

OpenHuman: Sinh cử chỉ dựa trên âm thanh

The synthesis system for conversational gestures based on emotion and semantics

Thanh Hoang-Minh

Giảng viên hướng dẫn:
TS. Lý Quốc Ngọc



Faculty of Information Technology
VNUHCM-University of Science

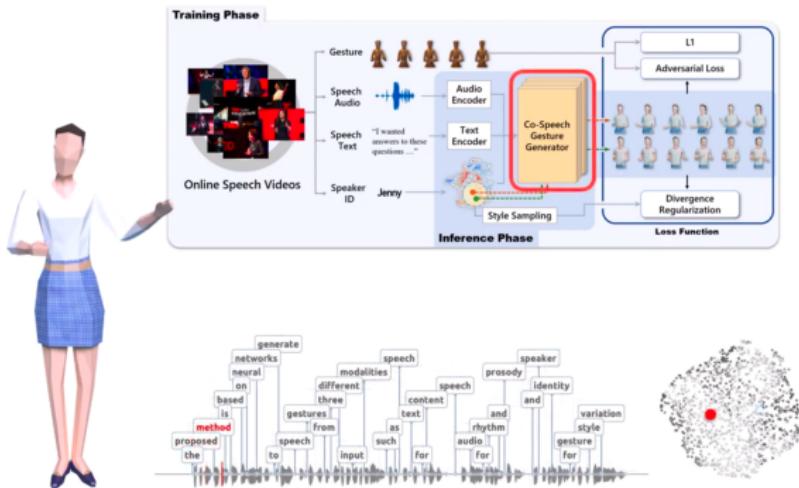
November 12, 2024

Outline

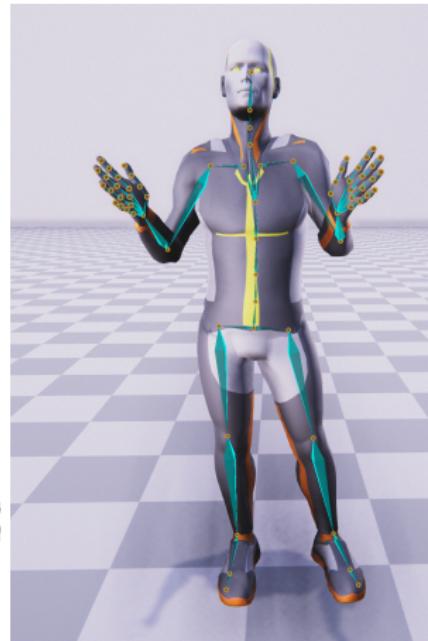
- ① Minh họa
- ② Giới thiệu
- ③ Công trình liên quan
- ④ Mô hình sinh cử chỉ
- ⑤ Thực nghiệm
- ⑥ Kết luận

Minh họa quá trình sinh cử chỉ

- Sinh cử chỉ (Gesture Generation) là gì?



ACM CCS: • Human-centered computing → Human computer interaction (HCI).



Demo Gesture
Generation: [Youtube](#)

Outline

- ① Minh họa
- ② Giới thiệu
- ③ Công trình liên quan
- ④ Mô hình sinh cử chỉ
- ⑤ Thực nghiệm
- ⑥ Kết luận

Động lực nghiên cứu

Text-base Deep Learning

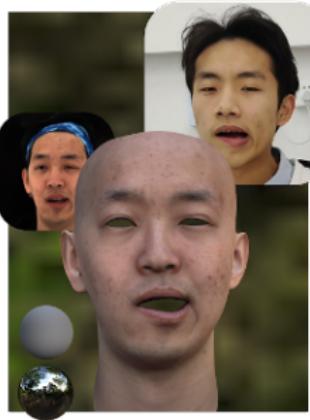
- ChatGPT, Alexa, Character.AI,..
- Text to speech, text to text,..

Text/audio to Realistic Digital Human

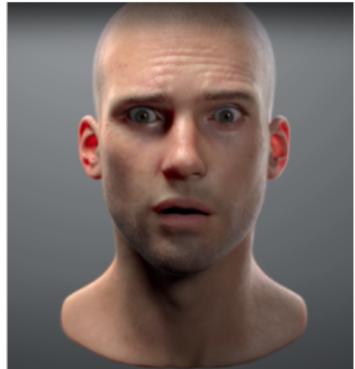
- Video-base (HeyGen, Midjourney,...)
- Rendering-base:
 - Character model (Gaussian Splatting)
 - Character animation: (3D keypoints)



(a)



(b)



Minh họa người kỹ
thuật số siêu thật
(realistic digital human)

Dữ liệu bài toán

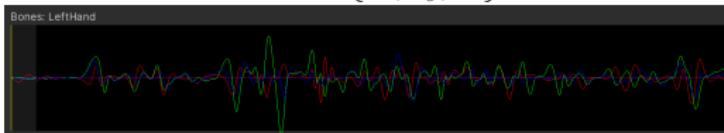
Dữ liệu

- Dữ liệu chuyển động thu nhận từ cảm biến.
- Âm thanh tương ứng với cử chỉ.

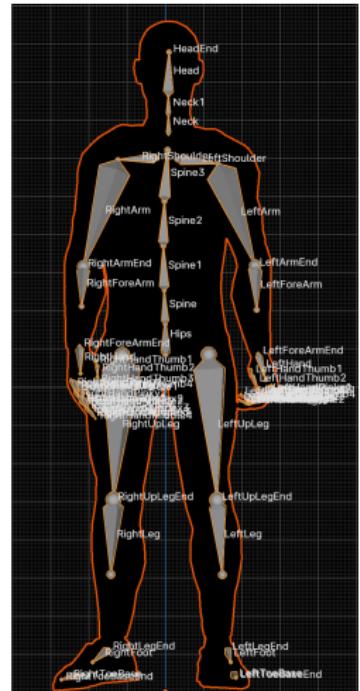
Kiến trúc khung xương

- Chuỗi chuyển động khung xương 60fps
 - Skeleton: 75 bones: $B = \{b_i\}^{75}$
(Head, Spine, Hips, LeftArm,RightArm...)

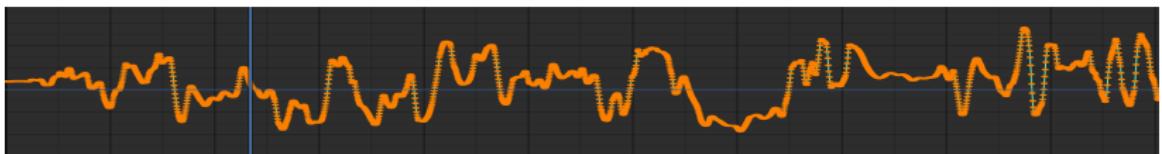
$$\text{Dữ liệu BVH: } b_i = \{r_x, r_y, r_z\}$$



- Dữ liệu sau xử lý: $\mathbf{g} \in \mathbb{R}^D$, với $D = 1141$
$$\mathbf{g} = [\mathbf{p}, \mathbf{r}, \mathbf{p}', \mathbf{r}', \mathbf{p}_{\text{joins}}, \mathbf{r}_{\text{joins}}, \mathbf{p}'_{\text{joins}}, \mathbf{r}'_{\text{joins}}, \mathbf{d}_{\text{gaze}}]$$



Chuyển động của tọa độ

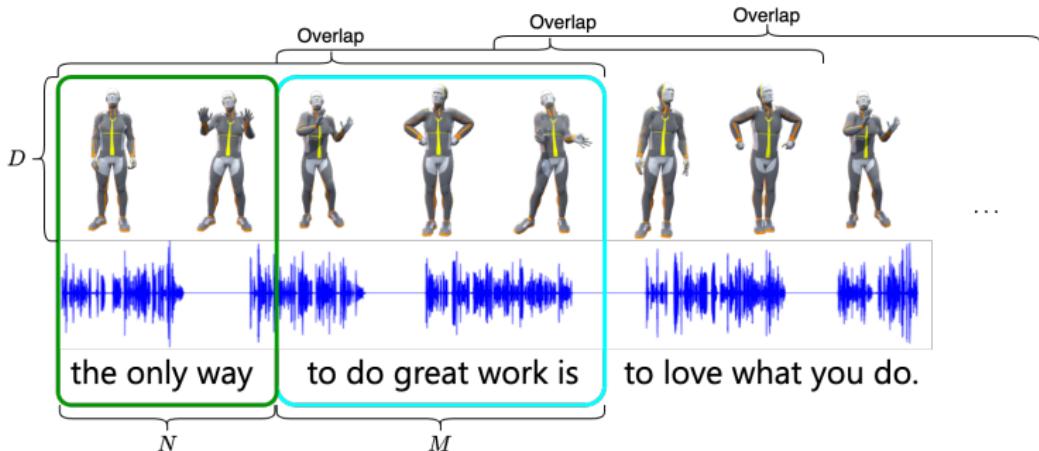


Minh họa chuyển động của góc quay một Bone trong tọa độ

- $\mathbf{p} \in \mathbb{R}^3$: Tọa độ của điểm gốc
- $\mathbf{r} \in \mathbb{R}^4$: 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
- $\mathbf{r}_{\text{joins}} \in \mathbb{R}^{3n_{\text{join}}}$: Góc quay của các khung xương
- $\mathbf{p}'_{\text{joins}} \in \mathbb{R}^{3n_{\text{join}}}$: 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

image here

Phát biểu bài toán



Input

- Chuỗi cử chỉ khởi tạo: $s \in \mathbb{R}^{N \times D}$
- Chuỗi âm thanh: a^{length} sample rate 16000 được cắt thành từng đoạn 4s: $a \in \mathbb{R}^{6400}$
- Cảm xúc: $e \in \mathbb{R}^6$ (Happy, Sad, Neutral, Angry, Old, Funny)

Output

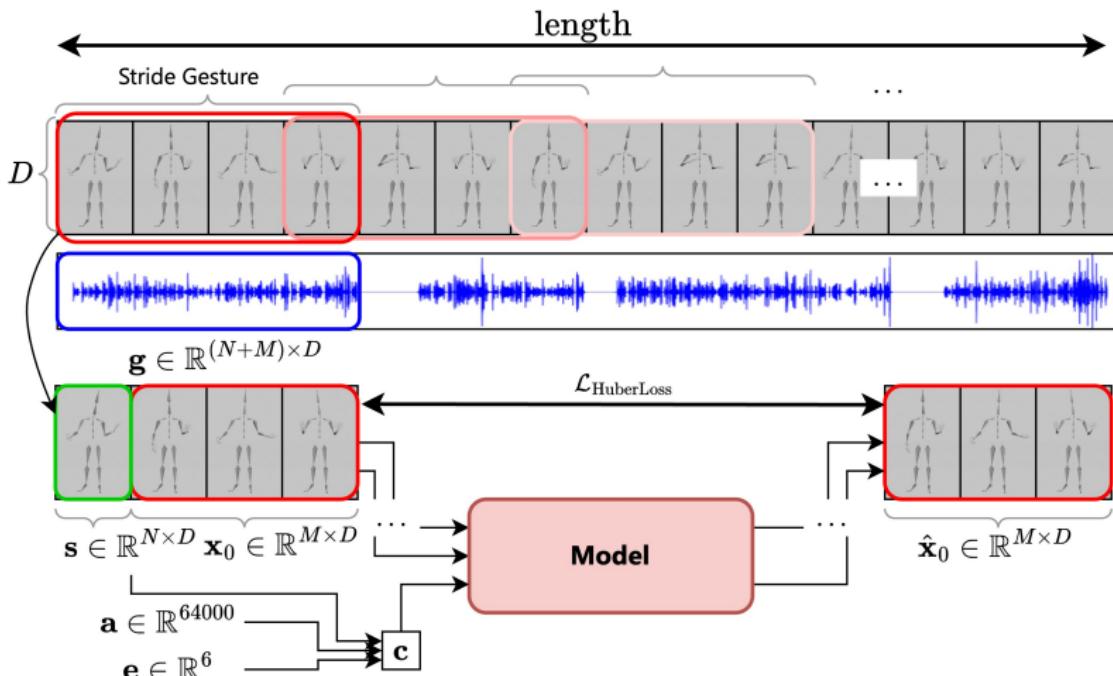
- Chuỗi cử chỉ dự đoán: $\hat{x} \in \mathbb{R}^{M \times D}$

Growth Truth

- Chuỗi cử chỉ gốc: $x \in \mathbb{R}^{M \times D}$

Khung chương trình

Sử dụng mô hình Diffusion Classifier-Free Guidance (có điều kiện) với
cứ chỉ $\mathbf{x}^{1:M \times D}$ làm x_0 và điều kiện $c = [\mathbf{s}, \mathbf{a}, \mathbf{e}]$. $N = 8$,
 $M = 80 \sim 4$ giây



Outline

- ① Minh họa
- ② Giới thiệu
- ③ Công trình liên quan
- ④ Mô hình sinh cử chỉ
- ⑤ Thực nghiệm
- ⑥ Kết luận

Các phương pháp

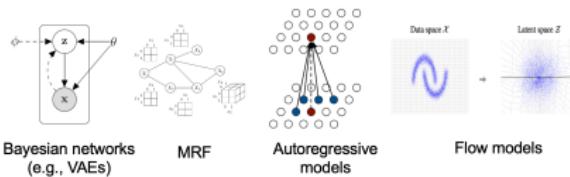
Rule Base & Statistic

- BEAT, Rule-based generation [1]
- Gesture Controllers [2]

Deep Learning

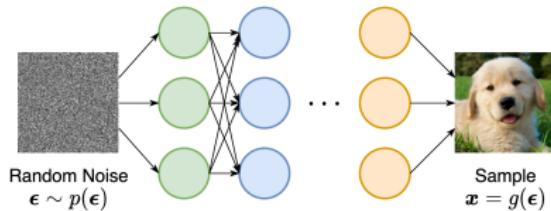
Likelihood-based models:

- **MLP**: Gesticulator
- **RNN**: Speech2AffectiveGestures, HA2G, TransGesture, ..
- **Normalising Flows**:
Text2gestures, Speech2Gesture
VAE/VQ-VAE: Audio2Gestures;



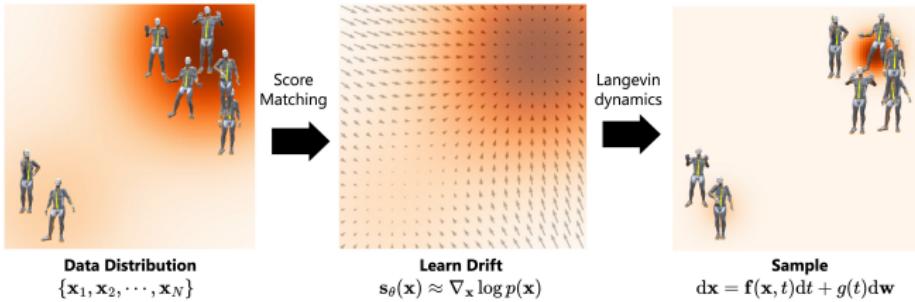
Implicit generative models:

- **GAN**: DiffGAN
- **Diffusion**: Listen denoise action [3], DiffuseStyleGesture [4], Taming Diffusion [5].
- Reinforcement: Contrastive Pre-trained Rewards [6]



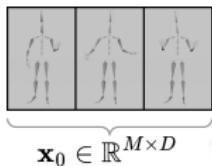
Tư tưởng cơ bản của Diffusion

Dataset $\mathcal{G} = \{\mathbf{x}_i^{M \times D}\}_1^n$, ta chuẩn hoá dữ liệu $\mathbf{x}_i^{M \times D} = \frac{\mathbf{x}_i^{M \times D} - \mu}{\sigma}$



Growth Truth Distribution

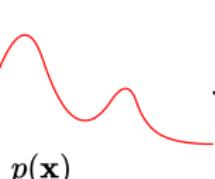
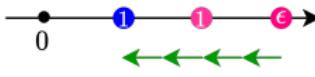
$$\mathbf{x}_0 = q(\mathbf{x})$$



$$\mathbf{x}_0 \in \mathbb{R}^{M \times D}$$

Training (Denoise)

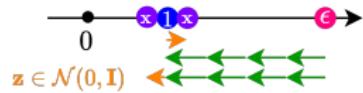
$$\text{Drift } \mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$$



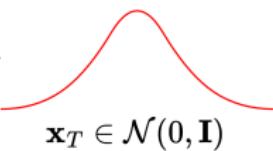
$$f_\theta$$

Sampling (Denoise)

$$\mathbf{x}_{\text{sample}} = \mathbf{s}_\theta + \sigma_t \cdot \mathbf{z}$$



$$\begin{matrix} 1 \rightarrow T \\ \xrightarrow{\text{Diffuse}} \end{matrix}$$



$$\mathbf{x}_T \in \mathcal{N}(0, \mathbf{I})$$

Công trình liên quan

Ký hiệu

Tham số: Đang huấn luyện θ , đã huấn luyện xong: θ' , $\hat{\mathbf{x}}$: dự đoán

Phân phối chuẩn

$$\mathcal{N}(\textcolor{red}{a}\mathbf{x}, \textcolor{blue}{b}^2)$$

- Một hàm $f(x) = ax + b\epsilon$ với $\epsilon \in \mathcal{N}(0, 1)$ được ký hiệu là $f(x) \sim \mathcal{N}(ax, b^2)$
- Trung bình: $\mu = \textcolor{red}{a}x = \frac{1}{n} \sum_{i=1}^n x_i$
- Phương sai: $\sigma^2 = \textcolor{blue}{b}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2$

Xác xuất có điều kiện

$$p(\textcolor{green}{x} | \textcolor{orange}{y})$$

- $p(\textcolor{green}{x} | \textcolor{orange}{y})$ là xác xuất có điều kiện.
- $\textcolor{orange}{y}$: xảy ra trước (bên phải)
- $\textcolor{green}{x}$: xảy ra sau y (bên trái)

Đặc điểm của việc học dữ liệu Motion

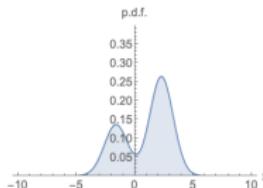
Quan hệ giữa dữ liệu cử chỉ và âm thanh:

- Một đoạn cử có thể bao gồm nhiều âm thanh.
- Mỗi âm thanh có thể tương ứng với nhiều đoạn cử chỉ khác nhau.

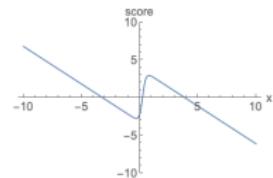
Khó khăn

- Dữ liệu ít, chi phí cao
- Thiếu nhãn và dữ liệu tương ứng giữa âm thanh, cử chỉ.
- Quá trình sinh có thể dễ điều khiển

liệu



Phải chuẩn hoá (diện tích dưới đường cong phải tích phân thành một)

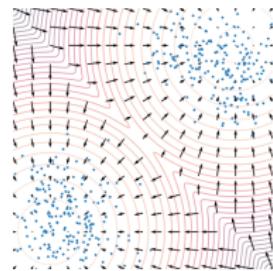


Không cần chuẩn hoá.

$p(\mathbf{x})$
probability density



$\nabla_{\mathbf{x}} \log p(\mathbf{x})$
score function



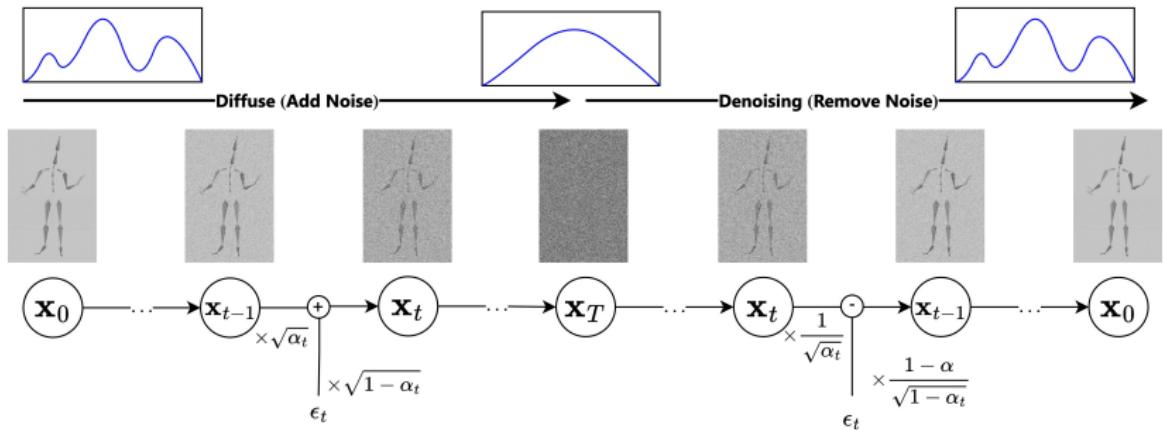
score function vs
probability density

Diffuse và Denoise thông thường

Diffuse: Cho $\{\alpha_t \in (0, 1)\}_{t=1}^T$ và $\alpha_1 > \alpha_2 > \dots > \alpha_T$.

Với nhiễu: $\epsilon_0, \epsilon_1, \dots, \epsilon_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\epsilon_i \neq \epsilon_j$ ($\forall i, j \in T$), nhiễu ϵ_t cố định.

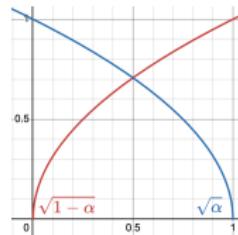
$$\mathbf{x}_t = \sqrt{\alpha_t} \cdot \mathbf{x}_{t-1} + \sqrt{1 - \alpha_t} \cdot \epsilon \quad (1)$$



Denoise

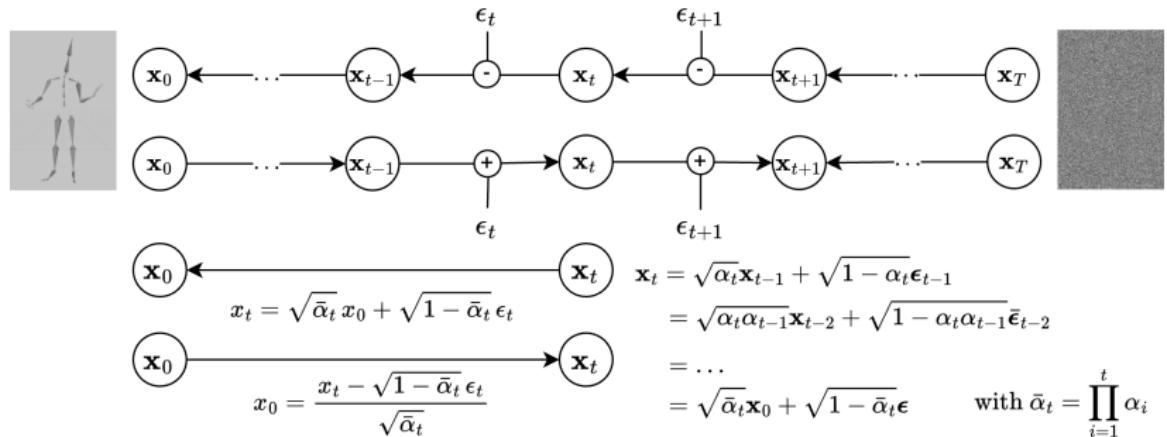
$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_t - \sqrt{1 - \alpha_t} \cdot \epsilon) \quad (2)$$

Công trình liên quan



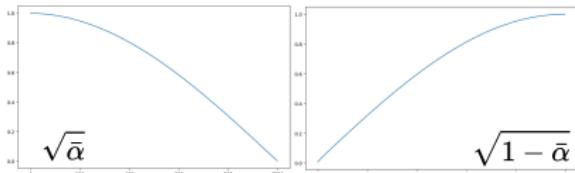
Quan hệ \mathbf{x}_0 , \mathbf{x}_t và \mathbf{x}_{t-1}

Ta có thể suy ra \mathbf{x}_t từ \mathbf{x}_0 và ngược lại. Với $\epsilon_{t-1}, \epsilon_{t-2}, \dots \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$



- $\bar{\alpha}_1 > \dots > \bar{\alpha}_T$
- Tổng hai nhiễu cũng là nhiễu:

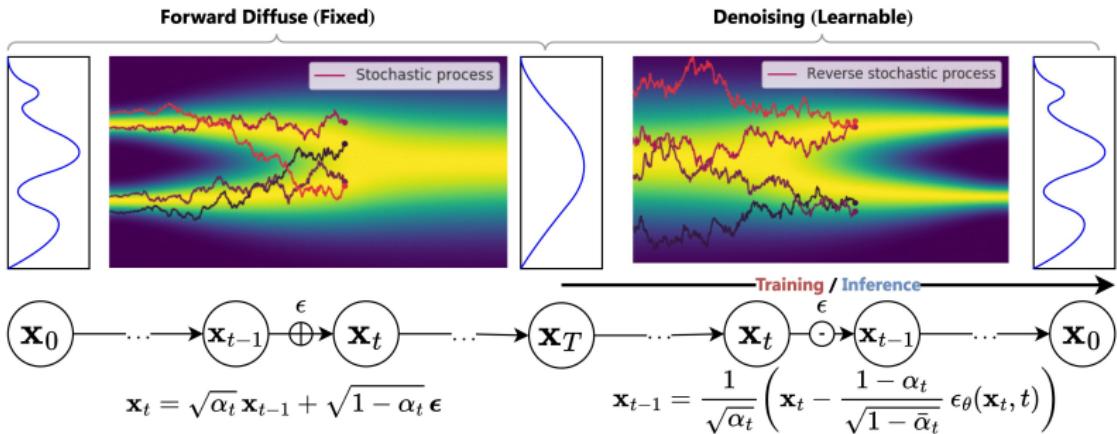
$$\mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I}) + \mathcal{N}(\mathbf{0}, \sigma_2^2 \mathbf{I}) = \mathcal{N}(\mathbf{0}, (\sigma_1^2 + \sigma_2^2) \mathbf{I})$$



$q(\mathbf{x}_t | \mathbf{x}_{t-1})$ và $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ trong DDPM

DDPM (Denoising Diffusion Probabilistic Model [7])

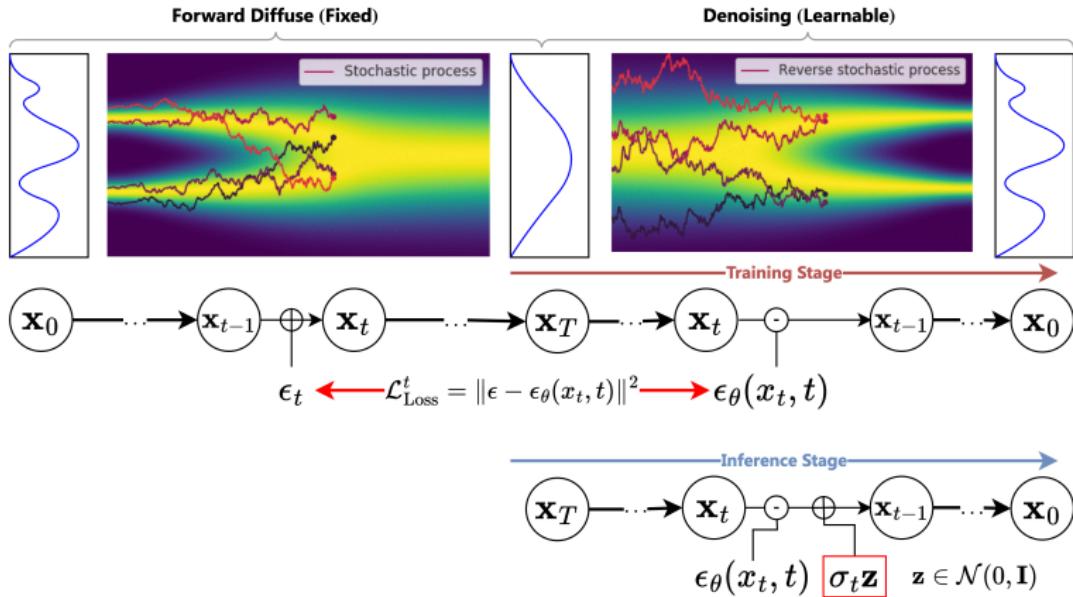
$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t} \mathbf{x}_{t-1}, (1 - \alpha_t) \mathbf{I}) \rightarrow p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \rightarrow$$



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

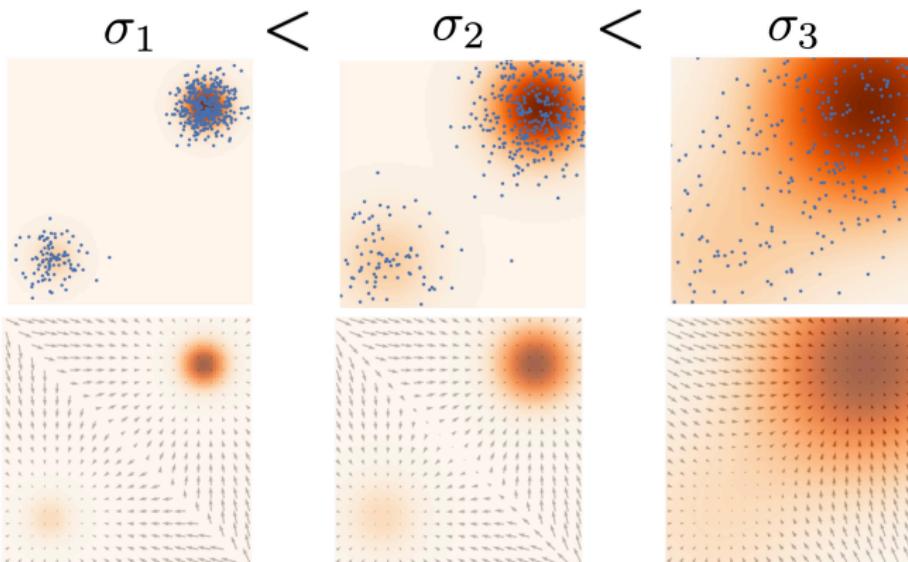
Vanilla Diffusion với ϵ Objective

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



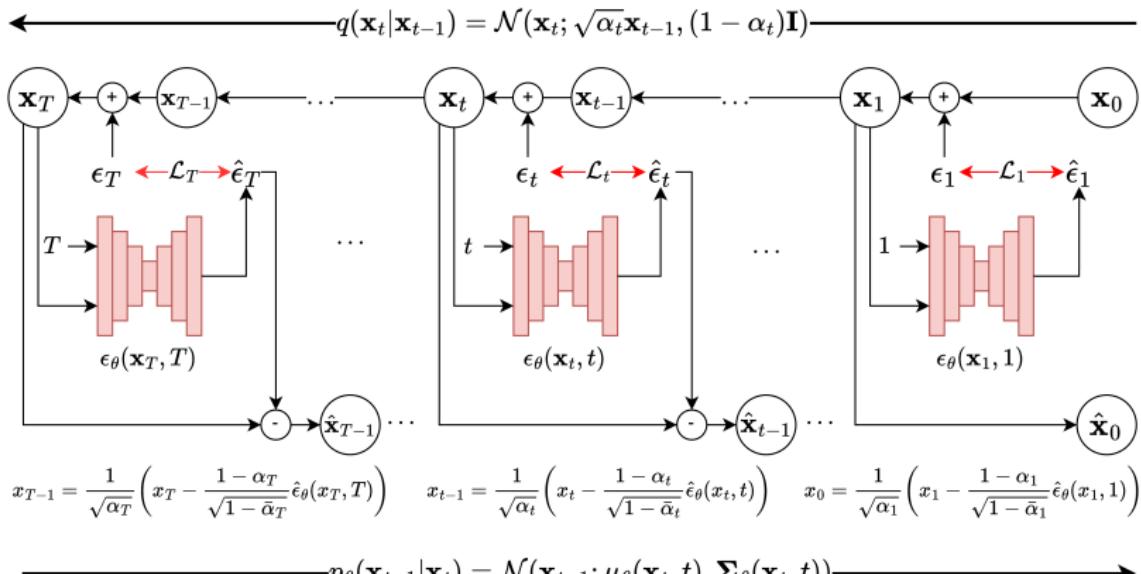
Điều khiển σ_t bằng với Langevin dynamics

Trong quá trình Denoise ($T \rightarrow 0$) $\sigma_T > \dots > \sigma_2 > \sigma_1$, ta sẽ giảm σ_t để giảm dần nhiễu, để mô hình có thể hội tụ ở những vùng có mật độ xác xuất cao.



Training DDPM

Hàm loss: $\mathcal{L} = \sum_{t=1}^T \mathcal{L}_t$. $\mathcal{L}_t = \mathbb{E}_{\mathbf{x}_0, \epsilon_t \sim \mathcal{N}(0, I), t} [\|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\|^2]$



Các bước huấn luyện với DDPM

- ❶ Tính sẵn các giá trị $\sqrt{\alpha_t}$, $\sqrt{1 - \alpha_t}$ và $\sqrt{\bar{\alpha}_t}$ ở mọi bước $t : 1 \rightarrow T$.
 $\{\alpha_t \in (0, 1)\}_{t=1}^T$, $\alpha_1 < \alpha_2 < \dots < \alpha_T$
- ❷ Lấy nhãn \mathbf{x}_0 từ phân bố của dữ liệu đã chuẩn hoá
- ❸ Random nhiễu ϵ_t ở mọi bước $t : 1 \rightarrow T$, với $\forall t : \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- ❹ Gây nhiễu (forward) \mathbf{x}_0 để thu được \mathbf{x}_t ở mọi bước $t : 1 \rightarrow T$

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

- ❺ for all t , lấy t **ngẫu nhiên** $t \sim [1, T]$
- ❻ Cho \mathbf{x}_t và t vào mô hình để dự đoán nhiễu $\hat{\epsilon} = \epsilon_\theta(\mathbf{x}_t, t)$
- ❼ Đạo hàm để cập nhật trọng số $\nabla_{\theta_t} \|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\|^2$

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

- ❽ Quay lại bước 6 cho đến khi hội tụ để thu được θ'

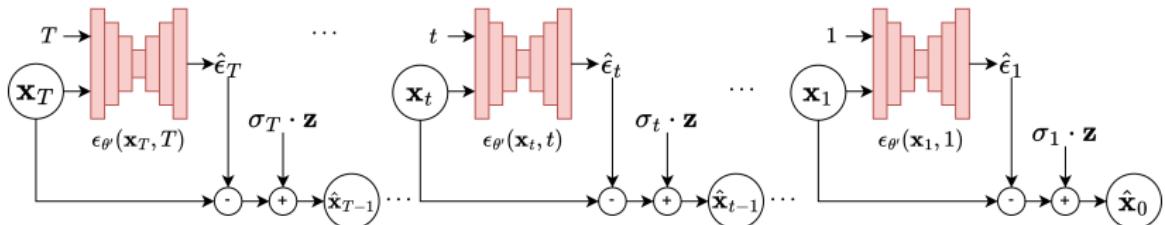
Sampling DDPM

Hàm sampling:

- $\mathbf{x}_T \in \mathcal{N}(0, \mathbf{I})$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \hat{\epsilon}_{\theta'}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$$

- Diffusion: $\mathcal{L}_{\text{loss}} = \mathbb{E}_{\mathbf{x}_0, \epsilon_t \sim \mathcal{N}(0, I), t} [\|\epsilon_t - \epsilon_{\theta}(\mathbf{x}_t, t)\|^2]$



$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \hat{\epsilon}_{\theta'}(x_t, t) \right) + \sigma_t \mathbf{z}$$

$$x_0 = \frac{1}{\sqrt{\alpha_1}} \left(x_1 - \frac{1 - \alpha_1}{\sqrt{1 - \bar{\alpha}_1}} \hat{\epsilon}_{\theta'}(x_1, 1) \right) + \sigma_1 \mathbf{z}$$

$$p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t, t), \Sigma_{\theta}(\mathbf{x}_t, t)) \longrightarrow$$

Các bước lấy mẫu với 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 $\sqrt{\bar{\alpha}_t}$ có được từ bước huấn luyện
- ③ Tính hệ số điều chỉnh nhiễu σ_t từ α_t ở mọi bước $t: 1 \rightarrow T$
$$\sigma_t = \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}} (1 - \alpha_t)$$
- ④ for all t , lấy t **tuần tự** $t \sim [T, \dots, 1]$
- ⑤ Random nhiễu $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
- ⑥ Đưa \mathbf{x}_t vào 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

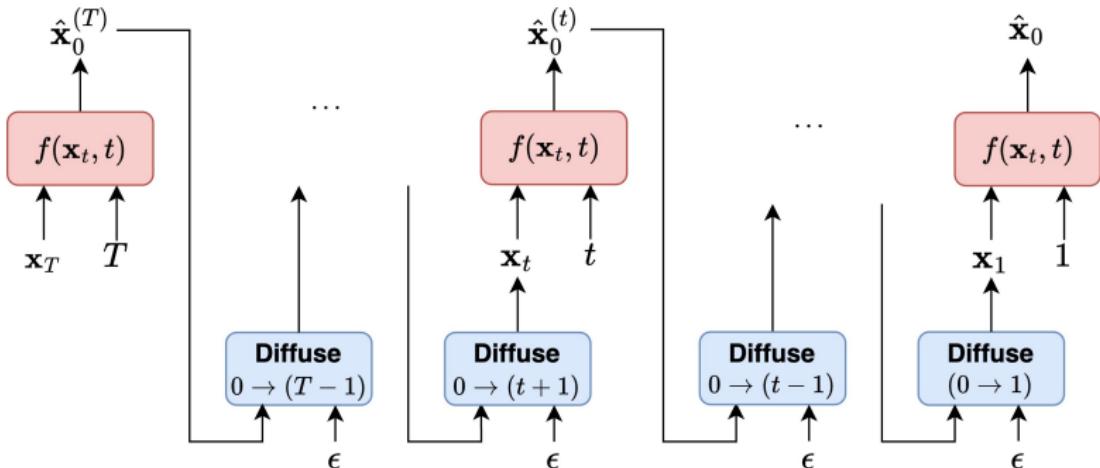
$$\mu = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta'}(\mathbf{x}_t, t) \right)$$

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

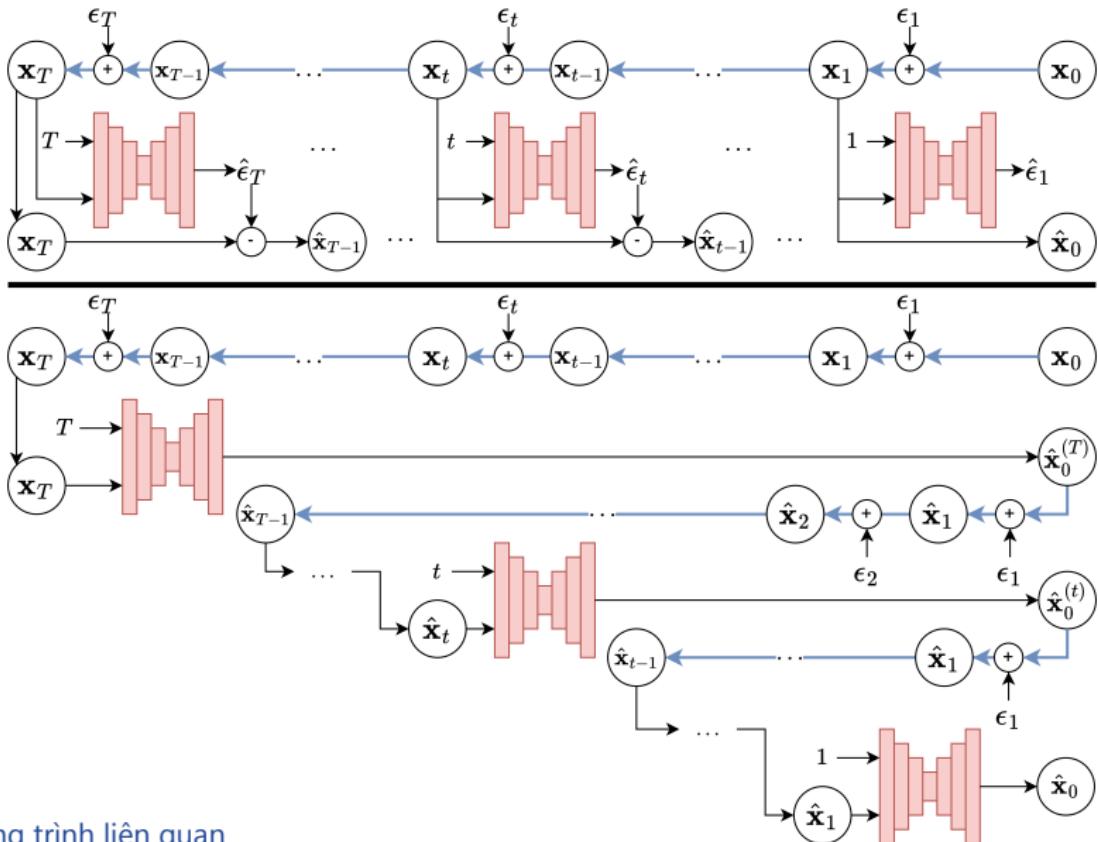
Cải tiến của với x_0 Objective (DALL E-2)

Những điểm cải tiến của x_0 so với ϵ objective

- Thay vì hàm $f_\theta(\mathbf{x}_t, t)$ dự đoán nhiễu ϵ_t thì $f_\theta(\mathbf{x}_t, t)$ dự đoán \mathbf{x}_0 .
- Sau khi có \mathbf{x}_0 thì ta thêm nhiễu ϵ đã có từ trước (từ forward process) để được $\hat{\mathbf{x}}_{t-1}$
- Tiếp tục cho đến khi được $\hat{\mathbf{x}}_0$



So sánh ϵ objective và x_0 objective

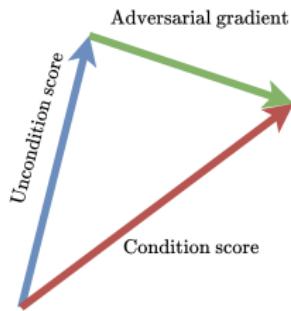


Công trình liên quan

Classifier-Free Guidance

Để có thể học được có điều kiện (condition c).

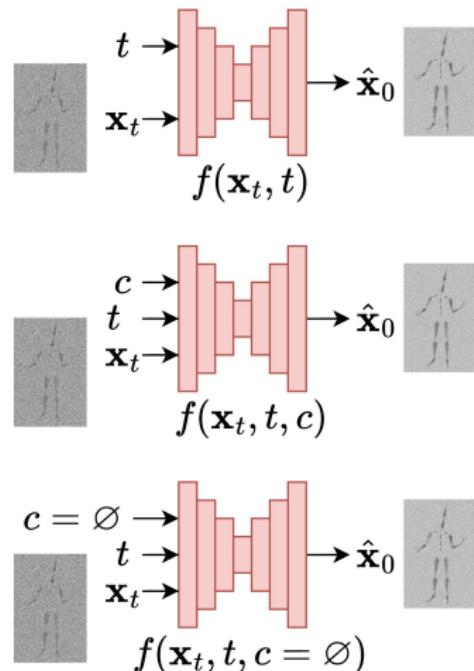
Với mỗi bước $t : T \rightarrow 1$,



$$\begin{aligned} \underbrace{\nabla \log p_\gamma(\mathbf{x}_t | \mathbf{y})}_{\text{Conditional score}} &= \underbrace{\nabla \log p(\mathbf{x}_t)}_{\text{Uncondition score}} + \underbrace{\gamma \nabla \log p(\mathbf{y} | \mathbf{x}_t)}_{\text{Adversarial gradient}} \\ &= \underbrace{\nabla \log p(\mathbf{x}_t)}_{\text{Uncondition score}} + \underbrace{\gamma (\nabla \log p(\mathbf{x}_t | \mathbf{y}) - \nabla \log p(\mathbf{x}_t))}_{\text{Adversarial gradient}} \\ &= \underbrace{(1 - \gamma) \nabla \log p(\mathbf{x}_t)}_{\text{Uncondition score}} + \underbrace{\gamma \nabla \log p(\mathbf{x}_t | \mathbf{y})}_{\text{Condition score}} \end{aligned}$$

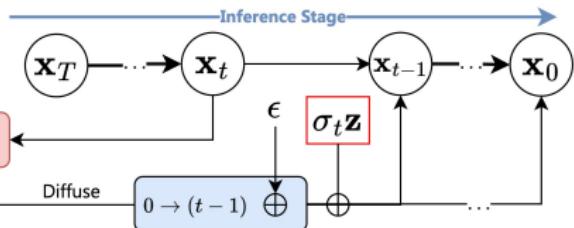
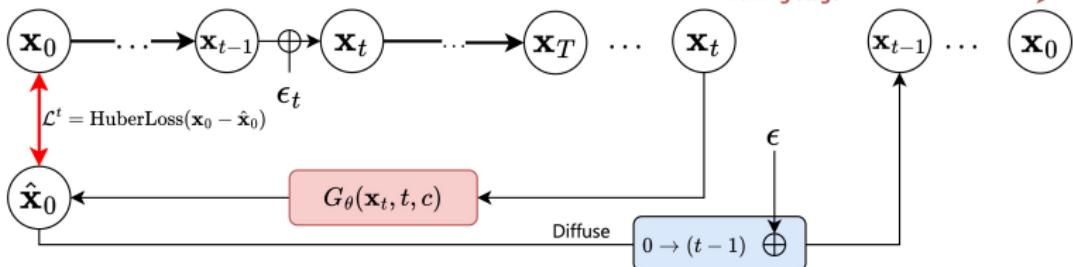
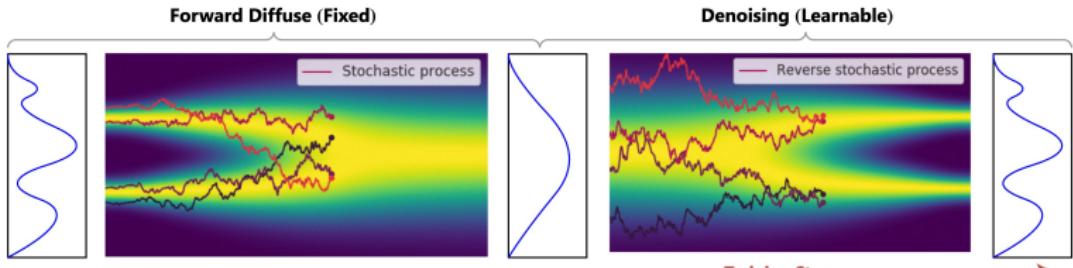
$$\hat{\mathbf{x}}_{0,\gamma,c} = \gamma f(\mathbf{x}_t, t, c) + (1 - \gamma) f(\mathbf{x}_t, t, c_\emptyset)$$

Công trình liên quan



Classifier-Free Guidance với x_0 Objective

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



Công trình liên quan

Outline

- ① Minh họa
- ② Giới thiệu
- ③ Công trình liên quan
- ④ Mô hình sinh cử chỉ
- ⑤ Thực nghiệm
- ⑥ Kết luận

Diffusion cho bài toán sinh cử chỉ

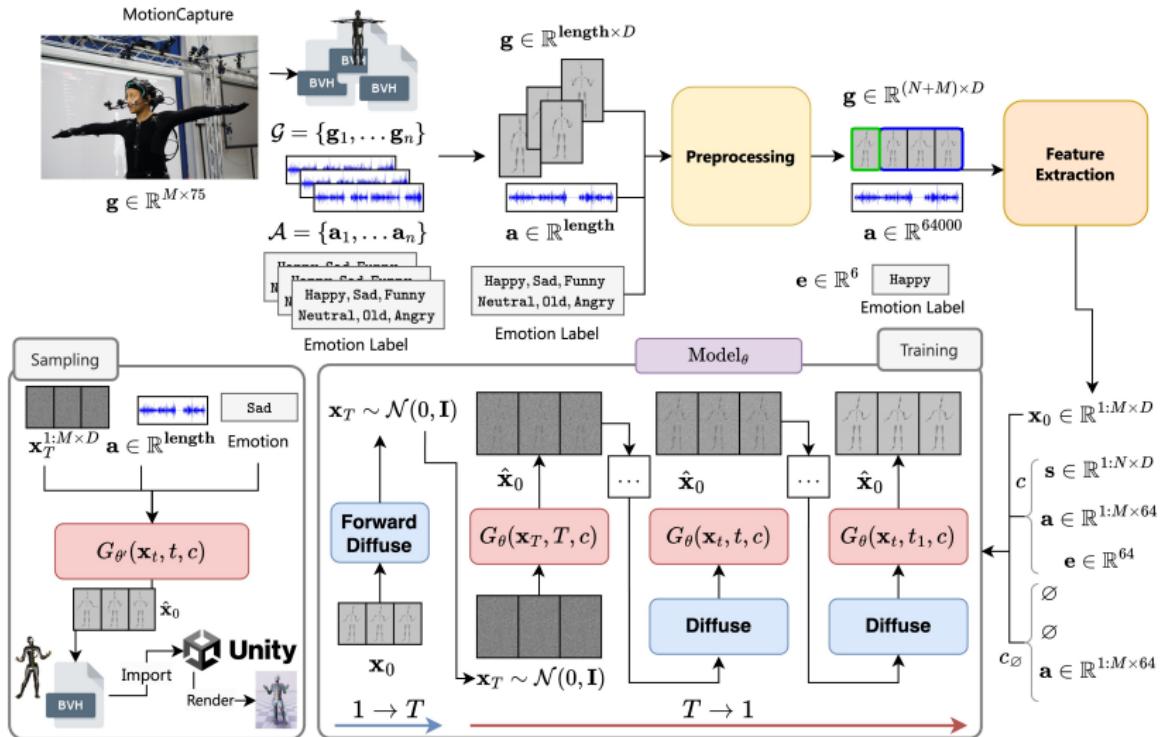
Điểm giống

- Sử dụng mô hình Diffusion [4] 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.

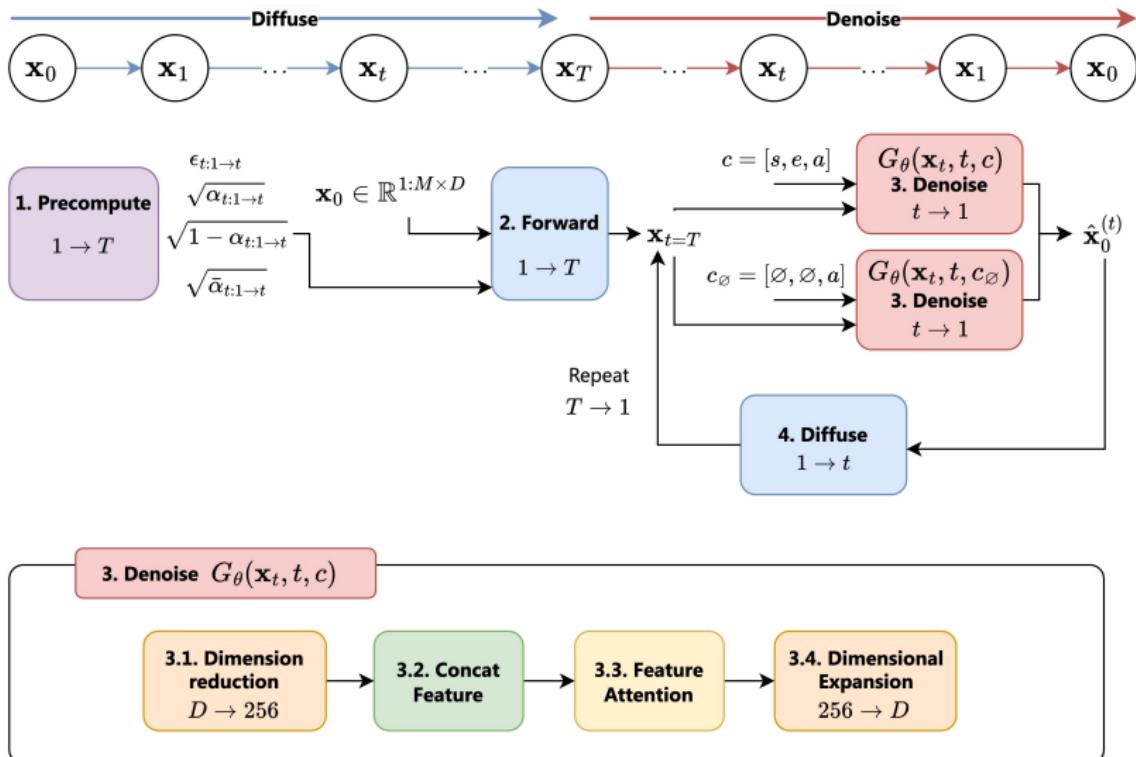
Điểm khác

- Sinh cử chỉ có điều kiện:
 - Điều kiện cảm xúc: $c = [\mathbf{s}, \mathbf{e}, \mathbf{a}]$ và $c_\emptyset = [\emptyset, \emptyset, \mathbf{a}]$.
 - Nội suy trạng thái giữa hai cảm xúc e_1, e_2 , sử dụng điều kiện: $c = [\mathbf{s}, \mathbf{e}_1, \mathbf{a}]$ và $c_\emptyset = [\mathbf{s}, \mathbf{e}_2, \mathbf{a}]$.
- 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 (Giống ControlNet - Pixel-wise Condition)

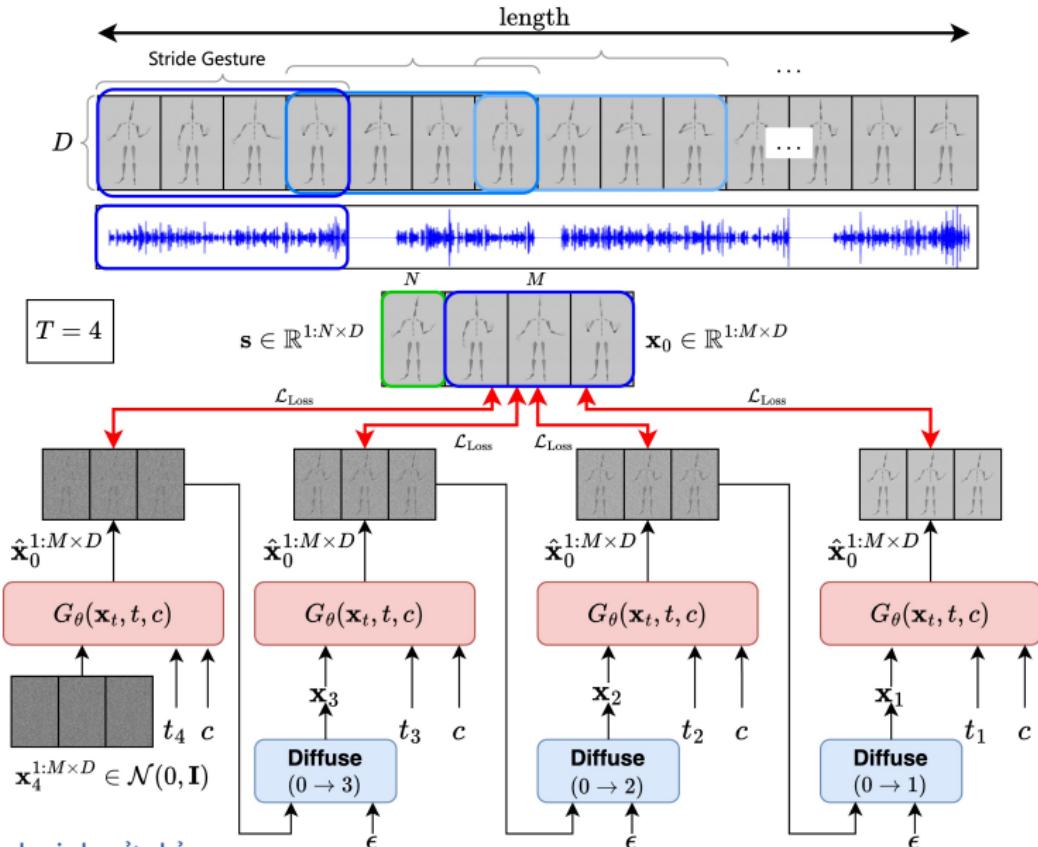
Tổng quan các công đoạn



Các công đoạn trong mô hình OHGesture

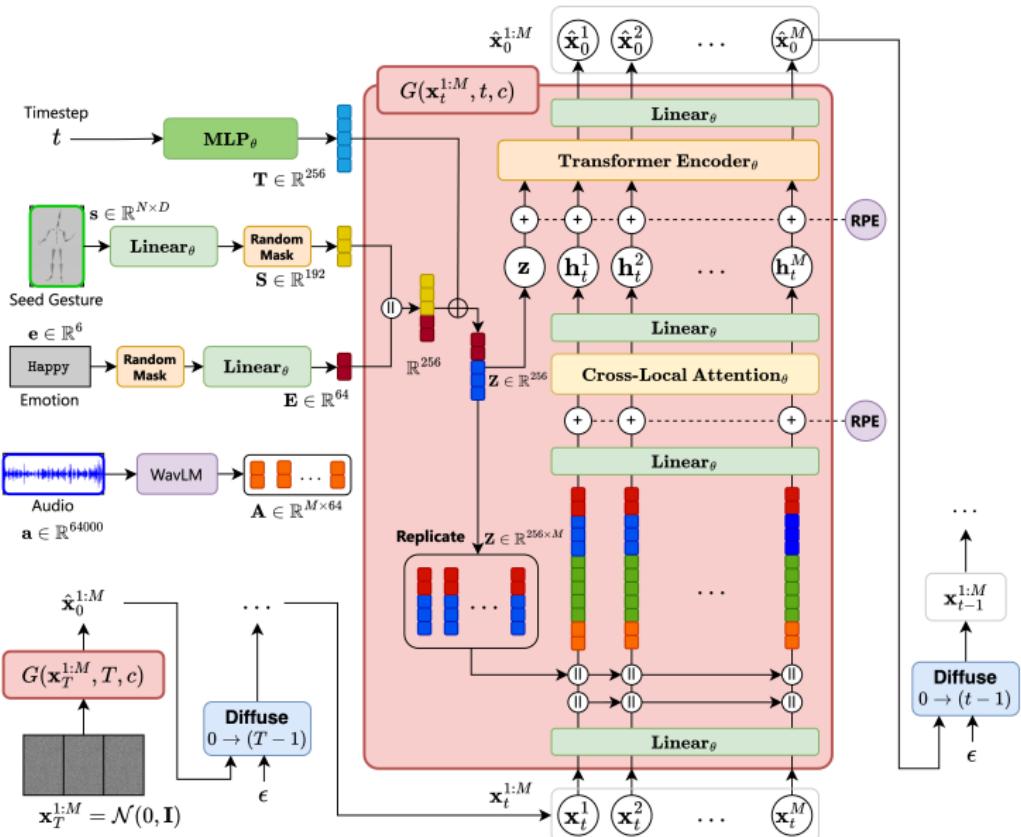


Diffusion cho bài toán sinh cử chỉ



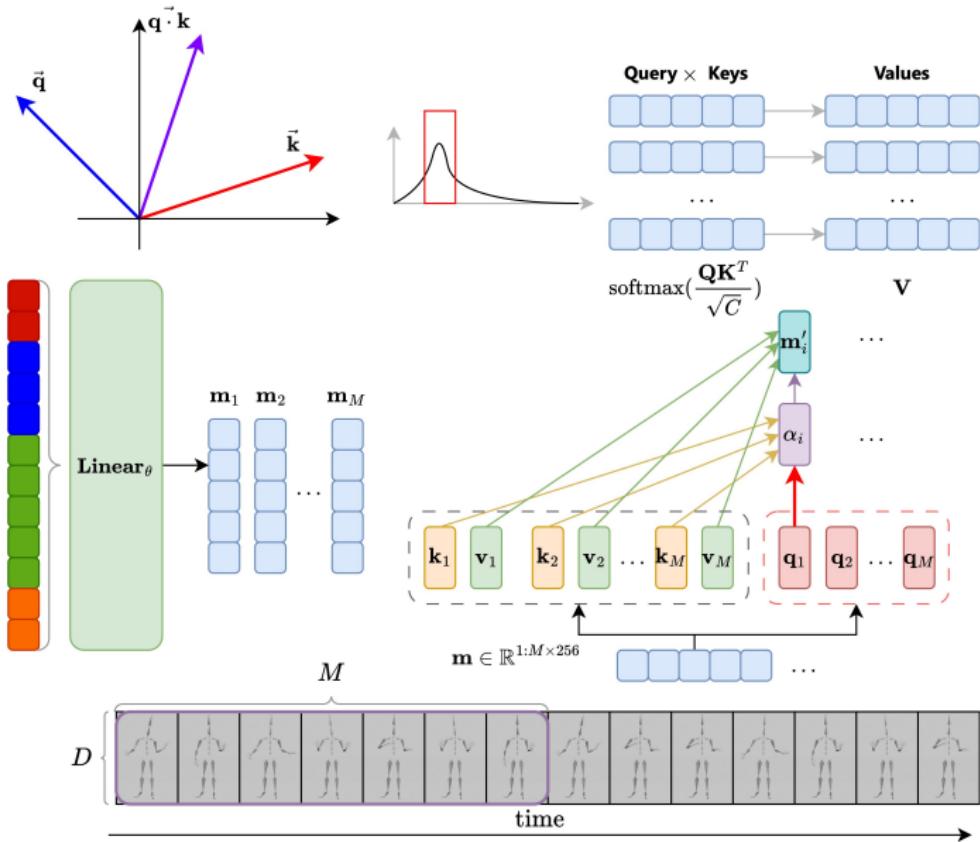
Mô hình sinh cử chỉ

Kiến trúc OHGesture



Mô hình sinh cử chỉ

Attention



Mô hình sinh cử chỉ

Cross-Local Attention and Self-Attention

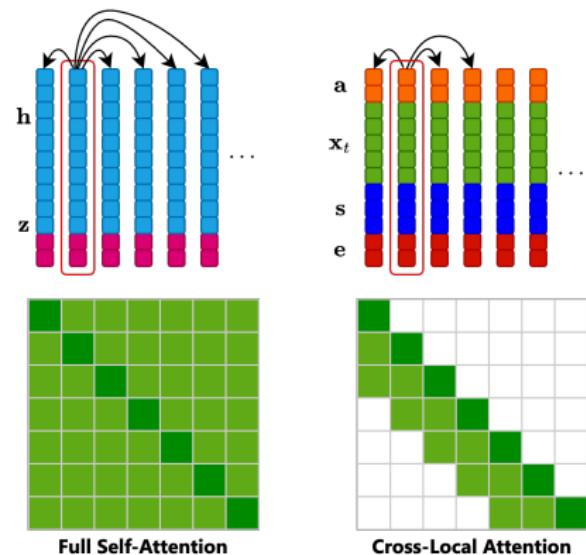
Cross-Local Attention

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

Self-Attention

$$f_{\text{MultiHead}}(\mathbf{x}) = \text{concat} (\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_h) \mathbf{W}_O$$

$$\mathbf{H}_i = \text{softmax} \left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_k}} \right) \mathbf{V}_i$$



Emotion-controllable Gesture Generation

Classifier-free guidance

- Điều khiển cảm xúc (Emotion-controllable) Conditional:

$c_1 = [s, e, a]$, Unconditional: $c_2 = [\emptyset, \emptyset, a]$

- Nội suy cảm xúc (Emotion-interpolating) Unconditional:

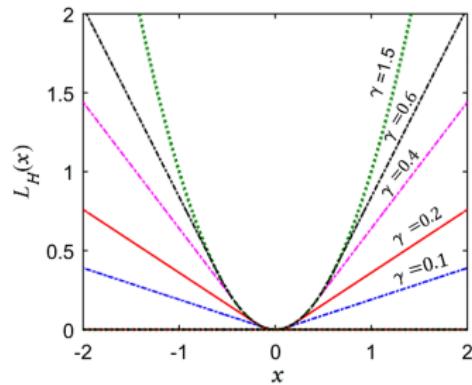
$c_1 = [s, e_1, a]$, Conditional: $c_2 = [s, e_2, a]$

$$\hat{\mathbf{x}}_{0\gamma, c_1, c_2} = \gamma G_\theta(\mathbf{x}_t, t, c_1) + (1 - \gamma) G_\theta(\mathbf{x}_t, t, c_2) \quad (3)$$

Huber Loss

$$\mathcal{L} = E_{\mathbf{x}_0 \sim q(\mathbf{x}_0 | c), t \sim [1, T]} [\text{HuberLoss}(\mathbf{x}_0 - \hat{\mathbf{x}}_0)] \quad (4)$$

- Nếu $|y - f(x)| \leq \delta$: Mượt mà và dễ tối ưu hóa khi các lỗi nhỏ. $\mathcal{L}_\delta(y, f(x)) = \frac{1}{2}(y - f(x))^2$
- Nếu $|y - f(x)| > \delta$: Giảm ảnh hưởng của các giá trị ngoại lai. $\mathcal{L}_\delta(y, f(x)) = \delta \cdot |y - f(x)| - \frac{1}{2}\delta^2$



Các bước huấn luyện với OHGesture

- ① Tính sẵn các giá trị, và các siêu tham số: $\gamma, \sqrt{\alpha_t}, \sqrt{1 - \alpha_t}, \sqrt{\bar{\alpha}_t}$,
Random nhiễu ϵ_t ở mọi bước $t : 1 \rightarrow T$. $\{\alpha_t \in (0, 1)\}_{t=1}^T$
- ② Lấy nhãn \mathbf{x}_0 từ phân bố của dữ liệu đã chuẩn hoá
- ③ Random Bernoulli masks $c_1 = [\mathbf{s}, \mathbf{e}_1, \mathbf{a}], c_2 = [\mathbf{s}, \mathbf{e}_2, \mathbf{a}]$ hoặc
 $c_2 = [\emptyset, \emptyset, \mathbf{a}]$
- ④ Forward \mathbf{x}_0 để có cử chỉ nhiễu $\mathbf{x}_t : \mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_t$
- ⑤ for all t , lấy t **ngẫu nhiên** $t \sim [1, T]$
- ⑥ Cho \mathbf{x}_t và t, c, c_\emptyset để dự đoán chuỗi cử chỉ

$$\hat{\mathbf{x}}_{0\gamma, c_1, c_2} = \gamma G_\theta(\mathbf{x}_t, t, c_1) + (1 - \gamma) G_\theta(\mathbf{x}_t, t, c_2) \quad (5)$$

- ⑦ Tính loss và đạo hàm để cập nhật trọng số θ

$$\mathcal{L}_t = \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \boldsymbol{\epsilon}_t} \left[\text{HuberLoss}(\mathbf{x}_0, \hat{\mathbf{x}}_0) \right] \quad (6)$$

- ⑧ Quay lại bước 6 cho đến khi hội tụ để thu được θ'

Các bước lấy mẫu (Sampling) với OHGesture

- 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 $\sqrt{\bar{\alpha}_t}$ có được từ bước huấn luyện, tính sẵn các giá trị σ_t từ α_t ở mọi bước $t : 1 \rightarrow T$
- ③ Cắt mỗi 4s âm thanh 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 và tạo cặp condition $c = [s, e, a]$
- ④ for all t , lấy t **tuần tự** $t \sim [T, \dots, 1]$
- ⑤ Random nhiễu $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
- ⑥ Đưa \mathbf{x}_t vào để suy luận $\hat{\mathbf{x}}_0^{(t)} = G_{\theta'}(\mathbf{x}_t, t, c)$
- ⑦ Forward $\hat{\mathbf{x}}_0^{(t)}$ $0 \rightarrow t$ để được $\hat{\mathbf{x}}_{t-1}^{(t)}$
- ⑧ Cộng thêm một lượng nhiễu $\hat{\mathbf{x}}_{t-1} = \hat{\mathbf{x}}_{t-1}^{(t)} + \sigma_t \mathbf{z}$
- ⑨ Quay lại bước 4, khi $t = 1$ ta thu được $\hat{\mathbf{x}}_0$ từ quá trình khử nhiễu

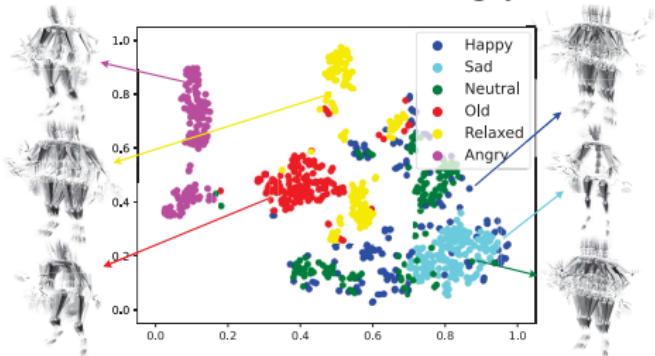
Outline

- ① Minh họa
- ② Giới thiệu
- ③ Công trình liên quan
- ④ Mô hình sinh cử chỉ
- ⑤ Thực nghiệm
- ⑥ Kết luận

Dataset

ZeroEGGs Dataset [8]:

- Bao gồm 67 đoạn độc thoại của diễn viên motion capture nữ
- Độ dài toàn bộ tập dữ liệu là 135 phút.
- Bao gồm 6 cảm xúc: Happy, Sad, Neutral, Old, Relaxed, Angry



Training Process

Toàn bộ quá trình học trong 1 ngày với tham số sau:

- $T = 1000$
- Sử dụng Card Nvidia 3090
- Chia dataset thành tỷ lệ: 8 : 1 : 1 cho lần lượt: tập training, testing và validation.
- Learning rate: 3×10^{-5}
- 384 batch size với 300000 samples.

Mã nguồn chương trình: hmthanh/OHGesture

Độ đo

Mean opinion scores (MOS)

- Human-likeness
- Gesture-Speech Appropriateness
- Gesture-style Appropriateness

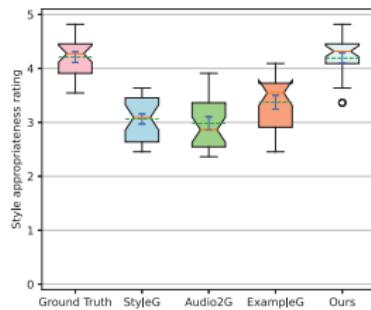
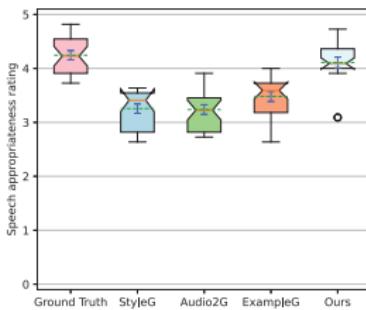
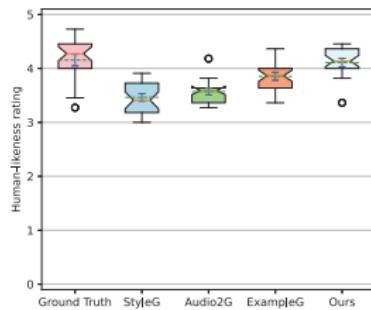
FID (Fréchet Inception Distance): Sự tương đồng về phân phối của
cử chỉ tạo sinh $\mathbf{g} \in \mathbb{R}^{1:M \times D}$ (generated) và $\mathbf{r} \in \mathbb{R}^{1:M \times D}$ (real)

$$FID = \|\mu_r - \mu_g\|^2 + \text{Tr} \left(\Sigma_r + \Sigma_g - 2\sqrt{\Sigma_r \Sigma_g} \right) \quad (7)$$

Trong đó

- μ : Trung bình các đặc trưng của cử chỉ
- Σ : Ma trận hiệp phương sai, Tr là tổng đường chéo chính.
- $\sqrt{\Sigma_r \Sigma_g}$: Sự tương đồng giữa các phân phối.

Kết quả



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

Outline

- ① Minh họa
- ② Giới thiệu
- ③ Công trình liên quan
- ④ Mô hình sinh cử chỉ
- ⑤ Thực nghiệm
- ⑥ Kết luận

Kết luận

- Mô hình **Diffusion**, ánh xạ xác suất làm tăng sự đa dạng đồng thời cho phép tạo ra các cử chỉ có chất lượng cao, giống con người.
- Mô hình của **OHGesture** tổng hợp các cử chỉ sao cho phù hợp với nhịp điệu âm thanh và ngữ nghĩa văn bản dựa trên Cross-Local Attention và Self-Attention.
- Sử dụng phương pháp **Classifier-free Guidance**, có thể điều khiển các điều kiện như cảm xúc, cử chỉ khởi tạo, có thể nội suy để suy luận ra giữa các cảm xúc khác nhau.

References I

- [1] Justine Cassell, Hannes Högni Vilhjálmsson, and Timothy Bickmore. "Beat: the behavior expression animation toolkit". In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. 2001, pp. 477–486.
- [2] Sergey Levine et al. "Gesture controllers". In: Acm siggraph 2010 papers. 2010, pp. 1–11.
- [3] Simon Alexanderson et al. "Listen, denoise, action! audio-driven motion synthesis with diffusion models". In: ACM Transactions on Graphics (TOG) 42.4 (2023), pp. 1–20.
- [4] Sicheng Yang et al. "Diffusestylegesture: Stylized audio-driven co-speech gesture generation with diffusion models". In: arXiv preprint arXiv:2305.04919 (2023).

References II

- [5] Lingting Zhu et al. "Taming diffusion models for audio-driven co-speech gesture generation". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 10544–10553.
- [6] Mingyang Sun et al. "Co-speech Gesture Synthesis by Reinforcement Learning with Contrastive Pre-trained Rewards". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 2331–2340.
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. 2020. arXiv: 2006.11239 [cs.LG]. URL: <https://arxiv.org/abs/2006.11239>.
- [8] Saeed Ghorbani et al. "ZeroEGGS: Zero-shot Example-based Gesture Generation from Speech". In: Computer Graphics Forum. Vol. 42. 1. Wiley Online Library. 2023, pp. 206–216.