ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Bài toán Text-To-Speech (TTS)

Giảng viên hướng dẫn:
PGS. TS. Lê Hồng Phương
CN. Phạm Ngọc Hải

Sinh viên thực hiện:
Cao Hải An - 23001818

Hà Nội, Ngày 3 tháng 12 năm 2025

Mục lục

Danh sách hình vẽ

Danh sách bảng

Chương 1

Tổng quan đề tài

1.1 Bối cảnh và động lực nghiên cứu

Trong kỷ nguyên số, giao tiếp người-máy đang chuyển dịch mạnh mẽ sang mô thức ***đa phương thức (multimodal)*** — nơi giọng nói đóng vai trò trung tâm. Bài toán ***Tổng hợp tiếng nói (Text-to-Speech - TTS)*** đã trải qua các bước tiến hóa về mặt mô hình toán học: từ các phương pháp *ghép nối đơn vị* (concatenative) dựa trên quy tắc, sang mô hình *thống kê tham số* (HMM-based), đến các kiến trúc *học sâu* (Deep Learning) và hiện nay là các mô hình *sinh tự hồi quy* (Autoregressive Generative Models) quy mô lớn.

Động lực nghiên cứu xuất phát từ nhu cầu giải quyết bài toán tối ưu hóa đa mục tiêu: nâng cao chất lượng âm thanh (tính tự nhiên), giảm thiểu độ trễ thực thi (latency), và khả năng thích ứng linh hoạt với các tác vụ như trợ lý ảo, đọc báo tự động hay tái tạo giọng nói (voice cloning).

1.2 Phạm vi nghiên cứu và mục tiêu chính

Báo cáo này tập trung giải quyết các mục tiêu cụ thể sau:

1. **Phân tích so sánh các kiến trúc TTS tiêu biểu:** Đối chiếu các hệ thống từ cổ điển (ghép nối), mô hình tham số (Neural Parametric - tách biệt mô hình âm học và bộ mã hóa tiếng nói), đến các mô hình sinh hiện đại (End-to-End, Non-autoregressive).

2. **Đánh giá hiệu năng và tính ứng dụng:** Khảo sát sự đánh đổi (trade-off) giữa các yếu tố: tốc độ suy luận, chi phí tài nguyên tính toán, độ tự nhiên của ngữ điệu (prosody), và khả năng mở rộng dữ liệu.

3. **Tổng hợp các kỹ thuật tối ưu hóa:** Tập trung vào các phương pháp nén mô hình (quantization, pruning), suy luận song song (parallel inference), và các kỹ thuật thích nghi giọng nói (speaker adaptation) nhằm tối ưu hóa bài toán triển khai thực tế.

4. **Nhận diện thách thức và định hướng tương lai:** Phân tích các vấn đề mở như mô hình hóa ngữ điệu cảm xúc, hỗ trợ ngôn ngữ tài nguyên thấp (low-resource languages), và các khía cạnh an toàn thông tin trong nhân bản giọng nói (water-marking, chống deepfake).

1.3 Chi tiết bài toán TTS và các nghiên cứu tiêu biểu

1.3.1 Định nghĩa và khái niệm cốt lõi

Text-to-Speech (TTS) Là quá trình ánh xạ dữ liệu văn bản rời rạc (discrete text) thành tín hiệu âm thanh liên tục (continuous waveform). Về mặt toán học, đây là bài toán mô hình hóa hàm mục tiêu $f : X \to Y$ với X là chuỗi ký tự/âm vị và Y là chuỗi mẫu tín hiệu âm thanh.

Front-end (Khối xử lý ngôn ngữ) Các bước tiền xử lý văn bản đầu vào, bao gồm: chuẩn hóa (normalization), tách từ (tokenization), chuyển đổi tự-vị sang âm-vị (Grapheme-to-Phoneme - G2P) và trích xuất đặc trưng ngữ điệu. Chất lượng của Front-end ảnh hưởng trực tiếp đến khả năng tổng quát hóa của hệ thống.

Acoustic Model (Mô hình âm học) Mô hình học sâu chịu trách nhiệm chuyển đổi chuỗi đặc trưng ngôn ngữ (phonemes) thành biểu diễn trung gian của âm thanh, thường là *phổ Mel (Mel-spectrogram)*[10]. Các kiến trúc tiêu biểu bao gồm Tacotron2 (RNN-based)[7] và FastSpeech2 (Transformer-based).[6][2]

Vocoder (Bộ mã hóa tiếng nói) Mô hình đảm nhiệm việc khôi phục dạng sóng âm thanh (waveform) từ biểu diễn phổ Mel, giải quyết bài toán khôi phục pha (phase reconstruction). Các đại diện hiện đại gồm WaveNet[8] và HiFi-GAN[4].

Zero-shot Voice Cloning Khả năng tái tạo giọng nói của một đối tượng chưa từng xuất hiện trong tập huấn luyện, chỉ dựa trên một mẫu tham chiếu ngắn (vài giây). Kỹ thuật này thường sử dụng các vector đặc trưng (speaker embedding) trong không gian tiềm ẩn.

1.3.2 Phạm vi và mục tiêu nghiên cứu

Báo cáo này tập trung vào việc khảo sát và phân tích các hướng phát triển TTS hiện nay:

1. So sánh ba lớp kiến trúc chính:
    - **Concatenative (Level 1):** Phương pháp rule-based, ghép nối đơn vị âm thanh, hiệu năng cao nhưng thiếu tự nhiên.
    - **Neural parametric (Level 2):** Mô hình học sâu tạo Mel-spectrogram, kết hợp vocoder để sinh speech chất lượng cao.
    - **Generative large models (Level 3):** Few-shot/zero-shot, mô hình autoregressive hoặc diffusion, tốn nhiều tài nguyên nhưng độ tự nhiên vượt trội.
2. **Nghiên cứu và áp dụng các kỹ thuật tối ưu hóa pipeline:**[2]
    - *Quantization (Lượng tử hóa):* Giảm độ chính xác số học của trọng số mô hình (ví dụ: chuyển từ dấu phẩy động 32-bit $FP32$ sang số nguyên 8-bit $INT8$) nhằm nén dung lượng và tăng tốc độ tính toán ma trận, chấp nhận một sai số lượng tử nhỏ.
    - *Pruning (Cắt tỉa tham số):* Phương pháp loại bỏ các liên kết hoặc nơ-ron có trọng số gần bằng 0 để tạo ra các ma trận thưa (sparse matrices), giúp giảm khối lượng tính toán (FLOPs) mà không làm suy giảm đáng kể độ chính xác.

- *Knowledge Distillation (Chưng cất tri thức):* Kỹ thuật huấn luyện mô hình nhỏ gọn (Student) học cách mô phỏng phân phối xác suất đầu ra (soft targets) của một mô hình lớn và phức tạp hơn (Teacher) thông qua việc tối ưu hóa hàm mất mát *KL-Divergence*.
- *Streaming Inference (Suy luận theo luồng):* Cơ chế xử lý và sinh âm thanh theo dạng cuốn chiếu (chunk-based/incremental processing) thay vì chờ xử lý toàn bộ văn bản, giúp giảm độ trễ phản hồi ban đầu (Time-to-First-Audio) xuống mức thời gian thực.

Chương 2

Phân tích kiến trúc kỹ thuật (Technical Landscape)

2.1 Level 1: Concatenative / Rule-based Synthesis

2.1.1 Khái quát

Level 1 TTS dựa trên ***concatenative synthesis***, tức là ghép nối các đơn vị âm thanh đã ghi sẵn từ cơ sở dữ liệu. Đây là lớp TTS cổ điển, ưu điểm là độ trễ thấp, sử dụng tài nguyên ít, nhưng nhược điểm là tính tự nhiên hạn chế và khó mở rộng. Các đơn vị thường là phonemes, diphthongs, syllables hoặc words.

Các nghiên cứu điển hình trong Level 1 bao gồm ba phương pháp chính:

2.1.2 Phương pháp 1: Unit Selection Synthesis (US)

Kiến trúc và pipeline chi tiết [3]

1. **Tiền xử lý văn bản (Text Preprocessing):** Mục tiêu là biến văn bản raw input thành chuỗi các đơn vị có thể chuyển thành âm thanh. Bao gồm:
    - ***Normalization:*** Chuyển các số, ngày tháng, ký hiệu đặc biệt thành dạng chữ viết đầy đủ. Ví dụ: "2025" → "hai nghìn không trăm hai mươi lăm".
    - ***Tokenization:*** Tách văn bản thành các token cơ bản như từ, syllable hoặc chữ cái tùy hệ thống.
    - ***Grapheme-to-Phoneme (G2P):*** Chuyển từ (grapheme) thành dãy âm vị (phoneme) để xác định cách phát âm. Đây là bước quan trọng vì US dựa trên phoneme-level để chọn đơn vị âm thanh.
    - ***Prosody annotation (tùy chọn):*** Dán nhãn sơ bộ ngữ điệu như dấu ngắt câu, trọng âm, hoặc nhịp câu để hỗ trợ bước ghép nối sau này.
2. **Database âm thanh (Speech Unit Database):** Là kho lưu trữ các đơn vị âm thanh đã ghi âm sẵn. Yếu tố quan trọng:
    - ***Phoneme/Diphone/Syllable labeling:*** Mỗi đơn vị được gán nhãn chính xác để máy biết nó thuộc phoneme nào, vị trí trong từ, và cách phát âm chuẩn.

- ***Contextual annotation:*** Thông tin về âm thanh xung quanh (ví dụ: âm trước/sau, âm vị liền kề) để cải thiện trôi chảy khi ghép nối.
- ***Prosody features:*** Pitch (cao độ), duration (thời lượng), intensity (cường độ) được đo và lưu lại cho mỗi đơn vị.
- Database càng đa dạng (nhiều giọng, tốc độ, cảm xúc), chất lượng output càng cao.

3. **Lựa chọn đơn vị (Unit Selection):** Đây là bước cốt lõi của US:
   - ***Target cost:*** Đo lường mức độ phù hợp giữa đơn vị trong database với âm vị yêu cầu từ input. Bao gồm:
     - Phonetic match: đơn vị có phát âm giống âm yêu cầu.
     - Prosody match: đơn vị có pitch, duration gần với ngữ cảnh.
   - ***Concatenation cost:*** Đánh giá độ trôi chảy khi nối hai đơn vị. Bao gồm:
     - Spectral continuity: sự mượt mà về phổ tần số.
     - Temporal alignment: đồng bộ thời gian giữa các đơn vị.
   - ***Dynamic Programming:*** Thuật toán tìm chuỗi đơn vị tối ưu sao cho tổng *target cost + concatenation cost* là nhỏ nhất. Đây là cách US giảm rời rạc và tối ưu trôi chảy.
4. **Ghép nối đơn vị (Concatenation & Post-processing):** Sau khi chọn đơn vị tối ưu:
   - Nối waveform của các đơn vị theo thứ tự.
   - Áp dụng ***pitch correction*** hoặc ***time-stretching*** để điều chỉnh cao độ và thời lượng cho trôi chảy.
   - Sử dụng fade-in/fade-out hoặc envelope shaping để giảm hiện tượng click, pop, hoặc rời rạc giữa các đơn vị.
   - Kết quả là waveform cuối cùng có thể phát lại như giọng nói tự nhiên trong giới hạn của hệ thống US.

Cách giải quyết nhược điểm [2]

- ***Rời rạc:*** Giảm bằng tối ưu *concatenation cost* và rule-based prosody adjustment.
- ***Thiếu prosody:*** Sử dụng nhãn prosody từ database và điều chỉnh pitch/duration dựa trên ngữ cảnh câu.
- ***Mở rộng giọng mới:*** Yêu cầu thu âm thêm database, nhưng với rule-based unit selection, có thể thêm từng nhóm đơn vị mà không cần ghi toàn bộ từ vựng.

Ứng dụng thực tế [2]

- Thiết bị nhúng IoT, trợ lý giọng nói cơ bản.
- Hệ thống đọc văn bản đơn giản như thông báo tại sân bay, tàu điện, hoặc hướng dẫn GPS.
- Ứng dụng yêu cầu tốc độ cực nhanh, độ trễ thấp, predictable output.

2.1.3 Phương pháp 2: Diphone Synthesis

Kiến trúc và pipeline chi tiết

1. **Database diphone:** Mỗi ***diphone*** bao gồm phần cuối của âm vị trước và phần đầu của âm vị sau. Vì vậy, database chứa khoảng n^2 diphone cho n âm vị trong ngôn ngữ. Các yếu tố chi tiết:
   - ***Labeling:*** Gán nhãn cho từng diphone: phoneme đầu, phoneme cuối, duration, pitch.
   - ***Recording consistency:*** Thu âm với cùng giọng, cùng tốc độ, cùng cường độ để giảm biến thiên giữa các diphone.
   - ***Contextual prosody:*** Ghi chú prosody cơ bản để hỗ trợ chỉnh sửa pitch/duration khi ghép nối.
2. **Text preprocessing:** Giống như Unit Selection, text input được chuẩn hóa, tokenized, và chuyển sang phoneme-level qua G2P. Sau đó sinh ra chuỗi diphone cần ghép:
   $$\text{Text} \xrightarrow{\text{G2P}} \text{Phoneme sequence} \xrightarrow{\text{Diphone mapping}} \text{Diphone sequence}$$
3. **Lựa chọn và nối diphone:**
   - Chọn diphone tương ứng từ database cho mỗi cặp âm vị liên tiếp trong chuỗi phoneme.
   - Do diphone đã chứa phần chuyển tiếp, độ trôi chảy của câu cao hơn phoneme-level.
   - Ghép nối waveform của các diphone. Đôi khi áp dụng kỹ thuật cross-fade hoặc envelope shaping để giảm click/pop.
4. **Prosody modeling và post-processing:**
   - ***Pitch adjustment:*** Rule-based theo dấu câu, trọng âm từ G2P hoặc các quy tắc ngữ điệu.
   - ***Duration adjustment:*** Tăng/giảm thời lượng diphone để câu nghe tự nhiên hơn.
   - ***Amplitude normalization:*** Điều chỉnh cường độ để thống nhất âm lượng giữa các diphone.

Cách giải quyết nhược điểm [2]

- ***Rời rạc:*** Sử dụng diphone chứa phần chuyển tiếp giữa hai âm vị để giảm cảm giác rời rạc so với phoneme-level.
- ***Prosody hạn chế:*** Rule-based prosody modeling dựa trên ngữ cảnh, dấu câu, và trọng âm từ G2P.
- ***Mở rộng giọng mới:*** Cần ghi âm lại bộ diphone mới, nhưng số lượng unit ít hơn nhiều so với Unit Selection toàn bộ phoneme, giúp tiết kiệm công sức.

Ưu/nhược điểm và ứng dụng [2]

✓ **Ưu điểm:** Âm thanh trôi chảy hơn phoneme-level, tốc độ cao, tiêu tốn ít tài nguyên.

✓ **Ưu điểm:** Dễ triển khai đa ngôn ngữ nếu database diphone đầy đủ.

× **Nhược điểm:** Prosody tự nhiên vẫn còn hạn chế, khó tạo biểu cảm phong phú.

× **Nhược điểm:** Mở rộng giọng mới yêu cầu thu âm diphone, hạn chế so với neural TTS.

Ứng dụng thực tế [2]

- Hệ thống nhúng, đọc văn bản trên các thiết bị có bộ nhớ hạn chế.
- Thông báo trên phương tiện công cộng, hướng dẫn đơn giản, trợ lý giọng nói cơ bản.
- Trường hợp yêu cầu tốc độ cao, độ trôi chảy chấp nhận được nhưng không cần biểu cảm phong phú.

2.1.4 Phương pháp 3: Domain-specific Unit Concatenation

Kiến trúc và pipeline chi tiết Phương pháp này tập trung vào các ứng dụng chuyên biệt, ví dụ như đọc tin tức, hệ thống GPS, hướng dẫn thoại, nơi từ vựng và cấu trúc câu có hạn. Pipeline chi tiết:

1. **Database domain-specific:**
   - Chỉ thu âm các từ, cụm từ phổ biến trong domain.
   - Gán nhãn phoneme, prosody cơ bản, duration, pitch.
   - Ưu điểm: số lượng unit ít, tiết kiệm bộ nhớ, dễ bảo trì.
   - Nhược điểm: không thể đọc toàn bộ ngôn ngữ, cần database riêng cho từng domain.
2. **Text preprocessing và mapping:**
   - Chuẩn hóa text, tokenization, G2P.
   - Map từ/phrase trong input sang các unit trong domain database.
   - Sử dụng ***lookup table*** thay cho cost function phức tạp, giúp tăng tốc và giảm độ trễ.
3. **Unit selection và concatenation:**
   - Lựa chọn unit dựa trên ngữ cảnh domain và bảng tra cứu.
   - Ghép nối waveform của các unit theo thứ tự câu.
   - Áp dụng rule-based prosody: điều chỉnh pitch và duration dựa trên dấu câu, trọng âm từ domain.
4. **Post-processing:**
   - ***Amplitude normalization:*** đồng bộ âm lượng giữa các unit.
   - ***Fade-in/Fade-out:*** giảm click/pop khi nối unit.
   - Tùy chỉnh prosody theo ngữ cảnh domain để câu nghe tự nhiên hơn.

Cách giải quyết nhược điểm [2]

- Giảm độ phức tạp: Chỉ tập trung vào từ/cụm từ phổ biến, không cần tối ưu cost function tổng quát.
- Prosody rule-based: Rule tùy biến theo kiểu câu thường gặp trong domain, ví dụ câu hướng dẫn GPS thường nhấn nhá vào các điểm địa danh.
- Dễ bảo trì: Chỉ cần cập nhật database khi domain thay đổi, không ảnh hưởng đến toàn bộ hệ thống.

Ưu/nhược điểm và ứng dụng [2]

✓ **Ưu điểm:** Tốc độ cực nhanh, độ trễ gần như bằng 0, tài nguyên thấp.

✓ **Ưu điểm:** Âm thanh tự nhiên hơn trong domain, dễ bảo trì và cập nhật database.

× **Nhược điểm:** Không tổng quát cho toàn ngôn ngữ, không phù hợp cho giọng đọc tự do ngoài domain.

× **Nhược điểm:** Cần database riêng cho từng domain, khó tái sử dụng nếu mở rộng ứng dụng.

Ứng dụng thực tế [2]

- GPS, hướng dẫn thoại, thông báo công cộng, đọc tin tức domain-specific.
- Ứng dụng nhúng cần tốc độ cực nhanh và tài nguyên hạn chế.
- Trường hợp người dùng không cần biểu cảm phong phú mà ưu tiên hiệu suất.

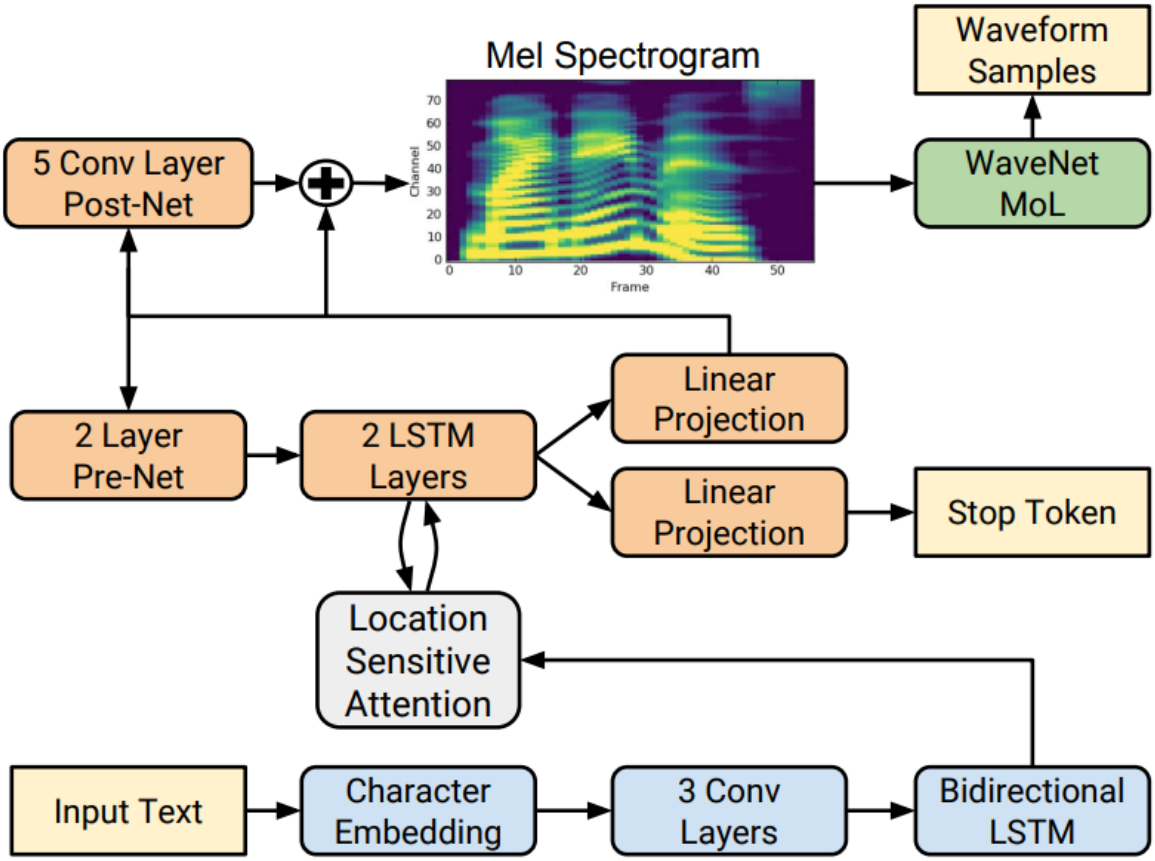
2.1.5 So sánh chung và pipeline tối ưu hóa

- **Unit Selection:** Trôi chảy, độ chính xác từ vựng cao, chi phí trung bình.
- **Diphone:** Trôi chảy hơn phoneme-level, ít đơn vị hơn nhưng prosody hạn chế.
- **Domain-specific:** Tốc độ cực nhanh, tối ưu cho domain nhưng ít linh hoạt.
- **Các chiến lược tối ưu:**
  - Tối ưu ***cost function*** hoặc lookup table và rule-based prosody cho domain.
  - Tạo database đa phong cách hoặc đa ngôn ngữ để mở rộng.
  - Áp dụng pitch/duration adjustment để giảm rời rạc, tăng tự nhiên trong domain.

2.2 Level 2: Neural / Parametric Deep Learning

Level 2 TTS sử dụng học sâu để mô hình hóa mối quan hệ giữa văn bản và đặc trưng âm học, cho phép tạo giọng tự nhiên hơn Level 1. Các hệ thống thường gồm hai thành phần chính: ***Acoustic Model*** và ***Vocoder***. Dưới đây là ba phương pháp điển hình.

2.2.1 Phương pháp 1: Tacotron2



Hình 2.1: Kiến trúc Tacotron2[7]

Kiến trúc và pipeline chi tiết

1. **Tiền xử lý văn bản (Text Preprocessing):** - Chuẩn hóa văn bản (text normalization): chuyển chữ viết tắt, số, ký hiệu về dạng chuẩn. - Tokenization: tách câu, từ hoặc phoneme. - G2P (Grapheme-to-Phoneme): ánh xạ ký tự sang âm vị (phoneme). Kết quả là một sequence phoneme có thể được encoder xử lý.

2. **Encoder:** - Sử dụng Bi-directional LSTM để mã hóa sequence phoneme thành *contextual embeddings*. - Embeddings chứa thông tin về âm vị, vị trí trong câu, prosody cơ bản (pitch, duration). - Đây là bước trích xuất tính năng quan trọng để mô hình học được mối liên hệ text → âm thanh.

3. **Attention Mechanism:** - Học trọng số ánh xạ encoder output → decoder input. - Giúp decoder biết thời điểm sinh frame Mel tương ứng với từng phoneme. - Cơ chế attention giảm lỗi alignment giữa text và âm thanh.

4. **Decoder:** - LSTM autoregressive sinh Mel-spectrogram frame theo thứ tự. - Prenet (2 fully-connected layers + dropout) trước decoder để tạo embedding mềm, ổn định. - Postnet (convolution layers) hiệu chỉnh Mel-spectrogram, thêm chi tiết tần số cao.

5. **Vocoder:** - WaveNet hoặc HiFi-GAN chuyển Mel-spectrogram thành waveform chất lượng cao. - Vocoder học mapping từ Mel feature → time-domain waveform, tạo ra giọng mượt, tự nhiên.

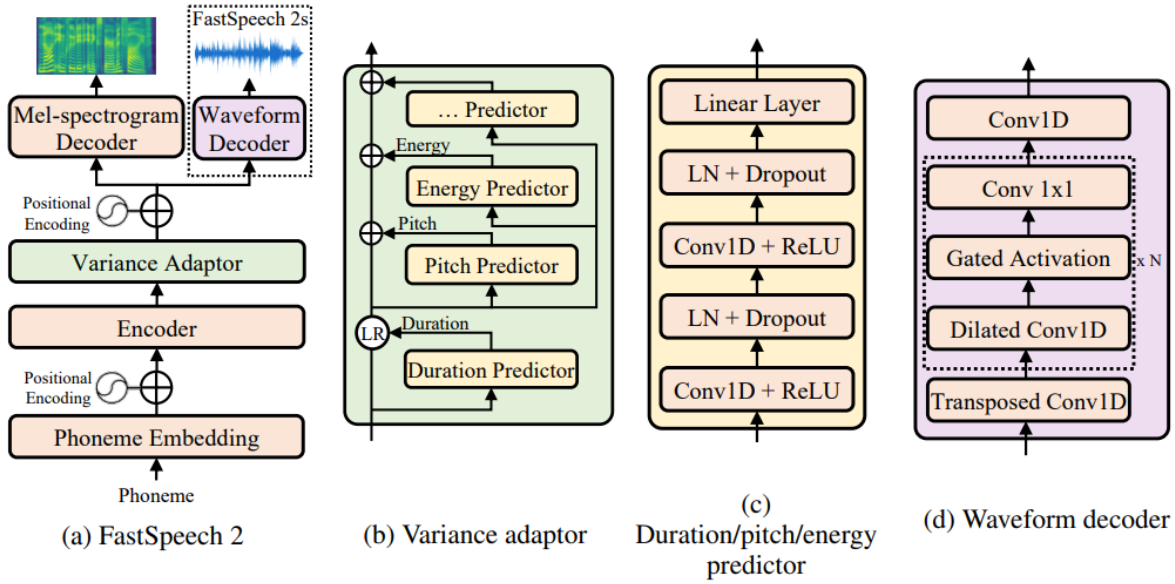
Cách giải quyết nhược điểm

- **Inference chậm:** autoregressive → sử dụng teacher forcing, scheduled sampling trong training để ổn định.
- **Giọng chưa tự nhiên:** thêm prosody tags (pitch, energy, duration) vào input.
- **Cá nhân hóa giọng:** fine-tune model với dữ liệu giọng người dùng.

Ứng dụng

- Voice assistant, đọc sách tự động, prototyping TTS.
- Khi cần giọng tự nhiên, mượt mà, và pipeline ổn định, nhưng tốc độ không quá quan trọng.

2.2.2 Phương pháp 2: FastSpeech2



Hình 2.2: Kiến trúc FastSpeech2[6]

Kiến trúc và pipeline chi tiết FastSpeech2 là model Transformer-based, giải quyết vấn đề tốc độ của Tacotron2 nhờ parallel decoding:

1. **Text Preprocessing:** Chuẩn hóa văn bản, tokenization, G2P → embeddings phoneme.
2. **Encoder:** Transformer encoder tạo contextual embeddings, capture thông tin ngữ cảnh dài hạn.

3. **Duration Predictor:** - Dự đoán số frame Mel cho mỗi phoneme. - Duration giúp giải quyết vấn đề alignment, không cần attention autoregressive. - Cho phép parallel decoding.

4. **Decoder:** Transformer decoder sinh Mel-spectrogram cho toàn bộ câu đồng thời. - Parallel decoding giảm inference time, tốc độ cao.

5. **Prosody Control:** Pitch, energy, duration embeddings thêm vào decoder để cải thiện tự nhiên và biểu cảm.

6. **Vocoder:** HiFi-GAN hoặc Parallel WaveGAN chuyển Mel-spectrogram thành waveform chất lượng cao, inference nhanh.

Cách giải quyết nhược điểm

- Parallel decoding giảm inference time so với Tacotron2.
- Prosody embeddings giúp biểu cảm giọng tự nhiên hơn.
- Transfer Learning: fine-tune theo giọng cá nhân với dataset nhỏ.
- Model compression: quantization, pruning, knowledge distillation để tiết kiệm GPU/CPU.

Ứng dụng

- Voice assistant, text reading, hệ thống TTS real-time.
- Khi cần giọng tự nhiên + tốc độ inference cao.

2.2.3 Phương pháp 3: VITS / Conformer-based TTS

Kiến trúc và pipeline chi tiết [2] VITS là end-to-end model kết hợp VAE và Flow-based models:

1. **Text Preprocessing:** chuẩn hóa, tokenization, G2P.

2. **Conformer Encoder:** kết hợp convolution + transformer, mã hóa phoneme thành latent vector.

3. **Posterior Encoder:** học phân phối latent từ Mel-spectrogram thật, capture prosody tự nhiên.

4. **Decoder / Vocoder:** Flow-based decoder sinh waveform trực tiếp từ latent vector, end-to-end.

5. **Prosody Control:** Latent prosody vector cho phép zero/few-shot style transfer, tạo biểu cảm phong phú.

Cách giải quyết nhược điểm

- End-to-end giảm phụ thuộc vocoder, tăng chất lượng tự nhiên.
- Latent prosody vector giúp zero-shot cloning, biểu cảm đa dạng.
- Knowledge distillation, quantization giảm chi phí inference.
- Fine-tune theo giọng cá nhân với dataset nhỏ.

Ứng dụng

- Voice cloning, đọc sách biểu cảm, dubbing, tạo voice avatar.
- Khi cần giọng tự nhiên, biểu cảm đa dạng, zero/few-shot voice cloning.

2.2.4 So sánh và chiến lược tối ưu hóa pipeline

- Tacotron2: giọng tự nhiên, tốt cho prototyping, inference chậm hơn FastSpeech2.
- FastSpeech2: parallel decoding, tốc độ cao, prosody ổn định, dễ fine-tune cá nhân hóa.
- VITS / Conformer-based: end-to-end, prosody phong phú, zero-shot, phù hợp voice cloning và ứng dụng biểu cảm.
- Chiến lược tối ưu:
  - Quantization, pruning, knowledge distillation giảm chi phí inference.
  - Streaming inference: chia Mel-spectrogram thành chunk, sinh âm thanh real-time.
  - Transfer learning: fine-tune giọng cá nhân với dataset nhỏ, giữ chất lượng giọng.

2.3 Level 3: Few-shot / Generative Large Models

Level 3 TTS là lớp công nghệ tiên tiến nhất, dựa trên các mô hình autoregressive, diffusion, hoặc transformer-based lớn, coi TTS như bài toán ***Language Modeling*** cho audio. Điểm nổi bật là khả năng *zero-shot* hoặc *few-shot voice cloning*, nghĩa là chỉ cần vài giây mẫu giọng của người dùng là có thể tái tạo giọng với ngữ điệu và biểu cảm gần giống thật.

2.3.1 Phương pháp 1: VALL-E

Kiến trúc và pipeline chi tiết [9]

1. **Input:** - Văn bản (text) cần chuyển sang giọng nói. - Một vài giây audio mẫu của giọng người dùng.
2. **Voice Encoder:** - Nhận audio mẫu → trích xuất ***voice embedding***. - Embedding lưu trữ đặc trưng cá nhân: tần số cơ bản, ngữ điệu, phong cách biểu cảm. - Encoder có thể dùng CNN + Transformer layers để capture cả đặc trưng tần số và temporal của giọng.

3. **Text Encoder:** - Mã hóa văn bản (phoneme hoặc subword) thành embeddings, giữ thông tin ngữ cảnh. - Sử dụng Transformer layers để nắm mối liên hệ dài hạn giữa các âm vị.

4. **Decoder / Acoustic Token Predictor:** - Dự đoán dãy ***acoustic tokens*** autoregressive dựa trên voice embedding + text embedding. - Acoustic token là đại diện rời rạc của Mel-spectrogram frame hoặc latent audio unit. - Cơ chế attention giúp kết hợp voice embedding với text embedding, đảm bảo giọng và ngữ điệu giống mẫu.

5. **Vocoder:** - HiFi-GAN, WaveNet, hoặc diffusion vocoder tái tạo waveform từ dãy acoustic token. - Tạo giọng mượt, tự nhiên, giữ đặc trưng cá nhân của mẫu.

Cách giải quyết nhược điểm

- **Inference chậm:** sử dụng caching embeddings, chunk-based decoding để sinh audio theo block, giảm latency.
- **Hallucination / sai prosody:** prompt engineering, chọn audio representative, attention masking để giảm lỗi alignment text audio.
- **Tài nguyên cao:** model compression bằng distillation, pruning, hoặc quantization.
- **Cá nhân hóa:** few-shot embedding giúp clone giọng với dataset cực nhỏ (1–5s audio).

Ứng dụng

- Voice cloning, tạo voice avatar, dubbing phim, đọc sách cá nhân hóa.
- Khi cần giọng biểu cảm, gần giống người thật, và khả năng zero/few-shot.

2.3.2 Phương pháp 2: YourTTS

Kiến trúc và pipeline chi tiết [1]

1. **Input:** text + audio mẫu (có thể 1–3s).

2. **Multi-speaker TTS Encoder:** - Dùng pretrained speaker encoder (ECAPA-TDNN hoặc Conformer-based) để tạo embedding giọng. - Embedding chứa thông tin tần số cơ bản, prosody, phong cách phát âm.

3. **Text-to-Mel Model:** - Transformer hoặc Conformer-based decoder dự đoán Mel-spectrogram frame từ text embedding + voice embedding. - Duration predictor và attention mechanism giúp mapping alignment chính xác giữa text và audio.

4. **Vocoder:** - HiFi-GAN hoặc diffusion vocoder, tái tạo waveform. - End-to-end latent prosody giúp zero/few-shot style transfer.

Cách giải quyết nhược điểm

- Sử dụng latent prosody vector để tránh hallucination và tăng tự nhiên.
- Fine-tune nhẹ với audio mẫu để nâng độ chính xác voice cloning.
- Knowledge distillation và pruning giúp giảm yêu cầu GPU.
- Attention + duration predictor giúp giữ alignment text audio, tăng prosody tự nhiên.

Ứng dụng

- Voice cloning đa ngôn ngữ, biểu cảm, đọc sách, dubbing, tạo AI assistant giọng cá nhân hóa.
- Phù hợp khi cần tạo nội dung audio nhanh với đặc trưng giọng cá nhân.

2.3.3 Phương pháp 3: Diffusion-based TTS

Kiến trúc và pipeline chi tiết [5]

1. **Input:** Text + vài giây audio mẫu (voice embedding).
2. **Diffusion Model:** - Sinh Mel-spectrogram bằng quá trình denoising từ Gaussian noise đến tín hiệu mong muốn. - Conditioning trên voice embedding và text embedding.
3. **Prosody Control:** - Prosody latent vector được trích xuất từ mẫu giúp mô hình tái tạo biểu cảm, ngữ điệu. - Zero-shot style transfer cho phép clone giọng với dữ liệu cực ít.
4. **Vocoder:** Diffusion vocoder hoặc HiFi-GAN end-to-end tái tạo waveform.

Cách giải quyết nhược điểm

- Denoising step-wise giúp giảm hallucination.
- Latent prosody embedding + attention alignment giúp tự nhiên hóa speech.
- Knowledge distillation, pruning giảm GPU/CPU usage.
- Chunk-based streaming inference tăng tốc độ, giảm latency.

Ứng dụng

- Tạo giọng tự nhiên, biểu cảm, multi-style, multi-speaker.
- Zero/few-shot voice cloning, voice avatars, dubbing phim, đọc sách cá nhân hóa.

2.3.4 So sánh và chiến lược tối ưu pipeline

- **VALL-E:** zero/few-shot, autoregressive, giọng tự nhiên, inference chậm.
- **YourTTS:** Conformer-based, prosody phong phú, zero/few-shot, tốc độ trung bình, end-to-end.
- **Diffusion TTS:** biểu cảm đa dạng, stable prosody, inference chậm hơn, GPU intensive.
- Chiến lược tối ưu:
  - ***Prompt Engineering:*** chọn mẫu audio representative để giảm hallucination.
  - ***Model Compression:*** knowledge distillation, pruning, quantization để giảm tài nguyên.
  - ***Adaptive decoding:*** duration + attention control, chunk-based streaming inference để tăng tốc và ổn định prosody.
  - Transfer learning: fine-tune với vài giây audio để cá nhân hóa giọng.

Chương 3

Ma trận đánh giá và chiến lược tối ưu

3.1 So sánh đặc tính kỹ thuật và ứng dụng

Bảng 3.1: So sánh kỹ thuật, hiệu năng và ứng dụng của ba hướng TTS

Tiêu chí	**Level 1: Concatenative / Rule-based**	**Level 2: Neural / Parametric DL**	**Level 3: Few-shot / Generative Large Models**
Latency	Rất thấp (<20ms), phù hợp realtime, nhúng	Trung bình (50–200ms), inference nhanh với parallel decoding	Cao (>1s), model lớn, autoregressive/diffusion
Resource	CPU/Embedded, ít RAM, database nhỏ	CPU/GPU, memory trung bình, dataset lớn	GPU mạnh hoặc multi-GPU, memory cao, storage lớn cho model
Naturalness	Thấp, giọng rời rạc, prosody hạn chế	Cao, mượt mà, prosody tốt, biểu cảm cơ bản	Rất cao, biểu cảm phong phú, phong cách gần thật
Prosody Control	Rule-based cơ bản	Fine-tune hoặc duration/pitch predictor	Embedding-based, style
emotion transfer			
Multi-lingual	Dễ mở rộng, chỉ cần database	Khó hơn, cần dataset ngôn ngữ	Có thể hỗ trợ nhờ pretraining đa ngôn ngữ
Personalization	Rất hạn chế	Trung bình (fine-tune theo cá nhân)	Zero/Few-shot, chỉ cần vài giây audio
Inference Cost	Rất thấp, CPU đủ	Trung bình, GPU đề xuất để tốc độ tốt	Cao, GPU-intensive, cần tối ưu hóa
Use-case	IoT, Smart Home, đọc văn bản đơn giản	Virtual Assistant, TTS cá nhân hóa, audiobook	Content Creation, Voice Cloning, dubbing, voice avatars

3.2 Chiến lược tối ưu pipeline cho từng level

3.2.1 Level 1: Concatenative / Rule-based

- Tối ưu ***cost function*** để giảm rời rạc khi nối các đơn vị.
- Rule-based prosody: pitch/duration adjustment theo câu, dấu câu, hoặc ngữ cảnh.
- Xây dựng database đa ngôn ngữ hoặc đa phong cách để tăng tính linh hoạt.
- Có thể kết hợp simple post-processing filter (fade-in/fade-out, smoothing) để giọng mượt hơn.

3.2.2 Level 2: Neural / Parametric Deep Learning

- Transformer-based acoustic model: FastSpeech2, Conformer, Tacotron2 encoder-decoder.
- Parallel vocoder (HiFi-GAN, Parallel WaveGAN) để tăng tốc độ inference.
- Fine-tune giọng cá nhân với dataset nhỏ để personalization.
- Model compression: pruning, quantization, knowledge distillation giảm inference cost và memory.
- Streaming inference: chia Mel-spectrogram thành chunk, sinh audio theo thời gian thực.

3.2.3 Level 3: Few-shot / Generative Large Models

- Embedding voice representation (voice encoder) để zero/few-shot voice cloning.
- Autoregressive / diffusion decoder kết hợp attention, duration control để tạo audio trôi chảy.
- Prompt engineering: chọn audio mẫu representative để giảm hallucination.
- Model compression: distillation, pruning, quantization để giảm GPU requirement.
- Adaptive decoding & chunk-based streaming inference để tăng tốc, giảm latency, ổn định prosody.
- Transfer learning: fine-tune nhẹ với vài giây audio để cá nhân hóa giọng.
- **Level 1:** - Ưu điểm: tốc độ cực nhanh, độ trễ thấp, tài nguyên yêu cầu thấp, dễ triển khai đa ngôn ngữ. - Nhược điểm: giọng rời rạc, prosody hạn chế, khó mở rộng giọng mới. - Ứng dụng: IoT, Smart Home, đọc văn bản đơn giản, thiết bị nhúng.
- **Level 2:** - Ưu điểm: giọng tự nhiên, prosody mượt, hỗ trợ fine-tune cá nhân hóa, pipeline ổn định. - Nhược điểm: cần GPU và dataset lớn, khó mở rộng đa ngôn ngữ nếu không có dữ liệu. - Ứng dụng: trợ lý ảo, TTS cá nhân hóa, audiobook, interactive voice applications.

- **Level 3:** - Ưu điểm: biểu cảm tự nhiên, hỗ trợ zero/few-shot, clone giọng từ vài giây mẫu, đa phong cách và cảm xúc. - Nhược điểm: inference chậm, GPU-intensive, hallucination có thể xảy ra, mô hình phức tạp. - Ứng dụng: sáng tạo nội dung, dubbing phim, voice avatars, voice cloning, multi-speaker TTS.

Tương lai TTS: Phát triển TTS hướng đến cân bằng giữa chất lượng âm thanh, biểu cảm, đa ngôn ngữ và chi phí tính toán. Kết hợp tối ưu pipeline, model compression, streaming inference, và phần cứng chuyên dụng là chìa khóa để đạt hiệu năng tối ưu cho từng ứng dụng cụ thể.

Chương 4

Kết luận và Hướng phát triển

4.1 Tổng kết các kết quả nghiên cứu

Báo cáo đã trình bày một cái nhìn toàn cảnh và có hệ thống về sự tiến hóa của bài toán ***Tổng hợp tiếng nói (Text-to-Speech)***. Quá trình phát triển này không đơn thuần là sự cải tiến về mặt kỹ thuật, mà là sự thay đổi căn bản trong tư duy ***mô hình hóa toán học*** (mathematical modeling paradigm):

- **Từ Rời rạc đến Liên tục:** Sự chuyển dịch từ các phương pháp ghép nối đơn vị (Concatenative - Level 1) hoạt động trên không gian trạng thái rời rạc, sang các mô hình tham số (Parametric - Level 2) hoạt động trên không gian vector liên tục.
- **Từ Định định đến Ngẫu nhiên:** Sự chuyển dịch từ các ánh xạ $1-1$ tất định (Deterministic mapping) như FastSpeech, sang các mô hình xác suất (Probabilistic models) như Diffusion/Flow-based (Level 3) để giải quyết bài toán ánh xạ $1-n$ (One-to-Many problem) của ngữ điệu con người.

Dựa trên ma trận đánh giá hiệu năng, chúng ta rút ra các kết luận cốt lõi sau:

1. **Về Hiệu năng (Performance):** Các mô hình ghép nối (Level 1) và mô hình tham số phi tự hồi quy (Non-autoregressive Level 2) vẫn chiếm ưu thế tuyệt đối về tốc độ thực thi và độ trễ thấp, là lựa chọn tối ưu cho các hệ thống nhúng hoặc ứng dụng thời gian thực (Real-time).
2. **Về Chất lượng và Biểu cảm (Fidelity & Expressiveness):** Các mô hình sinh quy mô lớn (Large Generative Models - Level 3) đã vượt qua "thung lũng kỳ lạ" (uncanny valley), đạt được độ tự nhiên tiệm cận giọng người thật nhờ khả năng học biểu diễn không giám sát (unsupervised representation learning) trên lượng dữ liệu khổng lồ.
3. **Về Khả năng Tổng quát hóa (Generalization):** Cơ chế *Zero-shot Voice Cloning* đã chứng minh tính hiệu quả của việc tách biệt và tái tổ hợp các đặc trưng nội dung (content content) và đặc trưng người nói (speaker identity) trong không gian tiềm ẩn.

4.2 Thách thức và Định hướng nghiên cứu tương lai

Dù đạt được những thành tựu đáng kể, lĩnh vực TTS vẫn đối mặt với những bài toán mở cần lời giải từ góc độ thuật toán và toán học:

Tối ưu hóa đa mục tiêu (Multi-objective Optimization) Thách thức nằm ở việc tìm điểm cân bằng Pareto giữa ba yếu tố: Chất lượng âm thanh cao — Tốc độ suy luận nhanh — Chi phí tính toán thấp. Hướng tiếp cận tiềm năng là áp dụng các kỹ thuật *Lượng tử hóa vector (Vector Quantization)* và *Chưng cất tri thức (Knowledge Distillation)* để nén các mô hình lớn (Level 3) xuống kích thước khả thi cho thiết bị biên.

Kiểm soát ngữ điệu hạt mịn (Fine-grained Prosody Control) Các mô hình hiện tại chủ yếu học ngữ điệu một cách ngầm định (implicit). Hướng nghiên cứu tương lai cần tập trung vào việc mô hình hóa tường minh (explicit modeling) các yếu tố như cảm xúc, nhấn trọng âm, và phong cách đọc thông qua các biến tiềm ẩn có cấu trúc (structured latent variables).

Đạo đức AI và Bảo mật (AI Ethics & Security) Với khả năng sao chép giọng nói chính xác, bài toán phân biệt giọng thật/giả (Anti-spoofing) và đánh dấu thủy vân (Audio Watermarking) trở thành yêu cầu bắt buộc để ngăn chặn việc lạm dụng công nghệ Deepfake.

4.3 Lời kết

Tóm lại, bài toán Text-to-Speech đã chuyển mình từ việc "máy đọc chữ" sang "máy tạo sinh tiếng nói biểu cảm". Sự kết hợp giữa lý thuyết Xác suất thống kê, Xử lý tín hiệu số và Học sâu hiện đại đang mở ra kỷ nguyên mới của giao tiếp người-máy tự nhiên, xóa nhòa ranh giới giữa thực và ảo.

Tài liệu tham khảo

[1] Edresson Casanova, Julian Weber, Christopher Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir Antonelli Ponti. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. *arXiv preprint arXiv:2112.02418*, 2021.

[2] ChatGPT and Gemini.

[3] Andrew J Hunt and Alan W Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 373–376, 1996.

[4] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis, 2020.

[5] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *Proceedings of the 38th International Conference on Machine Learning (ICML) – PMLR*, volume 139, pages 8599–8608, 2021.

[6] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech, 2022.

[7] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions, 2018.

[8] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016.

[9] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers, 2023.

[10] Quan Zhou, Jianhua Shan, Wenlong Ding, Chengyin Wang, Shi Yuan, Fuchun Sun, Haiyuan Li, and Bin Fang. Cough recognition based on mel-spectrogram and convolutional neural network. *Frontiers in Robotics and AI*, Volume 8 - 2021, 2021.