SỰ TIẾN HÓA, KIẾN TRÚC VÀ TƯƠNG LAI CỦA TỔNG HỢP TIẾNG NÓI (TTS) TRONG KỶ NGUYÊN AI TẠO SINH

1. Giới thiệu

1.1. Bối cảnh nghiên cứu và tinh thần tự học

Trong bối cảnh giáo dục và công nghệ hiện đại, khả năng tự học (self-directed learning) không chỉ là một kỹ năng mềm mà là yếu tố sống còn đối với sự phát triển nghề nghiệp, đặc biệt là trong lĩnh vực trí tuệ nhân tạo (AI) đang thay đổi từng giờ. Khi sinh viên rời ghế nhà trường, sự hướng dẫn trực tiếp từ giảng viên sẽ biến mất, thay vào đó là biển thông tin mênh mông từ Internet, các hệ thống tìm kiếm tiên tiến và sự hỗ trợ đắc lực của các AI Agent. Việc tận dụng các nguồn lực này để giải mã những bài toán phức tạp như Tổng hợp Tiếng nói (Text-To-Speech - TTS) chính là phép thử quan trọng cho năng lực nghiên cứu độc lập.

Báo cáo này được biên soạn nhằm phục vụ nhu cầu tìm hiểu của các "nhà nghiên cứu trẻ" về bài toán TTS, cung cấp một cái nhìn toàn diện, sâu sắc và có tính hệ thống về lịch sử, hiện trạng và tương lai của công nghệ này. Không chỉ dừng lại ở mức độ khái niệm, báo cáo sẽ đi sâu vào kiến trúc kỹ thuật, tối ưu hóa pipeline triển khai và các vấn đề đạo đức cốt lõi.

1.2. Tổng quan về bài toán Text-To-Speech (TTS)

TTS là quá trình chuyển đổi văn bản đầu vào thành giọng nói nhân tạo. Đây là một bài toán "một-nhiều" (one-to-many) phức tạp: một câu văn bản có thể được nói theo vô số cách khác nhau tùy thuộc vào ngữ điệu, cảm xúc, tốc độ và đặc trưng giọng nói của người nói.[1] Lịch sử phát triển của TTS có thể được phân loại thành ba cấp độ (Level) chính, mỗi cấp độ đại diện cho một bước nhảy vọt về tư duy kỹ thuật và khả năng mô phỏng con người.

Báo cáo này sẽ phân tích chi tiết ba cấp độ này, từ những hệ thống ghép nối sơ khai (Level 1) đến các mô hình Deep Learning tinh chỉnh (Level 2) và cuối cùng là các mô hình tạo sinh Zero-shot tiên tiến nhất (Level 3), đồng thời đề xuất các chiến lược xây dựng pipeline tối ưu để cân bằng giữa hiệu suất và chất lượng.

---

2. Level 1: Khởi đầu của TTS

Trước khi Deep Learning thống trị, TTS được xây dựng dựa trên sự kết hợp giữa kiến thức ngôn ngữ học sâu sắc và kỹ thuật xử lý tín hiệu số (DSP). Level 1 đại diện cho nền tảng của ngành, nơi "luật lệ" và "dữ liệu thô" đóng vai trò chủ đạo.

2.1. Tổng hợp dựa trên Luật (Rule-Based / Formant Synthesis)

Đây là dạng sơ khai nhất của TTS, nơi máy tính tạo ra âm thanh hoàn toàn nhân tạo mà không sử dụng bất kỳ bản ghi âm nào của con người.

- **Cơ chế hoạt động:** Hệ thống mô phỏng đường dẫn âm thanh (vocal tract) của con người thông qua các tham số vật lý như tần số cơ bản (F_0), các tần số formant (cộng hưởng âm thanh), và độ nhiễu. Các quy tắc ngôn ngữ học sẽ chuyển đổi văn bản thành các tham số này. Ví dụ nổi tiếng nhất là giọng nói của thiết bị hỗ trợ nhà vật lý Stephen Hawking.[1]
- **Ưu điểm:**
  - **Dung lượng cực thấp:** Không cần lưu trữ file âm thanh, chỉ cần lưu trữ các quy tắc toán học.
  - **Kiểm soát tuyệt đối:** Có thể điều chỉnh từng thông số nhỏ nhất của âm thanh.
- **Nhược điểm:** Âm thanh phát ra mang tính chất "robot" đặc trưng, thiếu hoàn toàn tính tự nhiên và cảm xúc của con người.

2.2. Tổng hợp ghép nối (Concatenative Synthesis)

Phương pháp ghép nối đã thống trị thị trường thương mại trong suốt thập niên 90 và 2000, trở thành tiêu chuẩn cho "Level 1" trong ngữ cảnh hiện đại nhờ khả năng tạo ra giọng nói dễ hiểu từ các đoạn ghi âm thực.

2.2.1. Kiến trúc và cơ chế Unit Selection

Cốt lõi của phương pháp này là một cơ sở dữ liệu khổng lồ chứa hàng chục giờ ghi âm của một diễn viên lồng tiếng chuyên nghiệp. Các bản ghi này được cắt nhỏ thành các đơn vị âm thanh (units) như âm tiết, âm vị (phonemes), hoặc diphones (đoạn chuyển tiếp giữa hai âm vị).[1]

Khi cần tổng hợp một câu mới, hệ thống sẽ thực hiện thuật toán **Unit Selection** (Lựa chọn đơn vị) dựa trên hàm chi phí (Cost Function) gồm hai thành phần:

1. **Target Cost (Chi phí Mục tiêu):** Đo lường sự khác biệt giữa đơn vị âm thanh trong kho dữ liệu và yêu cầu ngữ âm của văn bản cần nói (ví dụ: độ cao, độ dài).
2. **Concatenation Cost (Chi phí Ghép nối):** Đo lường độ mượt mà khi ghép hai đơn vị âm thanh lại với nhau. Nếu ghép không tốt, người nghe sẽ thấy tiếng "lách cách" hoặc sự thay đổi đột ngột về âm sắc.

2.2.2. Đánh giá ưu và nhược điểm (Level 1)

Việc phân tích sâu Level 1 cho thấy rõ sự đánh đổi giữa tính ổn định và tính linh hoạt.

Tiêu chí	Đánh giá Chi tiết
Tính Tự nhiên	**Trung bình - Kém.** Trong các miền hẹp (như thông báo ga tàu, đọc số điện thoại), âm thanh rất tự nhiên vì hệ thống có thể ghép nguyên cả từ hoặc cụm từ đã ghi âm sẵn. Tuy nhiên, với văn bản tự do, hiện tượng "giọng nói chắp vá" (Frankenstein effect) xảy ra khiến ngữ điệu bị đứt gãy.[3]
Hiệu suất	**Rất cao.** Thuật toán chủ yếu là tìm kiếm và ghép file, tiêu tốn rất ít CPU, phù hợp cho các thiết bị nhúng giá rẻ.
Tài nguyên	**Tốn dung lượng lưu trữ.** Cần lưu trữ lượng lớn file WAV chất lượng cao.
Độ đa dạng	**Rất thấp.** Để đổi giọng nói hoặc thậm chí chỉ đổi cảm xúc (vui sang buồn), ta phải ghi âm lại toàn bộ cơ sở dữ liệu mới. Không có khả năng "học" hay thích nghi.[4]

2.3. Kết luận cho nghiên cứu Level 1

Mặc dù bị coi là lỗi thời trong các ứng dụng cao cấp, Level 1 vẫn là bài học quan trọng về **Xử lý Ngôn ngữ Tự nhiên (NLP)** phần frontend (chuẩn hóa văn bản) và là giải pháp tối ưu cho các hệ thống yêu cầu độ trễ bằng 0 và không có GPU, như các thiết bị IoT đơn giản hoặc các hệ thống thông báo khẩn cấp công cộng.

---

3. Level 2: Cuộc cách mạng Deep Learning và Pipeline cá nhân hóa (Personalized Pipeline)

Sự ra đời của Deep Learning đã thay đổi hoàn toàn cuộc chơi. Thay vì "chọn và ghép" các mảnh âm thanh cũ, Level 2 chuyển sang "tạo sinh" các đặc trưng âm thanh mới từ đầu. Điểm nhấn của Level 2 theo yêu cầu bài tập là khả năng tạo ra **pipeline cá nhân hóa**: mỗi người dùng tự ghi âm và tinh chỉnh (fine-tune) model cho riêng mình.

3.1. Kiến trúc Pipeline của Neural TTS

Để hiểu cách Level 2 hoạt động, ta cần giải phẫu pipeline tiêu chuẩn gồm ba giai đoạn:

1. **Text Analysis (Phân tích Văn bản):**
   - *Chuẩn hóa (Normalization):* Chuyển đổi các ký tự đặc biệt. Ví dụ: "10kg" thành "mười ki-lô-gam".
   - *Phonemization (Chuyển vị ngữ âm):* Chuyển văn bản thành chuỗi âm vị (phonemes). Ví dụ: "Học" -> /h ɔ k/.
2. **Acoustic Model (Mô hình Âm học):**
   - Đây là "bộ não" của hệ thống. Nó nhận đầu vào là chuỗi âm vị và dự đoán **Mel-spectrogram** (biểu đồ phổ Mel) - một biểu diễn hình ảnh của âm thanh theo thời gian và tần số.
3. **Vocoder (Bộ tạo sóng):**
   - Chuyển đổi Mel-spectrogram thành dạng sóng âm thanh (waveform) có thể nghe được.[6]

3.2. Các kiến trúc Acoustic Model chủ đạo

Trong Level 2, có sự cạnh tranh gay gắt giữa hai dòng kiến trúc: Autoregressive (Tự hồi quy) và Non-Autoregressive (Phi tự hồi quy).

3.2.1. Tacotron 2 (Autoregressive)

- **Cơ chế:** Tacotron 2 sử dụng kiến trúc Encoder-Decoder với cơ chế Attention (Sự chú ý). Nó sinh ra từng khung hình (frame) của spectrogram một cách tuần tự. Khung hình thứ t được sinh ra dựa trên thông tin của khung hình thứ $t-1$.[6]
- **Ưu điểm:** Chất lượng âm thanh và ngữ điệu cực kỳ tự nhiên, giàu cảm xúc.
- **Nhược điểm:**
  - *Tốc độ chậm:* Do phải sinh tuần tự, độ trễ cao.
  - *Lỗi lặp/bỏ từ:* Đôi khi model bị "lắp bắp" hoặc bỏ qua từ do lỗi ở cơ chế Attention.[8]

3.2.2. FastSpeech 2 (Non-Autoregressive)

- **Cơ chế:** FastSpeech 2 sinh ra toàn bộ spectrogram cùng một lúc (song song). Nó sử dụng một mô đun **Duration Predictor** để dự đoán thời lượng của mỗi âm vị, sau đó mở rộng chuỗi đầu vào và tính toán song song.[9]
- **Ưu điểm:**
  - *Tốc độ siêu nhanh:* Nhanh hơn Tacotron 2 hàng trăm lần (Real-time Factor cực thấp).
  - *Ổn định:* Loại bỏ hoàn toàn lỗi lặp từ/bỏ từ.
  - *Kiểm soát:* Dễ dàng điều chỉnh tốc độ, độ cao, năng lượng của giọng nói.[8]
- **Nhược điểm:** Ngữ điệu có thể "phẳng" hơn một chút so với Tacotron 2, tuy nhiên với các cải tiến gần đây (như Variance Adaptor), khoảng cách này đã được thu hẹp đáng kể.

3.3. Chiến lược triển khai Pipeline cá nhân hóa (Fine-tuning)

Đây là trọng tâm của Level 2: Tối ưu hóa tài nguyên bằng cách tinh chỉnh model riêng cho từng người dùng.

3.3.1. Quy trình thực hiện (Pipeline)

1. **Pre-training:** Một model lớn (như FastSpeech 2) được huấn luyện trên hàng nghìn giờ dữ liệu đa giọng nói (Multi-speaker dataset như LibriTTS). Model này học được các quy luật vật lý của âm thanh và cách phát âm chuẩn.[12]
2. **Adaptation Data Collection (Thu thập dữ liệu thích nghi):** Người dùng cuối chỉ cần ghi âm một lượng dữ liệu nhỏ (khoảng 15 phút đến 1 giờ).
3. **Fine-tuning (Tinh chỉnh):**
   - Hệ thống đóng băng (freeze) các lớp cơ sở của model (phần hiểu ngôn ngữ).
   - Chỉ cập nhật trọng số của các lớp Decoder hoặc sử dụng **Speaker Embeddings** (các vector đặc trưng giọng nói như x-vectors hoặc d-vectors) để học chất giọng riêng của người dùng.[13]

3.3.2. Đánh giá Ưu và Nhược điểm (Level 2)

Tiêu chí	Đánh giá Chi tiết
Tính Tự nhiên	**Cao.** Giọng nói mượt mà, không còn các điểm nối lách cách. Ngữ điệu tốt hơn hẳn Level 1.
Hiệu suất	**Tốt.** Sau khi fine-tune, model chạy rất nhẹ (đặc biệt là FastSpeech 2), có thể chạy trên CPU hoặc thiết bị di động.[14]
Tài nguyên	**Trung bình.** Cần GPU để huấn luyện (giai đoạn fine-tune), nhưng khi chạy (inference) thì tốn ít tài nguyên hơn Level 3 rất nhiều.
Công sức người dùng	**Trung bình - Cao.** Người dùng phải bỏ công ghi âm và chờ đợi quá trình fine-tune (vài giờ). Không "ăn liền" như Level 3.

4. Level 3: Kỷ nguyên Generative Audio và Zero-Shot Synthesis

Level 3 đại diện cho sự dịch chuyển mô hình (paradigm shift) từ "xử lý tín hiệu" sang "mô hình hóa ngôn ngữ" (Language Modeling). Tại đây, âm thanh không còn được xử lý như các sóng liên tục mà được coi như một ngôn ngữ với các từ vựng riêng.

4.1. Bản chất của Zero-Shot TTS

Zero-shot (hoặc Few-shot) TTS là khả năng tổng hợp giọng nói của một người mà model **chưa từng gặp bao giờ** trong quá trình huấn luyện, chỉ dựa trên một mẫu âm thanh tham chiếu ngắn (3-5 giây).[15]

4.1.1. Từ Spectrogram đến Discrete Codes (Mã rời rạc)

Khác biệt lớn nhất về mặt kiến trúc của Level 3 (như VALL-E) so với Level 2 là việc sử dụng **Neural Audio Codecs** (như EnCodec hoặc SoundStream).

- Các mô hình này nén âm thanh liên tục thành các chuỗi mã số rời rạc (tokens), tương tự như cách văn bản được mã hóa thành token.
- Điều này cho phép áp dụng các kiến trúc Transformer mạnh mẽ (tương tự GPT-4) để "dự đoán token âm thanh tiếp theo" dựa trên token văn bản và token âm thanh mẫu.[16]

4.2. Các hướng tiếp cận kiến trúc chủ đạo

4.2.1. Language Modeling (VALL-E, XTTS)

- **Cơ chế:** Coi TTS là bài toán tiếp diễn chuỗi (sequence continuation). Đầu vào là "Văn bản + 3s âm thanh mẫu". Đầu ra là phần âm thanh còn lại.
- **Ưu điểm:** Khả năng "bắt chước" cực tốt. Nó có thể sao chép không chỉ giọng nói mà cả môi trường âm học (tiếng vang, nhiễu nền) và cảm xúc của mẫu tham chiếu.[18]
- **Nhược điểm:**
  - *Ảo giác (Hallucination):* Do bản chất xác suất, đôi khi model nói sai từ, lặp từ hoặc cười/khóc bất thình lình.
  - *Tài nguyên lớn:* Đòi hỏi model khổng lồ (hàng trăm triệu tham số) và GPU mạnh để chạy.[15]

4.2.2. Flow Matching (Voicebox, Matcha-TTS)

Đây là hướng đi mới nhất và hứa hẹn nhất, khắc phục nhược điểm của VALL-E.

- **Cơ chế:** Sử dụng lý thuyết **Flow Matching** (tương tự Diffusion nhưng hiệu quả hơn). Thay vì dự đoán token rời rạc, nó học một trường vectơ (vector field) để biến đổi nhiễu (noise)

thành Mel-spectrogram của giọng nói mục tiêu.[20]

- **Ưu điểm:**
  - *Ổn định hơn:* Ít bị ảo giác hơn VALL-E.
  - *Nhanh hơn:* Cần ít bước lấy mẫu hơn so với Diffusion truyền thống.[22]
  - *Đa năng:* Có thể dùng để khử nhiễu, sửa lỗi phát âm (editing) ngoài việc tạo tiếng nói.

4.2.3. Diffusion Models (Tortoise TTS)

- **Cơ chế:** Sử dụng mô hình khuếch tán để dần dần "khử nhiễu" từ một bức tranh nhiễu trắng thành spectrogram.
- **Ưu điểm:** Chất lượng âm thanh cực cao, giàu chi tiết.
- **Nhược điểm:** Tốc độ suy luận (inference) cực chậm, khó triển khai thời gian thực.[24]

4.3. Đánh giá uu và nhược điểm (Level 3)

Tiêu chí	Đánh giá Chi tiết
Tính Tự nhiên	**Rất cao (Human Parity).** Có thể tạo ra tiếng thở, tiếng cười, và các sắc thái tinh tế.
Công sức người dùng	**Rất thấp.** Chỉ cần upload 1 file âm thanh ngắn bất kỳ. Không cần ghi âm studio, không cần chờ training.
Tài nguyên	**Rất lớn.** Chi phí tính toán (Compute cost) chuyển từ giai đoạn training (Level 2) sang giai đoạn inference. Yêu cầu phần cứng mạnh mẽ để chạy thời gian thực.
Độ đa dạng ngôn ngữ	**Cao.** Hỗ trợ Cross-lingual voice cloning (Người Việt nói tiếng Anh bằng giọng của mình).[15]

---

5. Xây dựng Pipeline tối ưu: Tối thiểu hóa nhược điểm, Tối đa hóa ưu điểm

Phần này đề xuất các chiến lược kỹ thuật để kết hợp các ưu điểm của các Level và giải quyết các thách thức chung (Hiệu suất, Tài nguyên, Đa ngôn ngữ, Cảm xúc).

5.1. Chiến lược Hybrid (Kết hợp Level 2 và Level 3)

- **Vấn đề:** Level 3 rất linh hoạt nhưng tốn kém và chậm. Level 2 nhanh và nhẹ nhưng cần dữ liệu training.
- **Giải pháp:** Sử dụng Level 3 để **tạo dữ liệu** cho Level 2 (Synthetic Data Distillation).
  - *Bước 1:* Dùng model Zero-shot (Level 3) để tạo ra bộ dữ liệu giọng nói chất lượng cao từ một mẫu ngắn của người dùng.
  - *Bước 2:* Dùng bộ dữ liệu nhân tạo này để fine-tune một model nhẹ (như FastSpeech 2 hoặc VITS).
  - *Kết quả:* Ta có một pipeline "setup nhanh" (nhờ Level 3) nhưng "chạy nhanh và nhẹ" (nhờ Level 2) trên thiết bị người dùng.[26]

5.2. Tối ưu hóa độ trễ (Latency Optimization)

Để đạt được hiệu suất nhanh (như Level 1) với chất lượng Level 3, các kỹ thuật sau được áp dụng:

- **Streaming Architecture:** Không đợi sinh xong cả câu mới phát. Sử dụng kiến trúc **Chunk-based generation**. Ngay khi model sinh ra đoạn âm thanh đầu tiên, hệ thống gửi ngay về client qua giao thức **WebSocket** hoặc **gRPC** để phát (play). Điều này giảm "Time to First Byte" xuống dưới 200ms.[28]
- **KV Cache Optimization:** Với các model dạng Transformer (như VALL-E), việc quản lý bộ nhớ đệm Key-Value (KV Cache) là tối quan trọng để tránh tính toán lại các token cũ, giúp tăng tốc độ sinh chuỗi dài.[30]
- **Non-Autoregressive Backbones:** Ưu tiên sử dụng các kiến trúc Flow Matching hoặc FastSpeech cho các ứng dụng cần phản hồi tức thì, thay vì Autoregressive.[22]

5.3. Tối ưu hóa tài nguyên trên thiết bị (Edge Deployment)

Để chạy model trên điện thoại/laptop yếu:

- **Quantization (Lượng tử hóa):** Chuyển trọng số model từ dạng số thực 32-bit (FP32) sang số nguyên 8-bit (INT8). Kỹ thuật này giảm 4 lần dung lượng model và tăng tốc độ chạy trên CPU mà ít làm giảm chất lượng âm thanh.[31]
- **Knowledge Distillation (Chưng cất tri thức):** Dạy một model nhỏ (Student) học theo hành vi của model lớn (Teacher). Ví dụ: Dạy một model Flow Matching ít bước (4 bước) bắt chước một model Diffusion nhiều bước (50 bước).[33]

5.4. Thách thức về cảm xúc và đa ngôn ngữ

- **Kiểm soát Cảm xúc:** Sử dụng **Natural Language Prompting**. Thay vì chỉnh thanh trượt "vui/buồn", người dùng nhập lệnh: "Nói câu này với giọng thì thầm, sợ hãi". Các nghiên cứu mới đang tích hợp Large Language Model (LLM) vào TTS để hiểu các chỉ dẫn này.[34]
- **Đa ngôn ngữ (Cross-lingual):** Thách thức lớn là giữ được "màu giọng" (timbre) mà không mang theo "giọng địa phương" (accent) của ngôn ngữ gốc. Giải pháp là sử dụng các kiến trúc tách biệt (disentangled representation) để tách riêng thông tin về người nói và thông tin về ngôn ngữ.[36]

---

6. Đạo đức nghiên cứu: Watermarking và chống Deepfake

Với sức mạnh của Level 3, rủi ro về giả mạo giọng nói (Deepfake) là cực lớn. Yêu cầu về đạo đức nghiên cứu bắt buộc phải tích hợp cơ chế đánh dấu bản quyền (Watermarking).

6.1. Watermarking Không thể nghe thấy (Imperceptible Watermarking)

Đây là kỹ thuật nhúng một tín hiệu ẩn vào file âm thanh mà tai người không nghe thấy, nhưng máy tính có thể phát hiện.

- **SynthID (Google DeepMind):**
  - *Cơ chế:* Tác động trực tiếp vào quá trình sinh (generation process). Khi model dự đoán token tiếp theo, thay vì chọn ngẫu nhiên theo xác suất, nó sẽ ưu tiên các token tuân theo một quy luật toán học bí mật (g-function).
  - *Ưu điểm:* Cực kỳ bền vững (robust). Dù file âm thanh bị nén MP3, tăng tốc độ, hay thêm nhiễu nền, quy luật phân phối của các token vẫn còn đó để phát hiện.[37]
- **AudioSeal (Meta):**
  - *Cơ chế:* Tập trung vào khả năng **phát hiện cục bộ** (localized detection). Nó có thể chỉ ra chính xác giây thứ mấy trong một đoạn ghi âm dài là do AI tạo ra. Điều này quan trọng để phát hiện các file âm thanh "lai" (người thật nói xen kẽ AI).[39]

6.2. So sánh hiệu quả và thách thức

Các nghiên cứu (như AudioMarkBench) cho thấy cuộc đua giữa "tạo watermak" và "phá watermark" rất khốc liệt.

- **Thách thức:** Các tấn công "Hộp đen" (Black-box attacks) như việc ghi âm lại qua loa (re-recording) hoặc dùng các bộ lọc thay đổi cao độ có thể làm suy yếu watermark.[40]
- **Kết luận đạo đức:** Watermarking là lớp phòng thủ cần thiết nhưng chưa đủ. Cần kết hợp với chữ ký số (provenance) và quy định pháp lý về việc công bố nguồn gốc nội dung AI.

---

7. Tổng kết và định hướng

Qua quá trình khảo sát từ Level 1 đến Level 3, bức tranh toàn cảnh về TTS đã rõ ràng:

1. **Sự chuyển dịch:** Từ *Luật cứng* (Level 1) sang *Học thống kê* (Level 2) và cuối cùng là *Học ngữ cảnh* (Level 3).
2. **Sự đánh đổi:** Level 3 mang lại sự tiện lợi và chất lượng tối thượng nhưng đi kèm chi phí tính toán cao và rủi ro đạo đức. Level 2 (Fine-tuning) là điểm cân bằng tốt nhất hiện tại cho các ứng dụng thương mại cần tối ưu chi phí và hiệu năng.
3. **Lời khuyên cho bài tập:**
    - Khi xây dựng pipeline, hãy cân nhắc **Mục đích sử dụng**. Nếu làm ứng dụng đọc báo cho người khiếm thị trên điện thoại cũ, Level 2 (FastSpeech 2 + Quantization) là lựa chọn vàng. Nếu làm công cụ sáng tạo nội dung cho TikTok/Youtube, Level 3 (Zero-shot) là bắt buộc.
    - Luôn tích hợp **Watermarking** trong tư duy thiết kế hệ thống để đảm bảo trách nhiệm xã hội.

Với sự hỗ trợ của các thư viện mã nguồn mở (như Coqui TTS, TorToise, VALL-E implementations) và khả năng tự học, các bạn sinh viên hoàn toàn có thể tự tay xây dựng một hệ thống TTS hiện đại, đóng góp vào sự phát triển của lĩnh vực đầy tiềm năng này.

Bảng Tổng hợp So sánh Các Hướng Tiếp cận

Đặc điểm	Level 1 (Ghép nối)	Level 2 (Fine-Tuning)	Level 3 (Zero-Shot)
Công nghệ lõi	Unit Selection, DSP	Deep Neural Networks (CNN/RNN/Transformer)	Language Modeling, Flow Matching, Diffusion
Độ tự nhiên	Thấp (Robotic, Lắp ghép)	Cao (Mượt mà, ổn định)	Rất cao (Giàu cảm xúc, như người thật)

Yêu cầu dữ liệu	Lớn (Cơ sở dữ liệu âm thanh gốc)	Trung bình (Cần 15p-1h để fine-tune)	Thấp (Cần 3s prompt để clone)
Thời gian Setup	Tức thì (sau khi có DB)	Chậm (Cần thời gian training)	Tức thì (Real-time cloning)
Tài nguyên chạy	Thấp (CPU, RAM)	Trung bình (CPU/GPU nhẹ)	Cao (GPU mạnh, VRAM lớn)
Ứng dụng phù hợp	Thông báo công cộng, IoT	Trợ lý ảo, Đọc sách nói (cố định)	Dubbing phim, NPC Game, Content Creation

Nguồn trích dẫn

1. Speech synthesis - Wikipedia, , https://en.wikipedia.org/wiki/Speech_synthesis
2. The Rise of the Talking Machines: A Journey Through the History of Text to Speech Technology — Part 1 | by Tharuka KasthuriArachchi | Medium, , https://medium.com/analytics-vidhya/the-rise-of-the-talking-machines-a-journey-through-the-history-of-text-to-speech-technology-b76c8b0a5a1d
3. What are the differences between concatenative and parametric TTS? - Zilliz, , https://zilliz.com/ai-faq/what-are-the-differences-between-concatenative-and-parametric-tts
4. (PDF) Speech synthesis systems: Disadvantages and limitations - ResearchGate, , https://www.researchgate.net/publication/325554736_Speech_synthesis_systems_Disadvantages_and_limitations
5. What are the differences between concatenative and parametric TTS? - Milvus, , https://milvus.io/ai-quick-reference/what-are-the-differences-between-concatenative-and-parametric-tts
6. From Text to Speech: A Deep Dive into TTS Technologies | by Zilliz | Medium, , https://medium.com/@zilliz_learn/from-text-to-speech-a-deep-dive-into-tts-technologies-18ea409f20e8
7. Speech synthesis: A review of the best text to speech architectures with Deep Learning, , https://theaisummer.com/text-to-speech/
8. FastSpeech: Revolutionizing Speech Synthesis with Parallel Processing - Vapi AI Blog, , https://vapi.ai/blog/fast-speech
9. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech - Microsoft

Research, , https://www.microsoft.com/en-us/research/articles/fastspeech-2-fast-and-high-quality-end-to-end-text-to-speech/
10. [2006.04558] FastSpeech 2: Fast and High-Quality End-to-End Text to Speech - arXiv, , https://arxiv.org/abs/2006.04558
11. FastSpeech: New text-to-speech model improves on speed, accuracy, and controllability, , https://www.microsoft.com/en-us/research/blog/fastspeech-new-text-to-speech-model-improves-on-speed-accuracy-and-controllability/
12. What techniques are available for fine-tuning TTS models? - Milvus, , https://milvus.io/ai-quick-reference/what-techniques-are-available-for-finetuning-tts-models
13. Enhancing Multilingual Human-Like Speech and Voice Cloning with NVIDIA Riva TTS, , https://developer.nvidia.com/blog/enhancing-multilingual-human-like-speech-and-voice-cloning-with-nvidia-riva-tts/
14. How Text-to-Speech Models Work: Theory and Practice - it-jim, , https://www.it-jim.com/blog/how-text-to-speech-models-work-theory-and-practice/
15. XTTS: a Massively Multilingual Zero-Shot Text-to-Speech Model - arXiv, , https://arxiv.org/html/2406.04904v1
16. VALL-E - Microsoft, , https://www.microsoft.com/en-us/research/project/vall-e-x/
17. VALL-E - The Future of Text to Speech? | Towards Data Science, , https://towardsdatascience.com/vall-e-the-future-of-text-to-speech-d090b6ede07b/
18. VALL-E 2: Enhancing the robustness and naturalness of text-to-speech models - Microsoft, , https://www.microsoft.com/en-us/research/articles/vall-e-2-enhancing-the-robustness-and-naturalness-of-text-to-speech-models/
19. How to create custom voice using XTTS - Smallest.ai, , https://smallest.ai/blog/custom-voice-cloning-using-xtts
20. Introducing Voicebox: The first generative AI model for speech to generalize across tasks with state-of-the-art performance - AI at Meta, , https://ai.meta.com/blog/voicebox-generative-ai-model-speech/
21. Flow Matching-Based TTS Model - Emergent Mind, , https://www.emergentmind.com/topics/flow-matching-based-tts-model
22. [2306.15687] Voicebox: Text-Guided Multilingual Universal Speech Generation at Scale, , https://arxiv.org/abs/2306.15687
23. VoiceFlow: Efficient Text-to-Speech with Rectified Flow Matching - arXiv, , https://arxiv.org/html/2309.05027v3
24. How to Use Tortoise TTS Voice Models for Speech Generation? - ProjectPro, , https://www.projectpro.io/article/tortoise-tts-voice-models/1101
25. neonbjb/tortoise-tts: A multi-voice TTS system trained with an emphasis on quality - GitHub, , https://github.com/neonbjb/tortoise-tts
26. [2501.08566] Towards Lightweight and Stable Zero-shot TTS with Self-distilled Representation Disentanglement - arXiv, , https://arxiv.org/abs/2501.08566
27. StyleTTS-ZS: Efficient High-Quality Zero-Shot Text-to-Speech Synthesis with

Distilled Time-Varying Style Diffusion - ACL Anthology, , https://aclanthology.org/2025.naacl-long.242.pdf
28. How to optimise latency while buiding Voice AI agents | Blog, , https://comparevoiceai.com/blog/latency-optimisation-voice-agent
29. How to Cut TTS Latency for Real-Time Voice Apps: Practical, Measurable Steps - Dupdub AI, , https://www.dupdub.com/blog/tts-latency-optimization
30. Next Tokens Denoising for Speech Synthesis - arXiv, , https://arxiv.org/html/2507.22746v1
31. Post-training quantization | Google AI Edge, , https://ai.google.dev/edge/litert/models/post_training_quantization
32. Quantization aware training - Model optimization - TensorFlow, , https://www.tensorflow.org/model_optimization/guide/quantization/training
33. DMOSpeech: Direct Metric Optimization via Distilled Diffusion Model in Zero-Shot Speech Synthesis | OpenReview, , https://openreview.net/forum?id=ojF1rXmUdy¬eId=ybTDLmHr46
34. Controlling Emotion in Text-to-Speech with Natural Language Prompts - ISCA Archive, , https://www.isca-archive.org/interspeech_2024/bott24_interspeech.pdf
35. PromptEVC: Controllable Emotional Voice Conversion with Natural Language Prompts, , https://arxiv.org/html/2505.20678v1
36. Zero-Shot Cross-Lingual Text-to-Speech With Style-Enhanced Normalization and Auditory Feedback Training Mechanism - IEEE Xplore, , https://ieeexplore.ieee.org/iel8/10723155/10835842/10910244.pdf
37. SynthID - Google DeepMind, , https://deepmind.google/models/synthid/
38. SynthID: Tools for watermarking and detecting LLM-generated Text content_copy, , https://ai.google.dev/responsible/docs/safeguards/synthid
39. Proactive Detection of Voice Cloning with Localized Watermarking | Research - AI at Meta, , https://ai.meta.com/research/publications/proactive-detection-of-voice-cloning-with-localized-watermarking/
40. AudioMarkBench: Benchmarking Robustness of Audio Watermarking, , https://proceedings.neurips.cc/paper_files/paper/2024/file/5d9b7775296a641a1913ab6b4425d5e8-Paper-Datasets_and_Benchmarks_Track.pdf