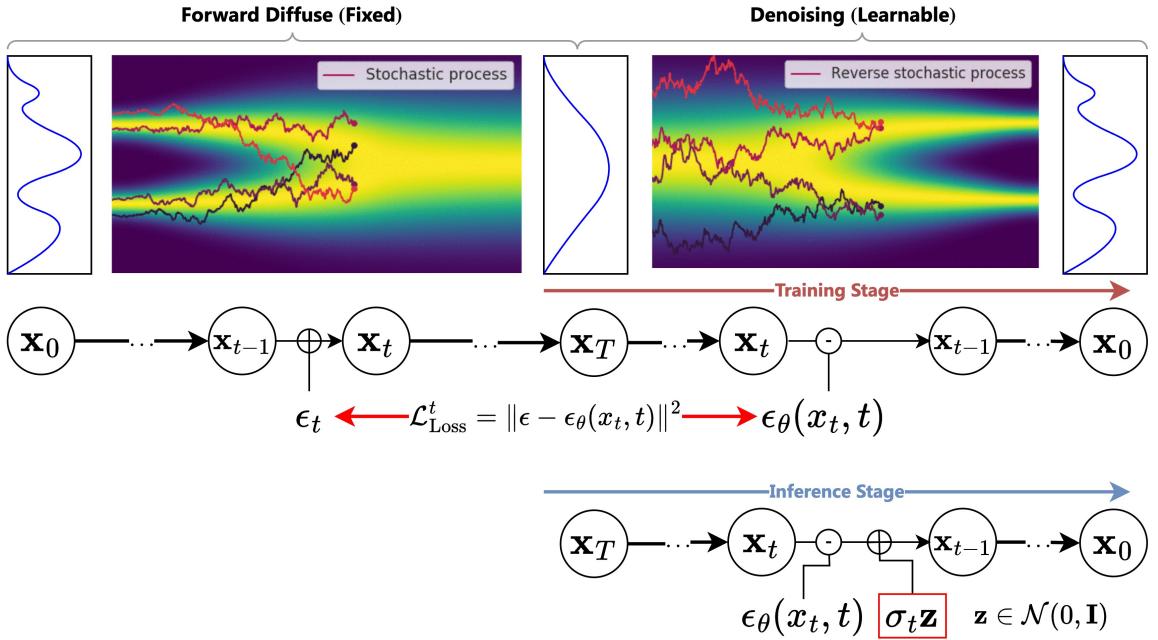
Sau khi thu được trọng số  $\theta'$ , ta sẽ dùng hàm denoising để khử nhiễu từ nhiễu  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ . Quá trình biến đổi từ nhiễu hoàn toàn  $\mathbf{x}_T$  sang dự đoán  $\hat{\mathbf{x}}_0$  như sau:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} f_{\theta'}(\mathbf{x}_t, t) \right) + \sqrt{1 - \alpha_t} \tilde{\epsilon}_t \quad (6)$$

Lưu ý rằng  $\epsilon_t$  là nhiễu cố định và lấy ngẫu nhiên trước quá trình huấn luyện, chỉ sử dụng lại kết quả ngẫu nhiên trong quá trình forward diffusion 3.1.2 ở công thức 1. Như hình 9, độ lỗi  $\epsilon_t$  là độ nhiễu của từng bước  $t$ , và hàm loss  $\mathcal{L}^t$  sẽ tính theo từng bước  $t$ .

Tiếp theo quá trình huấn luyện, thuật toán lấy mẫu (sampling) trong DDPM bắt đầu từ bước tạo nhiễu hoàn toàn, tức là  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ , nơi dữ liệu ban đầu hoàn toàn là nhiễu. Các giá trị  $\sqrt{\alpha_t}$ ,  $\sqrt{1 - \alpha_t}$  và  $\sqrt{\bar{\alpha}_t}$ , được tính từ quá trình huấn luyện, sẽ được sử dụng trong quá trình lấy mẫu để dự đoán lại dữ liệu gốc  $\mathbf{x}_0$ . Bước tiếp theo là tính

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_t, (1 - \alpha_t) \mathbf{I}) \rightarrow p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \rightarrow$$



Hình 9: Quá trình Training và Sampling trong mô hình Diffusion tiêu chuẩn

toán hệ số điều chỉnh nhiễu  $\sigma_t$ , dựa vào lịch trình nhiễu  $\alpha_t$  đã xác định trong quá trình huấn luyện. Các giá trị này sẽ ảnh hưởng đến độ nhiễu được thêm vào trong quá trình lấy mẫu ngược.

Thuật toán lấy mẫu sẽ được thực hiện từ bước  $T$  trở về 1, và trong mỗi bước, một nhiễu ngẫu nhiên  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$  được tạo ra để cộng thêm vào kết quả dự đoán. Tại mỗi bước  $t$ , mô hình sẽ dự đoán nhiễu  $\epsilon_\theta$  dựa trên dữ liệu nhiễu  $\mathbf{x}_t$  và bước thời gian  $t$ , sau đó sử dụng dự đoán này để tính toán giá trị  $\mu$ , là ước lượng của  $\mathbf{x}_0$ . Cuối cùng, một lượng nhiễu  $\sigma_t \mathbf{z}$  được cộng thêm vào  $\mu$  để thu được  $\hat{\mathbf{x}}_{t-1}$ , dữ liệu nhiễu tại bước  $t-1$ . Quá trình này tiếp tục cho đến khi  $t=1$ , và khi đó, chúng ta có được  $\hat{\mathbf{x}}_0$  — dự đoán cuối cùng của dữ liệu gốc từ quá trình khử nhiễu.

---

**Algorithm 2** Thuật toán sampling trong DDPM

---

1. Bắt đầu với nhiễu:  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ .
2. Các giá trị  $\sqrt{\alpha_t}$ ,  $\sqrt{1 - \alpha_t}$  và  $\sqrt{\bar{\alpha}_t}$  được lấy từ quá trình huấn luyện.
3. Tính hệ số điều chỉnh nhiễu  $\sigma_t$  từ  $\alpha_t$  ở mỗi bước  $t : 1 \rightarrow T$ :

$$\sigma_t = \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} (1 - \alpha_t)}$$

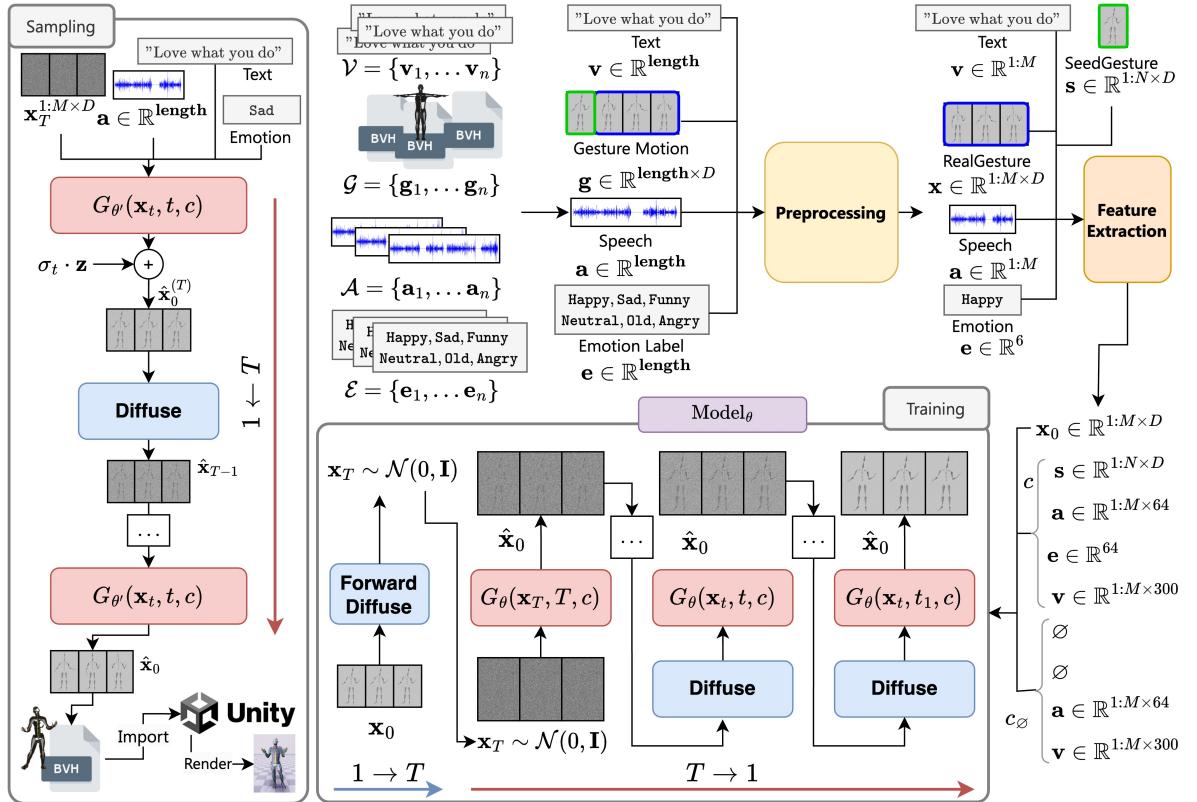
4. Với mỗi  $t$ , lấy  $t$  **tuần tự** từ  $[T, \dots, 1]$ .
5. Tạo nhiễu ngẫu nhiên  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ .
6. Đưa  $\mathbf{x}_t$  vào mô hình để suy luận nhiễu:  $\epsilon_{\theta'} = \epsilon_{\theta'}(\mathbf{x}_t, t)$ .
7. Dùng nhiễu dự đoán để trừ đi  $\mathbf{x}_t$  ở bước  $t$ :

$$\mu = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta'}(\mathbf{x}_t, t) \right)$$

8. Cộng thêm một lượng nhiễu:  $\hat{\mathbf{x}}_{t-1} = \mu + \sigma_t \mathbf{z}$ .
  9. Khi  $t = 1$ , thu được  $\hat{\mathbf{x}}_0$  từ quá trình khử nhiễu.
- 

Điều quan trọng nhất trong quá trình lấy mẫu (denoising sampling) đó chính là phải cộng thêm một độ nhiễu  $\mathbf{z} \in \mathcal{N}(0, \mathbf{I})$ , với  $\mathbf{z}$  sẽ được lần lượt thêm ở từng bước  $t$  bằng hệ số điều khiển  $\sigma_t$ . Mục tiêu của nhiễu  $\epsilon$  là để làm gờ phân phối (marginal noise distribution) cho mô hình  $f_\theta$  (hay  $\epsilon_\theta$ ) có thể học được nhiễu, còn với  $\mathbf{z}$  là để tăng tính đa dạng trong quá trình sinh và sự ổn định trong quá trình lấy mẫu. Thuật toán lấy mẫu được trình bày ở 2

## 3.2 Mô hình của đề xuất OHGesture



Hình 10: Quá trình Huấn luyện và lấy mẫu trong OHGesture

Mô hình đề xuất **OHGesture** của chúng tôi được dựa trên mô hình **DiffuseStyleGesture** [38] áp dụng mô hình Diffusion [14] có điều kiện [15] (Classifier-Free Diffusion Guidance) để điều khiển các đặc trưng trong quá trình khử nhiễu. Không giống như trong mô hình DiffuseStyleGesture, chúng tôi sử dụng thêm điều kiện văn bản được nhúng bằng mô hình Pretrain FastText [6].

Những điểm giống và khác của việc áp dụng mô hình Diffusion cho bài toán sinh cử chỉ so với mô hình Diffusion trong bài toán sinh ảnh:

### Điểm giống

- Chúng tôi sử dụng mô hình Diffusion [38] trên cử chỉ  $\mathbf{x}^{1:M \times D}$ , với  $M$  frame theo thời gian,  $D = 1141$  là các điểm toạ độ chuyển động của mỗi khung hình (tương tự width và height trong ảnh).
- Classifier-Free Diffusion Guidance với  $\mathbf{x}_0$  objective.

- Latent vector có số chiều là 256.

### Điểm khác

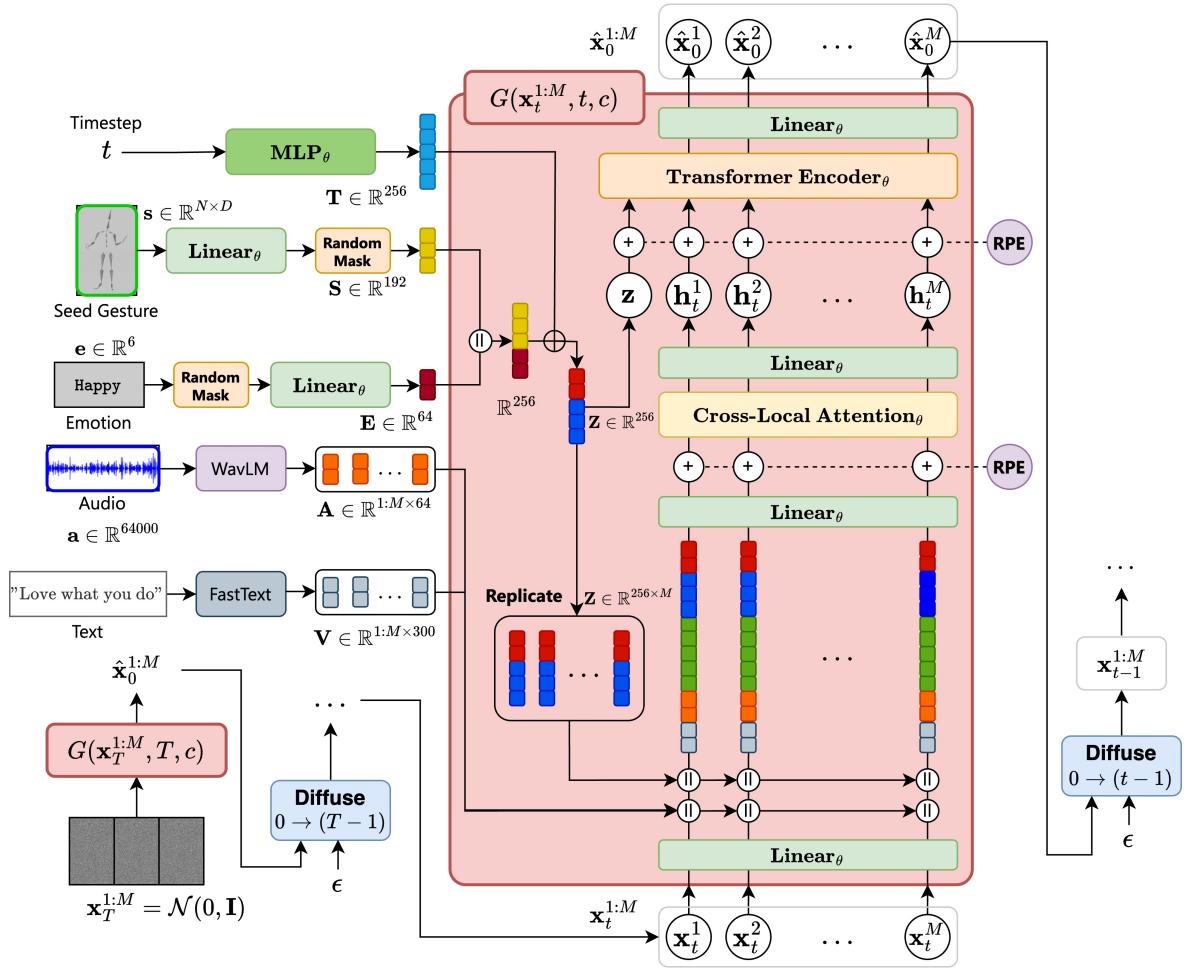
- Sinh cử chỉ có điều kiện:
  - Điều kiện cảm xúc:  $c = [s, e, a, v]$  và  $c_\emptyset = [\emptyset, \emptyset, a, v]$ .
  - Nội suy trạng thái giữa hai cảm xúc  $e_1, e_2$ , sử dụng điều kiện:  $c = [s, e_1, a, v]$  và  $c_\emptyset = [s, e_2, a, v]$ .
- Self-Attention: Mỗi liên hệ giữa các cảm xúc, cử chỉ khởi tạo và từng frame (tương tự DALL-E 2 - mỗi liên hệ giữa văn bản và ảnh).
- Concat âm thanh và văn bản (Giống ControlNet - Pixel-wise Condition)

Trong đó  $\mathbf{x}_0$  là chuỗi  $M$  khung hình cử chỉ  $\mathbf{x} \in \mathbb{R}^{1:M \times D}$  ( $D = 1141$ ), với điều kiện  $c = [s, e, a, v]$  bao gồm cử chỉ khởi tạo (seed gesture)  $s$ , cảm xúc (emotion)  $e$ , chuỗi âm thanh (audio)  $a$  tương ứng cử chỉ, và văn bản  $v$ .

Tương tự mô hình diffusion cơ bản bao gồm hai quá trình: quá trình tạo nhiễu (diffusion)  $q$  và quá trình khử nhiễu (denoising process)  $p_\theta$  với trọng số  $\theta$ . Quá trình Training và Sampling của mô hình để xuất **OHGesture** được minh họa ở hình 10.

#### 3.2.1 Quá trình trích xuất đặc trưng

Dữ liệu để học của chúng tôi được lấy từ tập dữ liệu ZeroEGGS [11]. Bao gồm âm thanh, cử chỉ khởi tạo, cảm xúc và văn bản.



Hình 11: Mô hình trong OHGesture

- **Bước thời gian**  $T \in \mathbb{R}^{256}$  : Bước thời gian ở mỗi quá trình  $t \in [0, T]$ , mục tiêu của mô hình là để với bước thời gian  $t$  bất kỳ, mô hình có thể tổng quát hoá quá trình khử nhiễu, và có thể học được việc với từng bước  $t$  thì giá trị sẽ thay đổi thế nào đối với kết quả dự đoán  $x_0$ . Giá trị timestep  $t$  được khởi tạo bằng việc mã hoá nhúng vị trí (Position Embedding) bằng hàm sinusoidal  $PE(t) = [\sin\left(\frac{t}{10000^{2i/d}}\right), \cos\left(\frac{t}{10000^{2i/d}}\right)]$ , sau đó được đưa vào một MLP (Multilayer perceptron) để biến đổi thành vector  $T \in \mathbb{R}^{256}$ .
- **Cử chỉ khởi tạo**  $S \in \mathbb{R}^{192}$ : Cử chỉ khởi tạo (seed gesture)  $s \in \mathbb{R}^{1:N \times D}$  có được từ cử chỉ ground truth  $g \in \mathbb{R}^{(N+M) \times D}$ , với mỗi frame là dữ liệu bao gồm 75 khớp xương, được xử lý theo công thức 1 để được vector 1141 chiều, trình bày ở công thức 1. Chúng tôi cắt  $N = 8$  frame đầu để làm cử chỉ khởi tạo  $s$  và  $M = 80$  (4 giây với  $20fps$ ) frame tiếp theo là nhãn để thực hiện quá trình khử nhiễu  $x^{1:M}$ . Vector  $s$  đi qua một lớp tuyến tính để được vector  $S \in \mathbb{R}^{192}$ , vector  $S$  sau đó

được Random Mask trong quá trình huấn luyện để bỏ trong trường hợp nếu một trong  $N$  bị thiếu thì sẽ ảnh hưởng như thế nào đến cử chỉ dự đoán.

- **Âm thanh A**  $\in \mathbb{R}^{1:M \times 64}$ : Tất cả dữ liệu âm thanh được giảm số sample rate (down sampled) xuống 16kHz, và âm thanh sẽ được lấy tương ứng với cử chỉ là 4s để được vector  $a \in \mathbb{R}^{64000}$ , chúng tôi sử dụng mô hình pre-train của WavLM Large [7] nhúng waveform thô vào để được vector tiềm ẩn. Sau đó sử dụng nội suy tuyến tính (interpolation) để căn chỉnh đặc trưng của vector tiềm ẩn trong WavLM theo chiều thời gian thành  $20fps$ , sau đó sử dụng một lớp tuyến tính để giảm kích thước của đặc trưng xuống còn vector đặc trưng với 64 để tạo thành ma trận cuối cùng  $A \in \mathbb{R}^{M \times 64}$ .
- **Cảm xúc E**  $\in \mathbb{R}^{64}$ : Cảm xúc (emotion)  $e \in \mathbb{R}^6$  bao gồm 6 cảm xúc khác nhau được biểu diễn thành một vector theo phương pháp one-hot encoding, sau đó chúng tôi đưa qua một vector tuyến tính để được vector đặc trưng  $E \in \mathbb{R}^{64}$ . Mục tiêu là để có thể concat được với vector khởi tạo  $S \in \mathbb{R}^{192}$  để tạo thành vector tiềm ẩn 256 chiều.
- **Văn bản V**  $\in \mathbb{R}^{1:M \times 300}$ : Văn bản là đoạn văn bản được dịch tương ứng với âm thanh, được đưa vào FastText [6] để thu được một vector có số chiều là 300. Sau đó, được replicate để căn chỉnh tương ứng với  $M$  khung hình để được vector đặc trưng  $V$  như hình 11.

### 3.2.2 Áp dụng cơ chế Attention cho bài toán sinh cử chỉ

Mục tiêu của việc áp dụng cơ chế attention trong mô hình của chúng tôi là mô hình có thể hiểu được mối liên hệ của từng khung hình với nhau, vì vậy phương pháp của chúng tôi là biểu diễn các đặc trưng của từng khung hình bằng các vector đặc trưng riêng lẻ, và với từng đặc trưng của khung hình riêng lẻ, chúng tôi mong muốn tìm được sự tương quan và mối quan hệ giữa các đặc trưng với nhau. Như hình minh họa 11, sau quá trình trích xuất đặc trưng. Để tính được sự tương quan của các đặc trưng xa, chúng tôi sẽ học để biểu diễn được các ngữ cảnh cực bộ (local context) theo [29].

Đầu tiên cử chỉ khởi tạo  $S$  và vector cảm xúc  $E$  được ghép lại với nhau để tạo thành một vectơ có kích thước 256, bởi vì kích thước 256 là kích thước vector tiềm ẩn. Sau đó được cộng thêm vector timestep  $T$  để tạo thành vector  $Z$ . Vector  $Z$  thể hiện thông tin chúng tôi thực sự muốn điều khiển bằng diffusion có điều kiện.

Để có thể căn chỉnh văn bản, âm thanh tương đồng với từng khung hình cử chỉ, chúng tôi sẽ concat các đặc trưng âm thanh, văn bản và cử chỉ đồng thời với nhau. Sau đó để căn chỉnh theo miền thời gian các đặc trưng về cảm xúc và cử chỉ khởi tạo chúng tôi sẽ copy vector  $Z$  để các bản sao của nó được ghép thành một chuỗi vector đặc trưng tương đồng với  $M$  khung hình. Sau đó đưa vào lớp local attention (chú ý cục bộ). Tương tự ý tưởng từ phương pháp Routing Transformer [30], local attention cho thấy rằng sự quan trọng trong việc xây dựng biểu diễn các vector đặc trưng trung gian. Các vector đặc trưng sẽ được cộng thêm một vector sinusoids hay mã hoá vị trí tương đối (Relative position encoding [35]) để thể hiện được đặc trưng thời gian trước khi đi qua lớp Cross-Local Attention

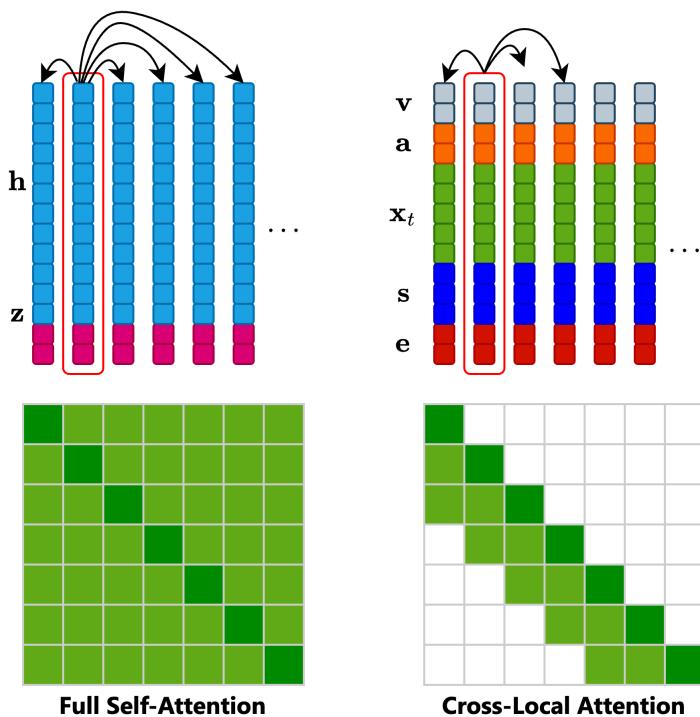
$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{M}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T + \mathbf{M}}{\sqrt{C}}\right) \mathbf{V} \quad (7)$$

Công thức Attention trên có  $\mathbf{Q}$ ,  $\mathbf{K}$ ,  $\mathbf{V}$  là các ma trận sau khi đi qua các ma trận biến đổi tuyến tính  $\mathbf{Q} = \mathbf{XW}_Q$ ,  $\mathbf{K} = \mathbf{XW}_K$ ,  $\mathbf{V} = \mathbf{XW}_V$ . Với đầu vào là ma trận biểu thị chuỗi  $M$  khung hình, với mỗi khung hình là một vector được concat từ các vector đặc trưng bao gồm cả cử chỉ khởi tạo, văn bản, âm thanh, cảm xúc và cử chỉ  $x_t$  mà chúng tôi muốn thực hiện khử nhiễu. Quá trình Local-Cross Attention được điều khiển chỉ để tính các đặc trưng cục bộ của chuyển động của các cử chỉ và đặc trưng trong các khung hình lân cận.

Hàm Attention hoạt động như một bộ từ điển, với các thông tin cuối cùng tra được là ma trận  $\mathbf{V}$  (value), còn  $\mathbf{Q}$  (query) là từ khoá muốn tìm kiếm,  $\mathbf{K}$  (key) là danh mục các từ khoá trong bộ từ điển tra cứu. Quá trình Attention sẽ tính toán mức độ tương đồng giữa  $\mathbf{Q}$  và  $\mathbf{K}$  để xác định trọng số cho các giá trị trong  $\mathbf{V}$ . Kết quả cuối cùng là tổ hợp các giá trị trong  $\mathbf{V}$ , trong đó các giá trị tương ứng với các khoá giống truy vấn nhất sẽ có trọng số cao hơn.  $\mathbf{M}$  là mặt nạ (mask) để thực hiện quá trình chú ý cục bộ.

Sau quá trình Cross-Local Attention, mô hình tiếp tục được đưa qua lớp tuyến tính để căn chỉnh tương ứng với  $M$  khung hình, bao gồm  $\mathbf{h}^{1:M}$ . Sau đó quá trình tương tự như xử lý các vector đặc trưng của mô hình BERT [9] văn bản. Chúng tôi sử dụng  $z$  là  $\mathbf{Z} \in \mathbb{R}^{256}$  token đầu tiên biểu thị đặc trưng biểu thị cho toàn bộ chuỗi  $M$  khung hình. Các vector  $\mathbf{h}$  biểu thị cho chuỗi  $M$  khung hình, tương tự như phương pháp Reformer [19], trước khi đi vào lớp Self-Attention trong Transformer Encoder, mô hình sẽ sử dụng lớp mã hoá vị trí tương đối (Relative Position Encoding - RPE) để thay thế mã hoá vị trí tuyệt đối, giúp mô hình hiệu quả hơn trong việc xử lý chuỗi dài. Khi vào

lớp Transformer Encoder [35] giúp tính toán được mối liên hệ giữa các chuỗi dữ liệu. Trong lớp Transformer Encoder mô hình sẽ áp dụng cơ chế tự chú ý tương tự như trên nhưng không sử dụng mặt nạ. Cuối cùng, đầu ra của Self-Attention được ánh xạ lại cùng kích thước như  $x_0$  sau một lớp biến đổi tuyến tính. Mục tiêu của toàn bộ mô hình  $G_\theta(x_t, t, c)$  là để học được trọng số  $\theta$ , với quá trình denoise sẽ giảm dần từ  $t = T$  đến khi  $t = 1$  ta thu được  $x_0$ .



Hình 12: Cơ chế Self-Attention trong Transformer Encoder và Cross-Local Attention

### 3.2.3 Điều khiển cảm xúc trong bài toán sinh cử chỉ

Ở các bước trên mô hình đã có thể học được cách sinh cử chỉ, nhưng để mô hình có thể học được các cảm xúc ở các tình huống khác nhau sẽ được giải quyết bằng cách tham số hóa và lần lượt thay đổi từng cảm xúc để sao cho khi thay đổi cảm xúc thì kết quả dự đoán phải có cảm xúc tương ứng.

Tương tự như các phương pháp sử dụng mô hình khử nhiễu có điều kiện [15], [34], chúng tôi sử dụng điều kiện  $c = [s, e, a, v]$ , bao gồm cử chỉ khởi tạo  $s$ , cảm xúc  $e$ , âm thanh tương ứng  $a$  và văn bản  $v$ . Mô hình diffusion có điều kiện  $c$  ở đây sẽ là tổng cả trường hợp ở từng bước  $t$  trong mô hình khử nhiễu  $G_\theta(x_t, t, c)$ , với  $c_\emptyset = [\emptyset, \emptyset, a, v]$  không điều kiện và  $c = [s, e, a, v]$  có điều kiện. Quá trình này có thể dễ dàng điều khiển bằng một lớp mặt nạ ngẫu nhiên (random mask) trên các vector đặc trưng của

cử chỉ khởi tạo  $s$  và cảm xúc  $e$ . Khi đó, mô hình chỉ việc thay đổi nhãn tương ứng với lớp mask đã lấy ngẫu nhiên để mô hình có thể tối ưu theo các điều kiện khác nhau.

$$\hat{x}_{0c,c_\emptyset,\gamma} = \gamma G(x_t, t, c) + (1 - \gamma)G(x_t, t, c_\emptyset) \quad (8)$$

Điểm đặc biệt là ta có thể dựa trên việc học có điều kiện và không có điều kiện của classifier-free guidance [15], ta có thể nội suy giữa hai cảm xúc  $e_1$  và cảm xúc  $e_2$  khác nhau bằng cách cho điều kiện  $c = [s, e_1, a, v]$  và  $c_\emptyset = [s, e_2, a, v]$ . Khi đó ta có thể viết lại  $\hat{x}_{0\gamma,c_1,c_2} = \gamma G(x_t, t, c_1) + (1 - \gamma)G(x_t, t, c_2)$ .

### 3.2.4 Quá trình huấn luyện

Thuật toán huấn luyện mô hình OHGesture bắt đầu bằng việc tính toán các giá trị và siêu tham số cần thiết như  $\gamma$ ,  $\sqrt{\alpha_t}$ ,  $\sqrt{1 - \alpha_t}$ ,  $\sqrt{\bar{\alpha}_t}$  và nhiễu ngẫu nhiên  $\epsilon_t$  cho từng bước thời gian  $t$  từ 1 đến  $T$ . Sau đó, nhãn ban đầu  $x_0$ , đại diện cho cử chỉ gốc, được lấy từ phân phối dữ liệu đã chuẩn hóa. Tiếp theo, các mặt nạ Bernoulli  $c_1$  và  $c_2$  được tạo ngẫu nhiên, mô phỏng các điều kiện khác nhau như cử chỉ, cảm xúc, âm thanh, hoặc văn bản, với một trong các mặt nạ có thể là không có thông tin cảm xúc. Sau khi có các mặt nạ, nhiễu được thêm vào để tạo thành cử chỉ nhiễu  $x_t$ , được tính bằng công thức kết hợp chuỗi cử chỉ gốc và nhiễu ngẫu nhiên. Quá trình tiếp theo là chọn ngẫu nhiên một bước thời gian  $t$  trong khoảng  $[1, T]$  và sử dụng cử chỉ nhiễu  $x_t$  cùng các mặt nạ để dự đoán lại chuỗi cử chỉ gốc thông qua mô hình, trong đó dự đoán được tính bằng một sự kết hợp của các hàm tạo ra từ các điều kiện mặt nạ. Sau khi có dự đoán, loss được tính bằng cách sử dụng HuberLoss giữa chuỗi cử chỉ gốc và dự đoán, từ đó đạo hàm loss để cập nhật các tham số trọng số  $\theta$ . Quá trình huấn luyện này được lặp lại cho đến khi mô hình hội tụ và thu được các tham số tối ưu  $\theta'$ .

---

**Algorithm 3** Thuật toán huấn luyện

---

1. Tính sẵn các giá trị và siêu tham số:  $\gamma, \sqrt{\alpha_t}, \sqrt{1 - \alpha_t}, \sqrt{\bar{\alpha}_t}$  và nhiễu ngẫu nhiên  $\epsilon_t$  tại mỗi bước  $t : 1 \rightarrow T$ . Định nghĩa lịch nhiễu  $\{\alpha_t \in (0, 1)\}_{t=1}^T$ .
2. Lấy nhãn ban đầu  $\mathbf{x}_0$  từ phân phối dữ liệu đã chuẩn hóa.
3. Random các mặt nạ Bernoulli  $c_1 = [\mathbf{s}, \mathbf{e}_1, \mathbf{a}, \mathbf{v}]$ ,  $c_2 = [\mathbf{s}, \mathbf{e}_2, \mathbf{a}, \mathbf{v}]$ , hoặc  $c_2 = [\emptyset, \emptyset, \mathbf{a}, \mathbf{v}]$ .
4. Thêm nhiễu để có cử chỉ nhiễu  $\mathbf{x}_t$ :

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t$$

5. Với mỗi bước  $t$ , lấy **ngẫu nhiên**  $t$  từ  $[1, T]$ .
6. Với  $\mathbf{x}_t$ ,  $t$  và các điều kiện mặt nạ  $c_1, c_2$ , dự đoán chuỗi cử chỉ:

$$\hat{\mathbf{x}}_{0\gamma, c_1, c_2} = \gamma G_\theta(\mathbf{x}_t, t, c_1) + (1 - \gamma) G_\theta(\mathbf{x}_t, t, c_2)$$

7. Tính loss và đạo hàm để cập nhật trọng số  $\theta$ :

$$\mathcal{L}_t = \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \epsilon_t} [\text{HuberLoss}(\mathbf{x}_0, \hat{\mathbf{x}}_0)]$$

8. Lặp lại từ bước 6 cho đến khi hội tụ, thu được các tham số tối ưu  $\theta'$ .
- 

### 3.2.5 Quá trình lấy mẫu

Để có thể sinh cử chỉ với chiều dài tùy ý, chúng tôi cắt chuỗi ban đầu thành các đoạn ngắn có chiều dài  $M$ . Trong quá trình huấn luyện, cử chỉ khởi tạo đầu tiên có thể được tạo ra bằng cách lấy ngẫu nhiên cử chỉ từ tập dữ liệu hoặc từ lấy trung bình từ đoạn cắt được. Cụ thể ở đây, sẽ lấy góc quay trung bình trong các đoạn đã cắt được. Tiếp theo chúng ta chỉ việc lấy lần lượt các frame đã sinh ra và chọn  $N = 8$  frame cuối cùng làm cử chỉ khởi tạo ở lượt tiếp theo. Đối với mỗi đoạn đã cắt ra, cử chỉ  $\mathbf{x}_t$  lần lượt sẽ được áp dụng hàm khử nhiễu  $\hat{\mathbf{x}}_0 = G_{\theta'}(\mathbf{x}_t, t, c)$ , sau khi có  $\hat{\mathbf{x}}_0$  ta sẽ Diffuse (thêm nhiễu) cho đến khi được  $\mathbf{x}_{t-1}$ , và  $\mathbf{x}_{t-1}$  sẽ tiếp tục được làm khử nhiễu cho đến bước khử nhiễu  $t = 1$  sẽ đạt được  $\mathbf{x}_0$ .

Thuật toán 4 lấy mẫu bắt đầu bằng việc khởi tạo cử chỉ nhiễu ban đầu  $\mathbf{x}_T$  từ phân phối chuẩn  $\mathcal{N}(0, \mathbf{I})$ . Sau đó, các giá trị  $\sqrt{\alpha_t}, \sqrt{1 - \alpha_t}$  và  $\sqrt{\bar{\alpha}_t}$  được lấy từ quá trình huấn luyện, cùng với giá trị  $\sigma_t$  được tính từ  $\alpha_t$  tại mỗi bước thời gian  $t$  từ 1 đến  $T$ . Mỗi đoạn âm thanh 4 giây được chia thành các chuỗi dữ liệu  $\mathbf{a}$ , và cử chỉ khởi tạo  $\mathbf{s}$  được

---

**Algorithm 4** Thuật toán lấy mẫu (sampling)

---

1. Khởi tạo với nhiễu:  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ .
  2. Các giá trị  $\sqrt{\alpha_t}$ ,  $\sqrt{1 - \alpha_t}$  và  $\sqrt{\bar{\alpha}_t}$  lấy từ quá trình huấn luyện, tính sẵn giá trị  $\sigma_t$  từ  $\alpha_t$  ở mỗi bước  $t : 1 \rightarrow T$ .
  3. Chia mỗi đoạn âm thanh 4 giây thành:  $\mathbf{a} \in \mathbb{R}^{64000}$ . Cử chỉ khởi tạo s ban đầu là trung bình dữ liệu, sau đó được lấy từ đoạn cử chỉ đã suy luận. Chọn cảm xúc mong muốn, văn bản được lấy từ âm thanh chuyển mã  $\mathbf{a}$ , tạo cặp điều kiện  $c = [\mathbf{s}, \mathbf{e}, \mathbf{a}, \mathbf{v}]$ .
  4. Với mỗi  $t$ , lấy  $t$  **tuần tự** từ  $[T, \dots, 1]$ .
  5. Tạo nhiễu ngẫu nhiên  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ .
  6. Đưa  $\mathbf{x}_t$  vào để suy luận  $\hat{\mathbf{x}}_0^{(t)} = G_{\theta'}(\mathbf{x}_t, t, c)$ .
  7. Chuyển tiếp  $\hat{\mathbf{x}}_0^{(t)}$  từ bước  $0 \rightarrow t$  để nhận  $\hat{\mathbf{x}}_{t-1}^{(t)}$ .
  8. Cộng thêm nhiễu:  $\hat{\mathbf{x}}_{t-1} = \hat{\mathbf{x}}_{t-1}^{(t)} + \sigma_t \mathbf{z}$ .
  9. Quay lại bước 4. Khi  $t = 1$ , thu được  $\hat{\mathbf{x}}_0$  từ quá trình khử nhiễu.
- 

lấy từ trung bình dữ liệu hoặc từ đoạn cử chỉ đã suy luận. Cảm xúc mong muốn và văn bản được lấy từ âm thanh chuyển mã  $\mathbf{a}$ , tạo thành một cặp điều kiện  $c = [\mathbf{s}, \mathbf{e}, \mathbf{a}, \mathbf{v}]$ . Thuật toán sau đó thực hiện các bước tuần tự, bắt đầu từ bước cuối cùng  $T$  và tiến ngược về 1. Ở mỗi bước, nhiễu ngẫu nhiên  $\mathbf{z}$  được tạo ra và mô hình dự đoán  $\hat{\mathbf{x}}_0^{(t)}$  từ cử chỉ nhiễu  $\mathbf{x}_t$ , thời gian  $t$ , và cặp điều kiện  $c$ . Dự đoán này được chuyển tiếp để tính toán  $\hat{\mathbf{x}}_{t-1}^{(t)}$  cho bước tiếp theo, và nhiễu được cộng vào để cập nhật cử chỉ tại bước  $t - 1$ . Quá trình này lặp lại cho đến khi đạt đến bước 1, khi đó thuật toán thu được cử chỉ khử nhiễu  $\hat{\mathbf{x}}_0$  là kết quả dự đoán cuối cùng.

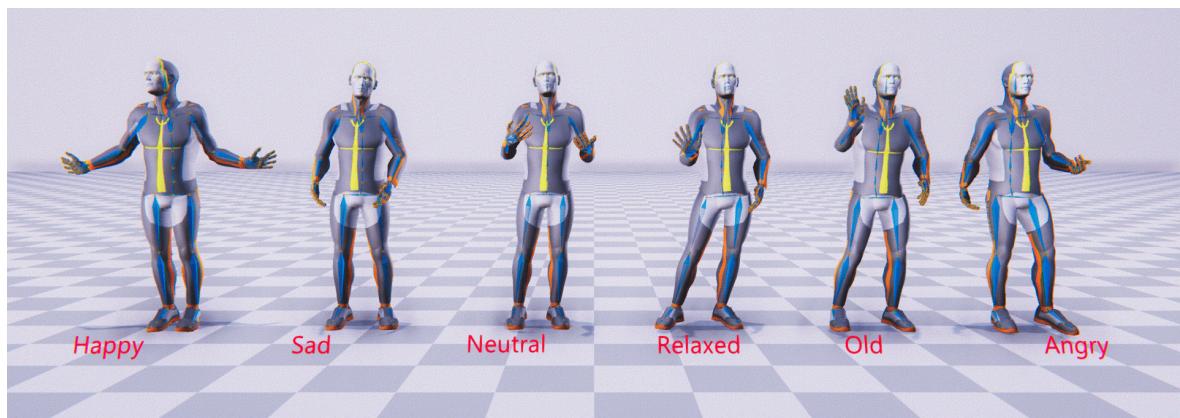
## Chương 4. THỰC NGHIỆM

### 4.1 Tập dữ liệu

Chúng tôi sử dụng tập dữ liệu ZeroEGGS [11] là một bộ dữ liệu motion capture được xây dựng để nghiên cứu và phát triển các mô hình tạo cử chỉ. Nó bao gồm 67 đoạn độc thoại do diễn viên motion capture nữ thực hiện, với tổng thời gian là 135 phút. Các đoạn hội thoại trong tập dữ liệu được biểu diễn với 6 cảm xúc khác nhau: Happy, Sad, Neutral, Old, Relaxed và Angry, giúp mô phỏng nhiều trạng thái cảm xúc khác nhau trong cử chỉ và chuyển động cơ thể. ZeroEGGS cung cấp một nền tảng phong phú để nghiên cứu khả năng kết hợp giữa bài nói và cử chỉ động, phục vụ cho việc tạo ra các mô hình có thể điều chỉnh cử chỉ tương ứng với cảm xúc và ngữ nghĩa của văn bản.

### 4.2 Quá trình xử lý dữ liệu

Dữ liệu chúng tôi bao gồm các tệp tin BVH (BioVision Motion Capture) được thu nhận từ các cảm biến bằng các hệ thống Motion Capture.



Hình 13: Minh họa về 6 cử chỉ Happy, Sad, Neutral, Old, Relaxed và Angry

**Hierachy:** bao gồm 75 Bone  $\{\mathbf{t}_i\}^{75}$ , có vị trí ban đầu  $\mathbf{t}_i = \{t_x, t_y, t_z\}$

**Motion:** Bone trong dữ liệu BVH bao gồm vị trí position<sub>local</sub>  $\in \mathbb{R}^3$  và góc quay rotation<sub>local</sub>  $\in \mathbb{R}^3$ .

Chúng tôi chuyển dữ liệu từ góc quay Euler sang góc quay Quaternion, với góc quay Quaternion là một véc-tơ gồm 4 phần tử.

Quá trình này được trình bày ở phụ lục B

### 4.3 Quá trình huấn luyện

Toàn bộ quá trình huấn luyện mô hình được thực hiện trong vòng 1 ngày với các tham số sau: số bước huấn luyện  $T = 1000$ , sử dụng GPU Nvidia 3090, và chia tập dữ liệu theo tỷ lệ 8 : 1 : 1 cho các tập training, testing và validation. Learning rate được thiết lập là  $3 \times 10^{-5}$ , với batch size là 384 và tổng cộng 300,000 mẫu.  $\gamma = 0.1$ .

$\beta$  bắt đầu từ 0.5 → 0.999

Quá trình huấn luyện được triển khai trên mã nguồn chương trình có sẵn tại: hmthanh/OHGesture .

### 4.4 Quá trình sử dụng Unity để kết xuất

Để trực quan hóa quá trình sinh cử chỉ từ dữ liệu đầu ra của mô hình, chúng tôi sử dụng Unity, kết thừa mã nguồn từ mô hình DeepPhase [33] . Dữ liệu sau khi sinh là file BVH (BioVision Motion Capture), trong Unity chúng tôi bổ sung mã nguồn CSharp để kết xuất theo vị trí toạ độ và nhãn tương ứng, với vị trí và góc quay của các xương được biểu diễn dưới dạng quaternion.

Chi tiết phần render cử chỉ được sinh ra, tôi trình bày ở phụ lục C.

Mã nguồn chương trình Unity được chúng tôi công khai ở DeepGesture/DeepGesture-Unity

## Chương 5. KẾT QUẢ VÀ ĐÁNH GIÁ

### 5.1 Phương pháp đánh giá

Quá trình đánh giá được thực hiện qua hai độ đo chính là: Mean Opinion Scores (MOS) và Fréchet Inception Distance (FID).

#### 5.1.1 Mean Opinion Scores (MOS)

Hiện tại, chưa có một độ đo chung cho bài toán sinh cử chỉ, đặc biệt là sinh cử chỉ từ âm thanh, vì vậy chúng tôi dựa vào đánh giá chủ quan của con người để thực hiện các đánh giá thực nghiệm. Tương tự như các phương pháp trước đây [42]; [21]; các mô hình sinh cử chỉ điều khiển bằng giọng nói vẫn thiếu các chỉ số mục tiêu phản ánh một cách nhất quán với nhận thức chủ quan của con người [2]. MOS được đo lường thông qua ba tiêu chí:

- Human-likeness (Mức độ giống con người)
- Gesture-Speech Appropriateness (Sự phù hợp giữa cử chỉ và lời nói)
- Gesture-style Appropriateness (Sự phù hợp giữa phong cách cử chỉ)

#### 5.1.2 Fréchet Inception Distance (FID)

FID đo sự tương đồng về phân phối giữa cử chỉ sinh ra g và cử chỉ thực tế r. Công thức tính FID là:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr} \left( \Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g} \right) \quad (1)$$

Trong đó:

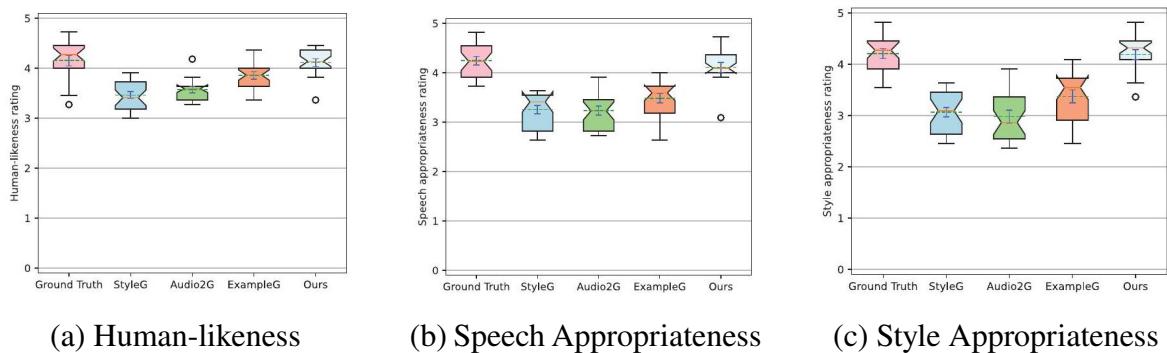
- $\mu$ : Trung bình các đặc trưng của cử chỉ.

- $\Sigma$ : Ma trận hiệp phương sai của đặc trưng.
- $\text{Tr}$ : Tổng đường chéo chính của ma trận hiệp phương sai.
- $\sqrt{\Sigma_r \Sigma_g}$ : Sự tương đồng giữa các phân phối cursive chỉ thực tế và cursive chỉ sinh ra.

FID thấp cho thấy phân phối của cursive chỉ sinh ra gần giống với cursive chỉ thực tế, trong khi FID cao gợi ý sự khác biệt lớn, cho thấy chất lượng cursive chỉ sinh ra kém hơn.

## 5.2 So sánh với các phương pháp hiện tại

Ở đây chúng tôi sử dụng lại kết quả đánh giá của mô hình baseline **DiffuseStyleGesture** [38] trong độ đo về cảm nhận đánh giá của con người do lĩnh vực sinh cursive vẫn còn rất mới, chi phí để ước lượng các mô hình còn lớn nên chúng tôi không thể đánh giá được mô hình OHGesture. Vì vậy các kết quả này chưa bao gồm thông tin kết quả về mô hình **OHGesture** mà chúng tôi đề xuất.



Để hiểu hiệu suất thị giác thực tế của phương pháp của chúng tôi, chúng tôi tiến hành một nghiên cứu người dùng so sánh các cursive chỉ được tạo ra từ phương pháp của chúng tôi và dữ liệu chụp chuyển động thực tế. Độ dài của các đoạn clip đánh giá dao động từ 11 đến 51 giây, với độ dài trung bình là 31.6 giây, dài hơn so với các đoạn trong đánh giá GENEVA [42] (8-10 giây), vì thời gian dài hơn có thể tạo ra kết quả rõ ràng và thuyết phục hơn [39]. Người tham gia đánh giá trên thang điểm từ 5 đến 1, với các nhãn từ excellent, good, fair, poor, đến bad.

Name	Human ↑ likeness	Gesture-speech ↑ appropriateness
Ground Truth	$4.15 \pm 0.11$	$4.25 \pm 0.09$
Ours	<b><math>4.11 \pm 0.08</math></b>	<b><math>4.11 \pm 0.10</math></b>
– WavLM	$4.05 \pm 0.10$	$3.91 \pm 0.11$
– Cross-local attention	$3.76 \pm 0.09$	$3.51 \pm 0.15$
– Self-attention	$3.55 \pm 0.13$	$3.08 \pm 0.10$
– Attention + GRU	$3.10 \pm 0.11$	$2.98 \pm 0.14$
+ Forward attention	$3.75 \pm 0.15$	$3.23 \pm 0.24$

Bảng 5.1: Kết quả của các nghiên cứu loại bỏ (Ablation studies). “–” chỉ các mô-đun không được sử dụng và “+” chỉ các mô-đun bổ sung. Chữ in đậm chỉ ra chỉ số tốt nhất.

### 5.3 Xây dựng và tiêu chuẩn hoá hệ thống đánh giá kết quả sinh cử chỉ

Hiện nay các mô hình sinh (gesture generation) cử chỉ được quan tâm nghiên cứu với rất nhiều mô hình khác nhau, tuy nhiên do không có độ đo chung, vì các độ đo truyền thống như FID (Fréchet Inception Distance) hay IS (Inception Score),.. không thể hiện được hết các tính chất Giống người (Human-likeness), tính phù hợp với giọng nói (Speech Appropriateness), và tính phù hợp với phong cách (Style Appropriateness) của cử chỉ. Các mô hình cũng được thực hiện và huấn luyện trên một tập dữ liệu khác nhau, vì vậy rất khó để biết được mô hình nào đã đạt kết quả tốt hơn, và mô hình nào là mô hình state-of-the-art, từ đó khó có thể đạt được sự tiến bộ trong lĩnh vực sinh cử chỉ. Việc thiếu tiêu chuẩn chung để đánh giá trong cộng đồng nghiên cứu khiến chúng tôi mong muốn xây dựng một hệ thống xếp hạng trực tuyến [26] GENEIA Leaderboard, là một bảng xếp hạng các mô hình sinh cử chỉ. Chúng tôi thu thập và xử lý một tập dữ liệu cử chỉ từ nhiều ngôn ngữ và từ nhiều tập dữ liệu khác nhau và tiêu chuẩn hoá để tạo ra một tập dữ liệu duy nhất. Sau đó, chúng tôi sẽ mời các tác giả của các mô hình để học và dự đoán trên tập dữ liệu đã được tiêu chuẩn hoá, sau khi có kết quả sinh cử chỉ, chúng tôi sẽ thuê những người tham gia nghiên cứu trên Prolific để đánh giá và xếp hạng kết quả sinh cử chỉ của các mô hình. Hiện tại chúng tôi đang xây dựng hệ thống trực tuyến hemvip/hemvip.github.io với mục tiêu đánh giá kết quả

sinh của cử chỉ thông qua việc thuê các người tham gia đánh giá kết quả thông qua Prolific. Chúng tôi sẽ bổ sung các đánh giá về mô hình OHGesture dựa trên các điểm số trên.

Thông qua hệ thống đánh giá này, nghiên cứu kỳ vọng sẽ thiết lập một tiêu chuẩn chung, từ đó tạo động lực cho sự tiến bộ trong lĩnh vực sinh cử chỉ.

## **Chương 6. KẾT LUẬN**

### **6.1 Kết quả đạt được**

Trong luận văn này, chúng tôi đã phát triển và thử nghiệm thành công mô hình sinh cử chỉ OHGesture, một hệ thống có khả năng tạo ra cử chỉ rất tự nhiên, mang lại cảm giác giống người. Điểm nổi bật của mô hình là khả năng đồng bộ chính xác giữa cử chỉ và cảm xúc, nội dung của âm thanh đầu vào, đồng thời có khả năng suy luận vượt ra khỏi các dữ liệu được huấn luyện ban đầu. Điều này có nghĩa là mô hình dựa trên Diffusion không chỉ phụ thuộc vào các mẫu dữ liệu cử chỉ đã được học mà còn có thể có tính khái quát hóa cao, và có thể sinh cử chỉ cho các âm thanh và ngữ cảnh có xác xuất dữ liệu thấp.

Một điểm đáng chú ý khác trong nghiên cứu này là việc mở rộng dữ liệu đầu vào. Chúng tôi không chỉ giới hạn ở cử chỉ, âm thanh và nhãn cảm xúc mà còn tích hợp thêm các công cụ chuyển đổi văn bản thành giọng nói để chuyển âm thanh thành văn bản. Nhờ có thêm một đặc trưng về văn bản, mô hình có thể bổ sung thêm đặc trưng về ngữ nghĩa của cử chỉ và giúp mô hình có thêm nguồn thông tin để đạt kết quả sinh tốt hơn, đồng thời giúp hệ thống hiểu rõ hơn ngữ cảnh và tạo ra cử chỉ phù hợp với từng nội dung cụ thể.

### **6.2 Ưu và nhược điểm của mô hình**

Mô hình sinh cử chỉ OHGesture có nhiều ưu điểm đáng kể, góp phần quan trọng trong việc phát triển các hệ thống tương tác người-máy tự nhiên và linh hoạt hơn. Tuy nhiên, vẫn còn một số nhược điểm để cải thiện hiệu quả trong tương lai.

#### **Ưu điểm:**

- Độ chân thực cao: Dựa vào kết quả sinh cử chỉ, có thể thấy mô hình OHGesture đã đạt được kết quả sinh cử chỉ có độ giống người cao. Các cử chỉ sinh ra phản

ánh được sắc thái và nhịp điệu của âm thanh, giúp hệ thống có khả năng phản hồi đồng bộ với thông tin cảm xúc và nội dung của lời nói.

- **Khả năng tổng quát tốt:** Nhờ mô hình khử nhiễu có thể phủ được các điểm dữ liệu có các xác suất thấp, mô hình có thể suy luận cử chỉ cho các tình huống và trạng thái cảm xúc chưa từng xuất hiện trong tập huấn luyện, cho thấy tiềm năng ứng dụng trong nhiều bối cảnh thực tế.
- **Có khả năng kiểm soát** được nhiều đặc trưng khác nhau: Mô hình diffusion có khả năng điều khiển được các cảm xúc khác nhau, có khả năng nội suy trạng thái giữa các cảm xúc khác nhau.

### **Nhược điểm:**

- Mô hình hiện chưa có khả năng sampling theo thời gian thực (real time) và cần nhiều bước để ra kết quả cuối cùng.
- Thông tin đặc trưng theo  $D = 1141$  được xử lý như một tấm ảnh, không thể hiện đúng đặc trưng chuyển động.
- Phụ thuộc vào dữ liệu đầu vào chất lượng cao: Mô hình yêu cầu dữ liệu âm thanh đầu vào có chất lượng tốt và rõ ràng để đảm bảo độ chính xác của cử chỉ sinh ra. Khi âm thanh đầu vào bị nhiễu hoặc chứa nhiều biến thể cảm xúc khó phân biệt, độ chính xác của cử chỉ sinh ra có thể giảm sút.

## **6.3 Phương hướng phát triển và nghiên cứu trong tương lai**

Trong tương lai, có nhiều hướng phát triển tiềm năng để cải thiện và mở rộng mô hình sinh cử chỉ OHGesture, nhằm đáp ứng tốt hơn các yêu cầu thực tế và nâng cao tính ứng dụng của hệ thống. Một số hướng nghiên cứu và phát triển chính bao gồm:

- **Tối ưu hóa mô hình** để có thể chạy theo thời gian thực, hiện tại mô hình phải chia chuỗi âm thanh, và kết quả sinh phải được import vào Unity mới có thể render. Trong tương lai, chúng tôi mong muốn có thể xây dựng các hệ thống với thời gian thực, để có thể tương tác và tăng tính ứng dụng của mô hình

- Tối ưu hóa quá trình sampling và giảm số bước lấy mẫu: Hiện tại, quá trình sinh cử chỉ đòi hỏi một số bước lấy mẫu (sampling steps) tương đối lớn, ảnh hưởng đến tốc độ và hiệu quả của hệ thống. Việc tối ưu hóa để giảm số bước sampling mà không làm suy giảm chất lượng cử chỉ sinh ra sẽ giúp mô hình đáp ứng nhanh hơn và phù hợp cho các ứng dụng thời gian thực.
- Tích hợp và thử nghiệm các kỹ thuật embedding mới: Sử dụng các phương pháp embedding mới, đa dạng hóa thông tin đầu vào có thể giúp mô hình hiểu và phản ánh tốt hơn ngữ cảnh và cảm xúc của giọng nói. Đây cũng là một hướng phát triển để nâng cao khả năng của hệ thống trong việc sinh cử chỉ phù hợp với ngữ nghĩa của các ngôn ngữ khác nhau.
- Mở rộng sang đa ngôn ngữ và đa văn hóa: Hiện tại, mô hình chủ yếu hoạt động với các dữ liệu âm thanh tiếng Anh. Nghiên cứu mở rộng mô hình để sinh cử chỉ cho các ngôn ngữ và nền văn hóa khác nhau sẽ là một bước tiến quan trọng, giúp hệ thống trở nên đa dạng và ứng dụng rộng rãi hơn.
- Kết hợp với mô hình DeepPhase [33] để đạt hiệu quả sinh cử chỉ thời gian thực: Mục tiêu của chúng tôi là tích hợp OHGesture với mô hình DeepPhase để phát triển hệ thống có khả năng phản hồi cử chỉ trong thời gian thực, ứng dụng trong các tình huống giao tiếp tự nhiên như hội thoại người-máy và các hệ thống điều khiển bằng giọng nói. Với mục tiêu là sẽ học các đặc trưng về pha của các chuyển động để có thể trích xuất được các đặc trưng chuyển động, thay vì sử dụng đặc trưng như một tấm ảnh của mô hình.
- Cải thiện đánh giá khách quan bằng các độ đo tự động: Để giảm sự phụ thuộc vào các đánh giá chủ quan, cần phát triển và tích hợp các phương pháp đánh giá tự động đáng tin cậy, cho phép mô hình có thể tự đánh giá và điều chỉnh theo các chỉ số khách quan.

## 6.4 Đóng góp của luận văn

Trong luận văn này, chúng tôi đã phát triển hệ thống sinh cử chỉ OHGesture, với các đóng góp quan trọng như sau:

- Phát triển mô hình sinh cử chỉ dựa trên Diffusion: Hệ thống OHGesture được thiết kế để tạo ra các cử chỉ đồng bộ với âm thanh đầu vào và phản ánh đúng

cảm xúc. Mô hình này còn có khả năng tổng quát hoá, cho phép sinh cử chỉ ngay cả với các mẫu âm thanh ngoài dữ liệu huấn luyện, mang lại độ chân thực cao.

- Công khai mã nguồn và mô hình trên các nền tảng mở: Để thúc đẩy ứng dụng và cải tiến từ cộng đồng, chúng tôi đã cung cấp mã nguồn trên GitHub, chúng tôi mở rộng mã nguồn, và được công khai ở hmthanh/OHGesture và một phiên bản pretrained trên Huggingface huggingface.co/openhuman/openhuman, giúp các nhà nghiên cứu khác dễ dàng tiếp cận, tái hiện và mở rộng hệ thống.
- Tích hợp thêm văn bản và âm thanh chuyển ngữ trong quá trình sinh cử chỉ: Với tập dữ liệu ZeroEGGS chỉ bao gồm âm thanh, cử chỉ và nhãn cảm xúc, chúng tôi sử dụng các API của Azure và Google để transcribe các tệp âm thanh thành văn bản. Giúp mở rộng dữ liệu đầu vào của mô hình sinh cử chỉ thêm đặc trưng văn bản, giúp hệ thống có thêm ngữ cảnh để sinh ra các cử chỉ phù hợp hơn với nội dung cụ thể.
- Đóng góp vào việc xây dựng hệ thống đánh giá chuẩn hóa: Chúng tôi đang phát triển một hệ thống xếp hạng trực tuyến GENEVA Leaderboard [26] cho các mô hình sinh cử chỉ. Chúng tôi đã thu thập và xử lý các dữ liệu cử chỉ từ nhiều nguồn ngôn ngữ và tập dữ liệu khác nhau, tạo thành một tập dữ liệu chuẩn hóa duy nhất. Xây dựng một bản xếp hạng để so sánh nhiều mô hình khác nhau. Và sử dụng người để đánh giá các mô hình sinh, giúp đánh giá chính xác kết quả sinh của cử chỉ so với các độ đo trước đây, vốn không thể đo được sự phức tạp và đa dạng trong chuyển động của cử chỉ tương ứng với âm thanh. Điều này giúp tạo nên một nền tảng dữ liệu thống nhất, hỗ trợ việc đánh giá đồng nhất giữa các mô hình. Từ đó thúc đẩy tính thống nhất trong lĩnh vực sinh cử chỉ trong cộng đồng nghiên cứu.
- Xây dựng công cụ trực quan hóa bằng Unity: Các hệ thống trực quan hóa cử chỉ hiện nay đề dự trên Blender, và không đạt kết quả hiển thị tốt khi sinh cử chỉ, bằng việc kết thừa mã nguồn từ mô hình [33], chúng tôi đã tạo ra một hệ thống trực quan hóa quá trình sinh cử chỉ và so sánh giữa các mô hình.
- Xây dựng hướng phát triển trong tương lai: Dựa trên nền tảng của mô hình diffusion và hiểu rõ bản chất của quá trình sinh cử chỉ, chúng tôi có thể kết

hợp các mô hình sinh cử chỉ dựa trên pha với các thuật toán xử lý và trích xuất thông tin, nhằm tối ưu hóa chất lượng và tính phù hợp của các cử chỉ được sinh ra. Hướng phát triển này mở ra triển vọng cho các cải tiến sâu rộng trong việc tương tác giữa cử chỉ và các yếu tố ngữ cảnh phức tạp như biểu cảm khuôn mặt, ngữ điệu và động lực cảm xúc, tạo nền tảng cho những bước tiến trong các ứng dụng giao tiếp người - máy và các lĩnh vực liên quan khác.

## 6.5 Lời kết

Qua quá trình thử nghiệm và phân tích kết quả sinh cử chỉ thực tế, mô hình OHGesture của chúng tôi đã kế thừa và phát triển từ mô hình DiffuseStyleGesture, có khả năng sinh cử chỉ chân thực, không chỉ trên các mẫu dữ liệu trong tập huấn luyện mà còn mở rộng được với những âm thanh không có trong dữ liệu huấn luyện, ví dụ như giọng nói của Steven Jobs. Điều này minh chứng cho tiềm năng của mô hình Diffusion trong việc sinh cử chỉ cho các dữ liệu hiếm, nơi xác suất dữ liệu thấp.

Bên cạnh đó, chúng tôi đã đóng góp các mã nguồn chỉnh sửa trên Github, bao gồm quá trình kết xuất (render) và xử lý dữ liệu trên nền tảng Unity, tạo nền tảng thuận lợi cho các nghiên cứu và cải tiến tiếp theo đối với mô hình OHGesture. Việc kết hợp thêm văn bản vào quá trình sinh cử chỉ cũng là một bước đột phá, mở ra nhiều hướng phát triển ứng dụng trong các lĩnh vực yêu cầu tương tác tự nhiên và hiệu quả hơn.

## TÀI LIỆU THAM KHẢO

- [1] Simon Alexanderson, Gustav Eje Henter, Taras Kucherenko, and Jonas Beskow. Style-controllable speech-driven gesture synthesis using normalising flows. *Computer Graphics Forum*, 39(2):487–496, 2020.
- [2] Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *CoRR*, abs/2211.09707, 2022.
- [3] Kirsten Bergmann, Volkan Aksu, and Stefan Kopp. The relation of speech and gestures: Temporal synchrony follows semantic synchrony. In *Proceedings of the 2nd Workshop on Gesture and Speech in Interaction (GeSpIn 2011)*, 2011.
- [4] Uttaran Bhattacharya, Elizabeth Childs, Nicholas Rewkowski, and Dinesh Manocha. Speech2affectivegestures: Synthesizing co-speech gestures with generative adversarial affective expression learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 2027–2036, 2021.
- [5] 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.
- [6] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.
- [7] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. Wavlm: Large-scale self-supervised pre-training for full

- stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, October 2022.
- [8] Chung-Cheng Chiu, Louis-Philippe Morency, and Stacy Marsella. Predicting co-verbal gestures: A deep and temporal modeling approach. In *Intelligent Virtual Agents: 15th International Conference, IVA 2015, Delft, The Netherlands, August 26-28, 2015, Proceedings 15*, pages 152–166. Springer, 2015.
  - [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
  - [10] Paul Ekman and Wallace V Friesen. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *semiotica*, 1(1):49–98, 1969.
  - [11] Saeed Ghorbani, Ylva Ferstl, Daniel Holden, Nikolaus F. Troje, and Marc-André Carbonneau. Zeroeggs: Zero-shot example-based gesture generation from speech, 2022.
  - [12] 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.
  - [13] Dai Hasegawa, Naoshi Kaneko, Shinichi Shirakawa, Hiroshi Sakuta, and Kazuhiko Sumi. Evaluation of speech-to-gesture generation using bi-directional lstm network. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 79–86, 2018.
  - [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
  - [15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
  - [16] Chien-Ming Huang and Bilge Mutlu. Robot behavior toolkit: generating effective social behaviors for robots. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 25–32, 2012.

- [17] Chris Jones. Ed - realistic digital human, 2014. Accessed: 2024-11-18.
- [18] Michael Kipp. *Gesture generation by imitation: From human behavior to computer character animation*. Universal-Publishers, 2005.
- [19] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.
- [20] Taras Kucherenko, Patrik Jonell, Sanne Van Waveren, Gustav Eje Henter, Simon Alexandersson, Iolanda Leite, and Hedvig Kjellström. Gesticulator: A framework for semantically-aware speech-driven gesture generation. In *Proceedings of the 2020 international conference on multimodal interaction*, pages 242–250, 2020.
- [21] Taras Kucherenko, Patrik Jonell, Youngwoo Yoon, Pieter Wolfert, and Gustav Eje Henter. A large, crowdsourced evaluation of gesture generation systems on common data: The genea challenge 2020. In *26th international conference on intelligent user interfaces*, pages 11–21, 2021.
- [22] Sergey Levine, Philipp Krähenbühl, Sebastian Thrun, and Vladlen Koltun. Gesture controllers. In *Acm siggraph 2010 papers*, pages 1–11. ACM, 2010.
- [23] Xian Liu, Qianyi Wu, Hang Zhou, Yinghao Xu, Rui Qian, Xinyi Lin, Xiaowei Zhou, Wayne Wu, Bo Dai, and Bolei Zhou. Learning hierarchical cross-modal association for co-speech gesture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10462–10472, 2022.
- [24] Evelyn McClave. Gestural beats: The rhythm hypothesis. *Journal of psycholinguistic research*, 23(1):45–66, 1994.
- [25] Adam Metallo, Vincent Rossi, Jonathan Blundell, Günter Waibel, Paul Graham, Graham Fyffe, Xueming Yu, and Paul Debevec. Scanning and printing a 3d portrait of president barack obama. In *SIGGRAPH 2015: Studio*, pages 1–1. ACM, 2015.
- [26] Rajmund Nagy, Hendric Voss, Youngwoo Yoon, Taras Kucherenko, Teodor Nikolov, Thanh Hoang-Minh, Rachel McDonnell, Stefan Kopp, Michael Neff,

and Gustav Eje Henter. Towards a genea leaderboard—an extended, living benchmark for evaluating and advancing conversational motion synthesis. *arXiv preprint arXiv:2410.06327*, 2024.

- [27] Michael Neff, Michael Kipp, Irene Albrecht, and Hans-Peter Seidel. Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions On Graphics (TOG)*, 27(1):1–24, 2008.
- [28] Shenhan Qian, Zhi Tu, Yihao Zhi, Wen Liu, and Shenghua Gao. Speech drives templates: Co-speech gesture synthesis with learned templates. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11077–11086, 2021.
- [29] Jack W Rae and Ali Razavi. Do transformers need deep long-range memory. *arXiv preprint arXiv:2007.03356*, 2020.
- [30] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.
- [31] Thomas A Sebeok and Jean Umiker-Sebeok. *Advances in visual semiotics: The semiotic web 1992-93*, volume 118. Walter de Gruyter, 2011.
- [32] Yang Song. Score-based generative modeling: A brief introduction, 2021. Accessed: 2024-11-18.
- [33] Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.
- [34] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.
- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

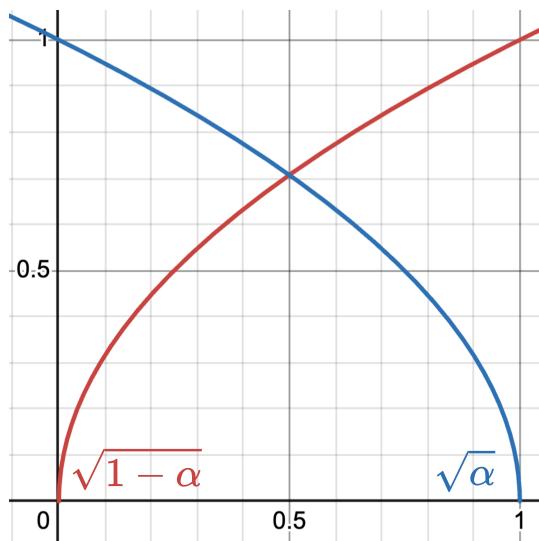
- [36] Rebecca A Webb. *Linguistic features of metaphoric gestures*. University of Rochester, 1997.
- [37] Lilian Weng. What are diffusion models? *lilianweng.github.io*, Jul 2021.
- [38] Sicheng Yang, Zhiyong Wu, Minglei Li, Zhensong Zhang, Lei Hao, Weihong Bao, Ming Cheng, and Long Xiao. Diffusestylegesture: Stylized audio-driven co-speech gesture generation with diffusion models. *arXiv preprint arXiv:2305.04919*, 2023.
- [39] Sicheng Yang, Zhiyong Wu, Minglei Li, Mengchen Zhao, Jiuxin Lin, Liyang Chen, and Weihong Bao. The reprgesture entry to the genea challenge 2022. In *Proceedings of the 2022 International Conference on Multimodal Interaction*, pages 758–763, 2022.
- [40] Sicheng Yang, Haiwei Xue, Zhensong Zhang, Minglei Li, Zhiyong Wu, Xiaofei Wu, Songcen Xu, and Zonghong Dai. The diffusestylegesture+ entry to the genea challenge 2023. In *Proceedings of the 2023 International Conference on Multimodal Interaction*, 2023.
- [41] 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. *ACM Transactions on Graphics (TOG)*, 39(6):1–16, 2020.
- [42] Youngwoo Yoon, Pieter Wolfert, Taras Kucherenko, Carla Viegas, Teodor Nikolov, Mihail Tsakov, and Gustav Eje Henter. The genea challenge 2022: A large evaluation of data-driven co-speech gesture generation. In *Proceedings of the 2022 International Conference on Multimodal Interaction*, pages 736–747, 2022.
- [43] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022.

## **DANH MỤC CÔNG TRÌNH CỦA HVCH**

1. Towards a GENEVA Leaderboard – an Extended, Living Benchmark for Evaluating and Advancing Conversational Motion Synthesis [26].

## Phụ lục A. CÁC THAM SỐ TRONG DIFFUSION

### A.1 Sự thay đổi của $\sqrt{\alpha}$ và $\sqrt{1 - \alpha}$



Hình 14: Sự thay đổi của  $\sqrt{\alpha}$  và  $\sqrt{1 - \alpha}$

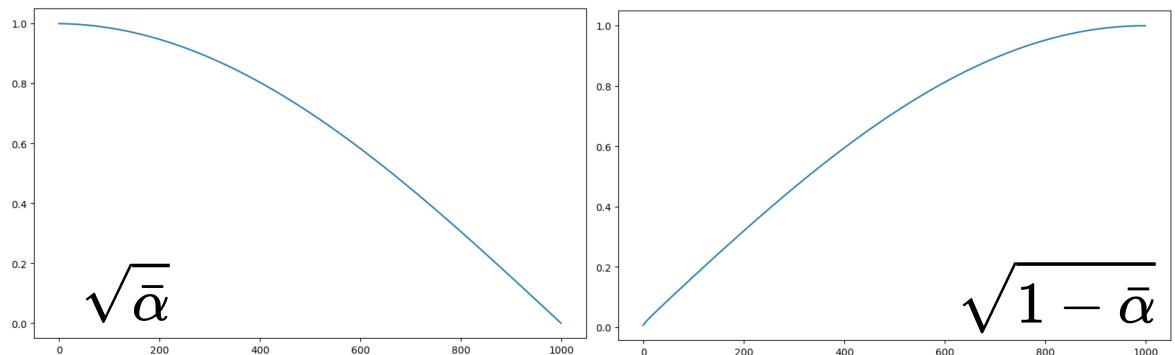
Trong quá trình diffusion, hệ thống muốn giảm dần sự hiện diện của  $x$  và tăng dần sự hiện diện của nhiễu  $\epsilon_t$ . Với  $x_t = \sqrt{\alpha_t}x_t + \sqrt{1 - \alpha_t}\epsilon_t$ . Các hệ số  $\sqrt{\alpha_t}$  và  $\sqrt{1 - \alpha_t}$  giúp điều khiển quá trình này như sau:

- $\sqrt{\alpha_t}$ : Ban đầu giá trị lớn nhưng giảm dần để giảm sự ảnh hưởng vào kết quả tổng giữa nhiễu và  $x_t$
- $\sqrt{1 - \alpha_t}$ : Ban đầu giá trị nhỏ nhưng tăng dần theo từng bước. Với mục tiêu là đến cuối cùng thì sự ảnh hưởng của nhiễu càng lớn và trở thành nhiễu hoàn toàn.

Cả hai đại lượng này thay đổi theo thời gian trong quá trình diffusion, từ đó quyết định mức độ nhiễu được thêm vào từng bước  $t$  và mức độ giữ lại trạng thái trước đó qua từng bước.

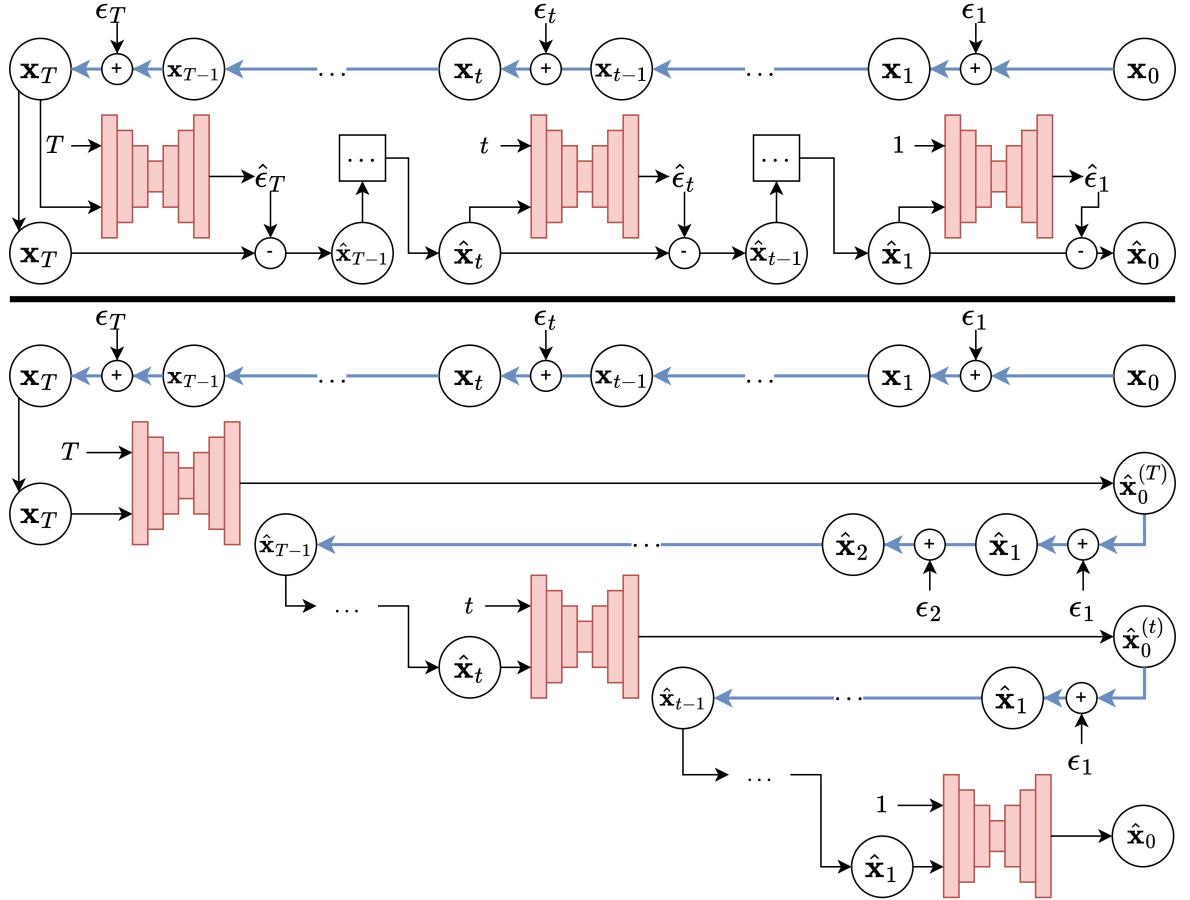
## A.2 Sự thay đổi của $\sqrt{\bar{\alpha}}$ và $\sqrt{1 - \bar{\alpha}}$

$\bar{\alpha}_t = \prod_{i=1}^t \alpha_i \cdot \sqrt{1 - \bar{\alpha}_t} = \sqrt{1 - \prod_{i=1}^t \alpha_i}$ . Cả  $\sqrt{\bar{\alpha}}$  và  $\sqrt{1 - \bar{\alpha}}$  đóng một vai trò quan trọng trong quá trình huấn luyện và sinh cử chỉ. Giúp kiểm soát mức độ nhiễu được thêm vào và cử chỉ, ta có thể thấy  $\sqrt{\bar{\alpha}}$  giảm dần còn  $\sqrt{1 - \bar{\alpha}}$  thì tăng dần..



Hình 15: Quá trình thay đổi của  $\sqrt{\bar{\alpha}}$  và  $\sqrt{1 - \bar{\alpha}}$

### A.3 So sánh $\epsilon$ objective và $x_0$



Hình 16: So sánh  $\epsilon$  objective (bên trên) và  $x_0$  (bên dưới).

- **$\epsilon$  objective** : mô hình sẽ dự đoán lỗi. Bắt đầu từ việc forward quá trình gây nhiễu để lấy được  $x_T$ , khi có được  $x_T$  ta sẽ sử dụng  $x_T \in \mathcal{N}(0, I)$  để đưa vào quá trình denoise. Trong quá trình denoise, mô hình sẽ dự đoán nhiễu  $\hat{\epsilon}_t$  đã được thêm vào từ quá trình forward, là nhiễu  $\epsilon_t$  và tối ưu lỗi giữa nhiễu dự đoán và nhiễu thực tế từ quá trình forward.
- **$x_0$  objective** : tương tự mô hình sẽ forward quá trình gây nhiễu để lấy được  $x_T$ , khi có được  $x_T$  ta sẽ sử dụng  $x_T \in \mathcal{N}(0, I)$  để đưa vào quá trình denoise. Mô hình sẽ dự đoán trực tiếp  $x_0$ , sau khi có  $x_0$  mô hình sẽ tiếp tục gây nhiễu (Diffuse) đến bước thứ  $x_{t-1}$ , và tiếp tục sử dụng  $x_{t-1}$  để đưa vào mô hình dự đoán  $x_0$

## Phụ lục B. QUÁ TRÌNH XỬ LÝ DỮ LIỆU BVH

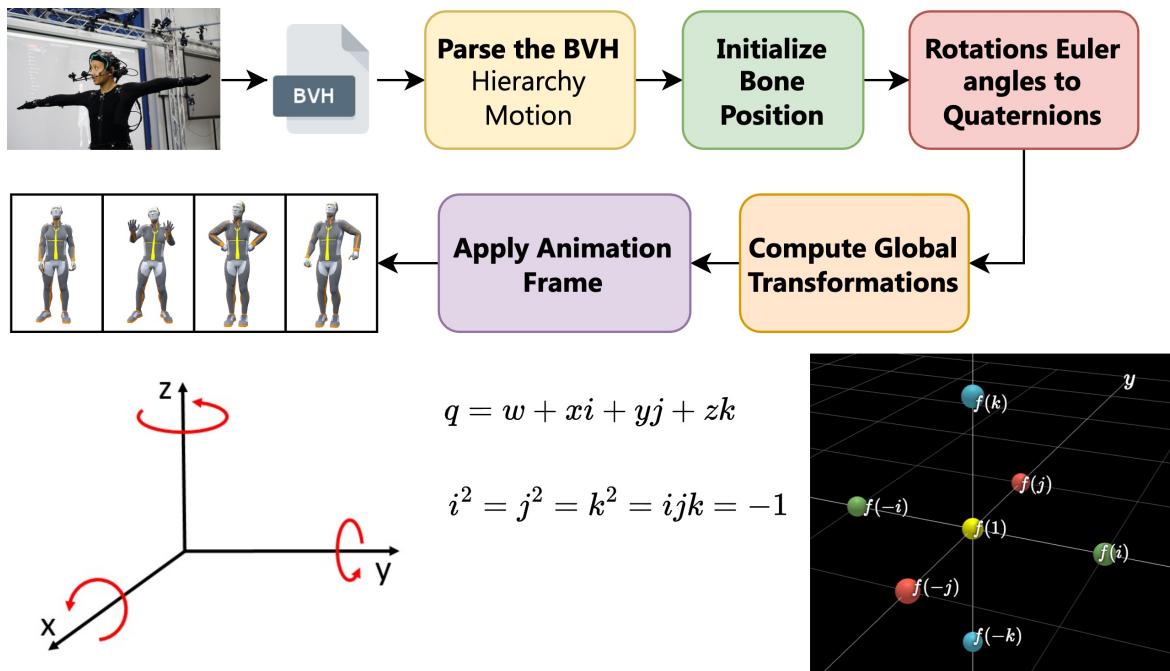
### B.1 Cấu trúc khung xương của một nhân vật

Một số tên khung xương nhân vật trong 75 khung xương chuyển động bao gồm:

Hips, Spine, Neck, Head, RightShoulder, RightArm, RightForeArm, RightHand, LeftShoulder, LeftArm, LeftForeArm, LeftHand, RightUpLeg, RightLeg, RightFoot, RightToeBase, LeftUpLeg, LeftLeg, LeftFoot, LeftToeBase, ...



Hình 1: Khung xương của một nhân vật



Hình 16: Quá trình xử lý tệp tin BVH

## B.2 Cấu trúc tệp dữ liệu BVH

```

HIERARCHY
ROOT Hips
{
    OFFSET -318.810852 89.264893 -575.244751
    CHANNELS 6 Xposition Yposition Zposition Zrotation
    JOINT Spine
    {
        OFFSET -0.000000 8.814246 -2.080107
        CHANNELS 3 Zrotation Yrotation Xrotation
        JOINT Spine1
        {
            OFFSET -0.000000 8.844141 0.000000
            CHANNELS 3 Zrotation Yrotation Xrotation
            JOINT Spine2
        }
    }
}

```

File BVH (Biovision Hierarchy) là một định dạng file dữ liệu chứa thông tin về cấu trúc xương và dữ liệu về chuyển động của xương trong một hệ thống xương. File BVH bao gồm hai phần chính: phần khai báo cấu trúc xương và phần dữ liệu chuyển động của xương.

$\text{rotation}_i^{\text{local}} = \{\alpha, \beta, \gamma\}$  lần lượt là góc quay quanh các trục  $Z, Y$ , và  $X$ , góc quay tổng hợp ba trục trong không gian Eule là

$$R = R_Z(\alpha)R_Y(\beta)R_X(\gamma) \quad (1)$$

Trong đó:

1. Ma trận quay quanh trục  $Z$ :

$$R_Z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Ma trận quay quanh trục  $Y$ :

$$R_Y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

3. Ma trận quay quanh trục  $X$ :

$$R_X(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix}$$

Để tính toạ độ chuyển động của một nhân vật, ta thực hiện phép toán sau:

$$\text{position}_{\text{global}} = R \cdot \text{position}_{\text{local}} + \mathbf{t} \quad (2)$$

### B.3 Quá trình chuyển góc quay Euler sang Quaternion

Để tránh Gimbal lock, ta phải chuyển dữ liệu ở dạng góc quay Euler sang dạng góc quay Quaternion. Trong đó góc quay từng Bone dạng Euler ZYX sang dạng Quaternion, mỗi Bone biểu diễn bằng 4 phần tử  $q = (q_w, q_x, q_y, q_z)$ , với các giá trị được tính như sau:

Đầu tiên, chúng ta tính các giá trị cos và sin của một nửa góc quay cho mỗi trục:

- $c_\alpha = \cos\left(\frac{\alpha}{2}\right), \quad s_\alpha = \sin\left(\frac{\alpha}{2}\right)$
- $c_\beta = \cos\left(\frac{\beta}{2}\right), \quad s_\beta = \sin\left(\frac{\beta}{2}\right)$
- $c_\gamma = \cos\left(\frac{\gamma}{2}\right), \quad s_\gamma = \sin\left(\frac{\gamma}{2}\right)$

Dựa trên các giá trị tính được ở trên, các thành phần của quaternion được tính như sau:

- $q_w = c_\alpha c_\beta c_\gamma + s_\alpha s_\beta s_\gamma$

$$\bullet \quad q_x = c_\alpha c_\beta s_\gamma - s_\alpha s_\beta c_\gamma$$

$$\bullet \quad q_y = c_\alpha s_\beta c_\gamma + s_\alpha c_\beta s_\gamma$$

$$\bullet \quad q_z = s_\alpha c_\beta c_\gamma - c_\alpha s_\beta s_\gamma$$

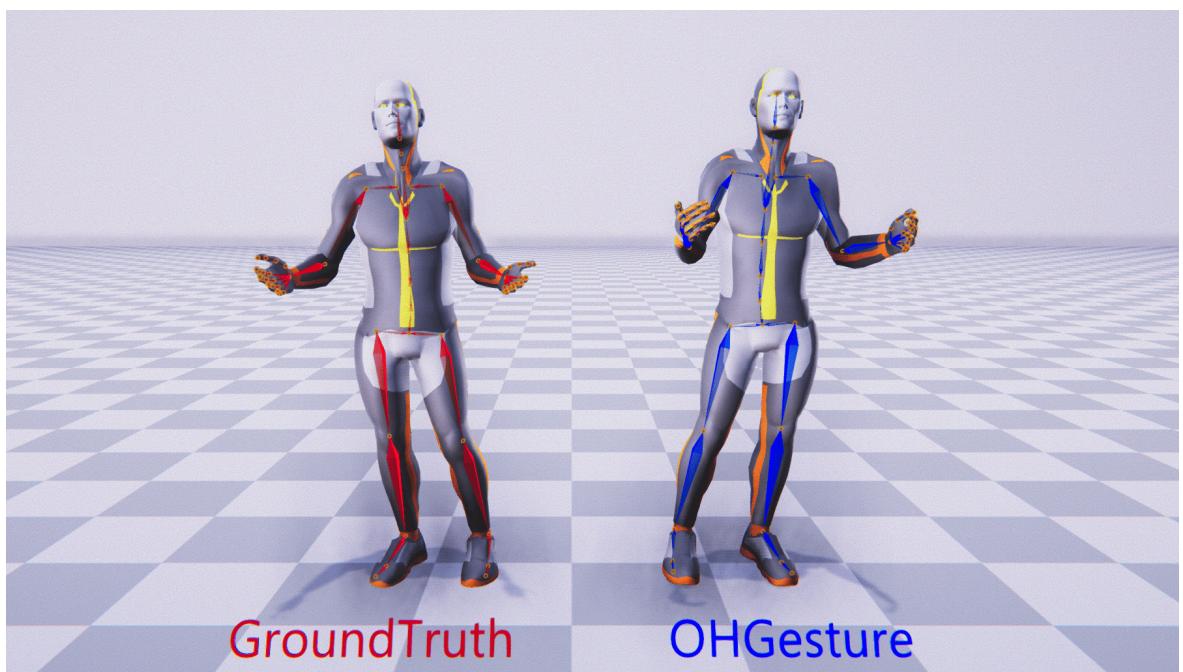
Với quaternion  $q$  đã tính, vị trí toàn cục của bone  $p_{\text{global}}$  được xác định bằng cách quay vị trí cục bộ  $p_{\text{local}}$  thông qua công thức:

$$p_{\text{global}} = q \cdot p_{\text{local}} \cdot q^{-1} + t \quad (3)$$

với  $t$  là vị trí gốc của bone trong không gian toàn cục.

## Phụ lục C. MINH HOẠ KẾT QUẢ SUY LUẬN CỦA CỦ CHỈ

### C.1 So sánh giữa cử chỉ groundtruth và cử chỉ dự đoán



Kết quả cử chỉ ground truth và cử chỉ dự đoán của mô hình ở frame 3821 dự đoán trên mẫu âm thanh 003\_Neutral\_2\_x\_1\_0

## C.2 Minh họa việc sinh cử chỉ với âm thanh nằm ngoài tập huấn luyện



Click vào ảnh để xem video

Minh họa sinh cử chỉ tương ứng với âm thanh của Steven Job

## C.3 Minh họa chuyển động của nhân vật



Click vào ảnh để xem video

Minh họa cử chỉ được lấy 6 khung hình phía trước và 6 khung hình phía sau của cử chỉ.