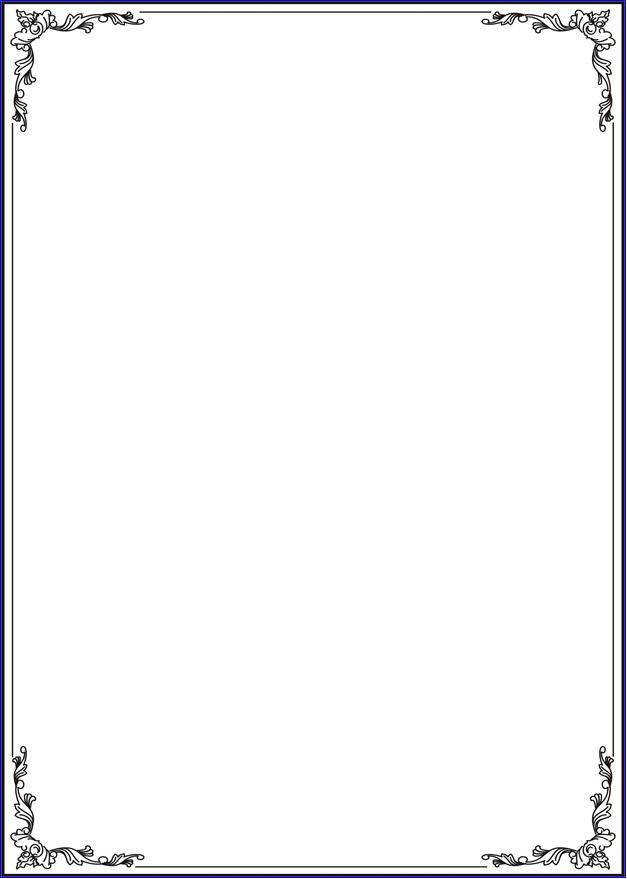
** BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO KẾT QUẢ NGHIỆM THU CỦA QUÁ TRÌNH NGHIÊN CỨU KHOA HỌC SINH VIÊN**

**ĐỀ TÀI: Achieving Text-to-Speech Capabilities Comparable to Human Levels via Style Diffusion and Adversarial Training Utilizing Extensive Speech Language Models.**

**GVHD: Th.S. Phạm Thị Miên**

TP.Hồ Chí Minh , ngày 12 tháng 05 năm 2024

**TABLE OF CONTENT**

[Achieving Text-to-Speech Capabilities Comparable to Human Levels via Style Diffusion and Adversarial Training Utilizing Extensive Speech Language Models. 3](#_Toc18118)

[LỜI NÓI ĐẦU 4](#_Toc29112)

[ABSTRACT 6](#_Toc19393)

[I. INTRODUCTION 7](#_Toc18229)

[II. RELATED WORK 10](#_Toc10655)

[III. METHODOLOGY 11](#_Toc28603)

[1. Baseline Model: 11](#_Toc21064)

[2. Improve method: 14](#_Toc21807)

[IV. EXPERIMENT AND RESULT 16](#_Toc12608)

[1. Datasets: 16](#_Toc367)

[2. Data pre-processing: 18](#_Toc27567)

[3. Implementation and Metrics: 18](#_Toc17594)

[4. Result: 24](#_Toc14758)

[5. Future work: 26](#_Toc17510)

[V. IMPROVE THE FUTURE WORK 27](#_Toc27276)

[VI. CONCLUSION 28](#_Toc23581)

[VII. REFERENCE 29](#_Toc30157)

**LỜI NÓI ĐẦU**

Sáng tạo và đổi mới là quy tắc mà thế giới này vận hành, vì vậy để tránh không bị xã hội đào thải, chúng ta phải luôn học tập nâng cao kỹ năng, không ngừng cải tiến tư duy trong và ngoài lĩnh vực mà chúng ta đang tìm hiểu như một giải pháp để ta có thể nâng cao năng lực trong bản thân. Nghiên cứu tìm tòi học hỏi có lẽ là một quá trình lâu dài gắn liền với cuộc đời của mỗi con người chúng ta. Những đại dương sâu thẩm, những căn bệnh siêu hiếm, một vũ trụ bí ẩn và muôn vàn thứ khác xung quanh chúng ta, hàng tỷ tỷ điều chúng ta cần tìm hiểu và sau cùng những thứ ấy luôn đời chúng ta nghiên cứu và khám phá. Nói như vậy để thấy được sự quan trọng và lớn lao mà nghiên cứu khoa học đem đến. Nó đã, đang và sẽ thay đổi và thúc đẩy nền văn minh của nhân loại The World is flat của Thomas Friedman mà một trong những minh chứng rõ ràng cho việc nghiên cứu khoa học đã tác động trực tiếp vào thế giới quan của con người – Quyển sách như một lời khẳng định rằng chúng ta đã và đang sống trong một kỷ nguyên mới, một kỷ nguyên mà con người có thể xích lại gần nhau hơn, một kỷ nguyên của thông tin và số. Như thế bằng cách nhìn tổng quan trong sự vận động, phát triển, đổi mới và đào thải của thế giới có thể thấy công nghệ thông tin hiện đang là ngành có ảnh hưởng trực tiếp vào trong đời sống cũng như công việc hàng ngày của chúng ta

Như vậy việc nghiên cứu khoa học trong lĩnh vực Công nghệ thông tin là một việc làm hết sức cần thiết và có ý nghĩa to lớn. Với những nghiên cứu đi đôi với ứng dụng tốt sẽ làm nên những thay đổi cho cuộc sống của con người, đẩy mạnh và bắt kịp quá trình thông tin hóa đang diễn ra hiện nay của thế giới. Bài luận này được nhóm chúng tôi viết nhằm tổng kết, đúc kết cũng như thực hành việc nghiên cứu dựa trên lý thuyết được truyền giảng trên lớp và quá trình học hỏi, tìm tòi bán chuyên sâu của chúng tôi ngoài thực tế. Qua đây, Tôi xin dành lời cảm ơn đặc biệt đến Th.S Phạm Thị Miên đã có những ý kiến quan tâm, đóng góp luận điểm cá nhân và đề xuất ý tưởng cải tiến đến đề tài nghiên cứu của nhóm chúng tôi. Nội dung bài luận ngoài việc tham khảo tài liệu, nội dung hoàn toàn mang quan điểm của cá nhân, viết theo văn phong cá nhân. Do đó sẽ không thể tránh khỏi những sai sót không muốn. Rất mong quý hội đồng nghiên cứu tích cực nhận xét để cải thiện.

Đại diện nhóm sinh viên thực hiện

Trưởng nhóm

**ABSTRACT**

Dự án Voice Clone mà chúng tôi triển khai đề xuất một phương pháp tiên tiến trong lĩnh vực sao chép giọng nói, kết hợp các kỹ thuật học máy và xử lý tín hiệu âm thanh để tạo ra một hệ thống có khả năng sao chép giọng nói chất lượng cao và linh hoạt. Mục tiêu chính của nghiên cứu là phát triển một mô hình giọng nói đa chiều sâu sắc, có khả năng tái tạo cả ngữ intonation và đặc trưng cá nhân của người mẫu giọng nói. Chúng tôi tiến hành quá trình thu thập dữ liệu mẫu giọng nói đa dạng và phong phú để huấn luyện mô hình. Sự kết hợp của mạng học sâu và các thuật toán học máy tiên tiến được sử dụng để tối ưu hóa quá trình học và tạo ra một biểu diễn biểu diễn giọng nói chất lượng cao.

Chúng tôi cũng đặt sự chú ý đặc biệt vào khía cạnh triển khai của hệ thống, tập trung vào việc giảm độ phức tạp triển khai và đảm bảo tích hợp linh hoạt vào nhiều ứng dụng. Điều này đòi hỏi sự tối ưu hóa cả về hiệu suất và tài nguyên để đảm bảo khả năng sử dụng rộng rãi trong các hệ thống thực tế.

Ứng dụng của Voice Clone có thể đặt ra trong nhiều lĩnh vực, bao gồm giả mạo giọng nói trong các ứng dụng trí tuệ nhân tạo, gọi điện thoại tự động và thậm chí là tạo ra các nhân vật giọng nói độc đáo trong lĩnh vực giải trí đa phương tiện. Chúng tôi kỳ vọng rằng nghiên cứu của chúng tôi sẽ mang lại đóng góp quan trọng cho việc phát triển các công nghệ sao chép giọng nói và mở ra những khả năng mới trong tương tác giọng nói máy người.

1. **INTRODUCTION**

Chuyển văn bản thành giọng nói (TTS), còn được biết đến với tên gọi tổng hợp giọng nói, đã đạt được tiến triển đáng kể với sự xuất hiện của học sâu, tạo ra giọng tổng hợp ngày càng giống giọng người [1], [2], [3]. Tuy nhiên, thách thức vẫn là việc tạo ra giọng nói biểu cảm, bao gồm toàn bộ phổ âm, thời gian và đặc trưng phổ, còn được biết đến là thông tin về ngôn ngữ [4]. Việc tổng hợp giọng nói với các biến thể được ảnh hưởng bởi ngữ cảnh, tông điệu cảm xúc và thói quen ngôn ngữ của người nói là một nhiệm vụ phức tạp, biến TTS thành một vấn đề ánh xạ một-nhiều đòi hỏi các phương pháp sáng tạo. Nhiều chiến lược đã được đề xuất để giải quyết thách thức này, bao gồm các phương pháp dựa trên suy luận biến thiên [1], [5], [6], [7], mô hình dựa trên dòng chảy [1], [8], [9], kiểm soát âm, thời gian và năng lượng [10], [11], và sử dụng bộ mã hóa ngôn ngữ bên ngoại [12], [13], [14]. Mặc dù có những nỗ lực này, sản xuất giọng nói tổng hợp vẫn chưa thực sự đạt được giọng nói người thật, đặc biệt là trong việc mô hình chính xác các phong cách nói và tông điệu cảm xúc của các người nói khác nhau.

Nhiều nỗ lực đã được thực hiện để tích hợp thông tin về phong cách vào mô hình TTS [12], [13], [15], [16]. Những cách tiếp cận này chủ yếu dựa trên các mô hình tự hồi quy như Tacotron. Ngược lại, các mô hình TTS song song không tự hồi quy như Fastspeech [17] và Glow-TTS [9] có nhiều ưu điểm hơn so với các mô hình tự hồi quy. Các mô hình này tạo ra giọng nói song song, cho phép tổng hợp giọng nhanh chóng và chúng cũng mạnh mẽ hơn đối với các đoạn văn dài và đoạn văn nằm ngoài phân phối. Tuy nhiên, chúng vẫn có nhược điểm, đặc biệt là tập trung chủ yếu vào tổng hợp giọng nói từ một người nói mục tiêu và thường đạt được mở rộng cho nhiều người nói bằng cách ghép nối các vectơ nhúng người nói với đầu ra bộ mã hóa. Các mô hình khám phá phong cách nói cũng tích hợp phong cách bằng cách ghép nối vectơ phong cách và nhúng âm vị vào bộ giải mã. Phương pháp này có thể không hiệu quả trong việc nắm bắt biến thiên thời gian của các đặc trưng âm thanh trong giọng nói mục tiêu. Ngược lại, lĩnh vực chuyển giao phong cách giới thiệu phong cách thông qua các phương pháp chuẩn hóa điều kiện có điều kiện như chuẩn hóa thích ứng với ví dụ (AdaIN). Kỹ thuật này đã chứng minh hiệu quả trong việc chuyển giao phong cách [20], [21], [22], mô hình sinh [23], [24], [25], và chỉnh sửa hình ảnh nơ-ron [26], [27]. Việc áp dụng các phương pháp này trong tổng hợp giọng nói vẫn chưa được khám phá rộng rãi, chủ yếu hạn chế vào chuyển đổi giọng và điều chỉnh người nói [28], [29], [30], [31], [32].

Cấu trúc của các mô hình TTS song song cho phép toàn bộ giọng nói được tổng hợp, mở ra cơ hội sử dụng module AdaIN mạnh mẽ để tích hợp các phong cách tổng quát trong tổng hợp giọng nói đa dạng. Các mô hình hiện đại hàng đầu chủ yếu sử dụng khung làm việc TTS song song không tự hồi quy, nhưng do chúng không cân bằng trực tiếp văn bản đầu vào và giọng nói như mô hình tự hồi quy, thường cần một công cụ căn chỉnh bên ngoại như Montreal Forced Aligner [33] được huấn luyện trước trên một bộ dữ liệu lớn. Vì bộ căn chỉnh bên ngoại không được huấn luyện trên dữ liệu và mục tiêu của TTS, các căn chỉnh không phải lúc nào cũng phù hợp tối ưu cho nhiệm vụ TTS. Mặc dù việc huấn luyện các bộ căn chỉnh nội tại giảm bớt một số vấn đề tổng quát [1], [9], [34], [35], nhưng có thể xảy ra quá mức khớp khi các bộ căn chỉnh được huấn luyện trên một bộ dữ liệu TTS nhỏ với chỉ mất mát tái tạo mel.

Ở đây, chúng tôi giới thiệu StyleTTS, một mô hình giải quyết các thách thức nêu trên của việc tích hợp các phong cách nói đa dạng và học một bộ căn chỉnh đồng điều độ đáng tin cậy. StyleTTS tích hợp mô hình sinh dựa trên phong cách vào khung TTS song song để cho phép tổng hợp giọng nói tự nhiên và biểu cảm. Nó tận dụng AdaIN để tích hợp vectơ phong cách tạo ra từ âm thanh tham chiếu, nắm bắt toàn bộ phổ chức năng của đặc điểm phong cách của người nói. Điều này cho phép mô hình của chúng tôi tổng hợp giọng nói mô phỏng các mẫu âm vị và tông điệu cảm xúc trong âm thanh tham chiếu. Với nhiều âm thanh tham chiếu khác nhau, chúng tôi có thể tổng hợp cùng một văn bản theo nhiều phong cách nói khác nhau, thực hiện một ánh xạ một-nhiều mà nhiều hệ thống TTS thường gặp khó khăn. Ngoài ra, mô hình của chúng tôi sử dụng một Bộ căn chỉnh đồng điều độ chuyển giao (TMA) mới để tìm ra sự căn chỉnh văn bản tối ưu, được hỗ trợ bởi một phương pháp tăng cường dữ liệu không biến đổi thời lượng mới để tạo ra dự đoán nguyên tử tự nhiên chống lại các dự đoán thời lượng có thể không tối ưu. Thiết kế của mô hình của chúng tôi mạnh mẽ chống lại các vấn đề tổng quát của bộ căn chỉnh bên ngoại và các vấn đề quá mức khớp có thể xuất phát từ các bộ căn chỉnh nội tại.

Nghiên cứu này giới thiệu ba đóng góp mới sau đây: (i) sự giới thiệu về Bộ căn chỉnh đồng điều độ chuyển giao (TMA), một phương pháp học chuyển giao mới để làm sáng tạo bộ căn chỉnh văn bản được đào tạo trước cho nhiệm vụ TTS, (ii) một phương pháp tăng cường dữ liệu không biến đổi thời lượng để cải thiện dự đoán nguyên tử, và (iii) một mô hình TTS song song tích hợp các phong cách nói tổng quát để tổng hợp giọng nói

1. **RELATED WORK**

Trong số các phương pháp khác được sử dụng bởi các nghiên cứu khác nhau, bài báo cáo này là bài báo cáo đầu tiên trình bày sự tự động hóa của các mô hình khác nhau để chuẩn đoán trên hình ảnh X-quang trong quy mô nghiên cứu học thuật sinh viên mà có thể chẩn đoán viêm phổi ở giai đoạn rất sớm. Trong khi các phương pháp thông thường và phóng xạ đã đề cập có thể có hiệu quả, nghiên cứu của nhóm chúng tôi trình bày một cách tiếp cận học sâu phân loại viêm phổi này. Trong khi nghiên cứu tôi tập trung vào 14 bệnh. Tuy nhiên, do thử nghiệm để chiết xuất 14 loại bệnh khác nhau với một mô hình chỉ có thể đạt được độ chính xác 71,3%. Hơn nữa, LSTM sử dụng nhiều hình ảnh để phân loại một hình ảnh duy nhất, trong khi thử nghiệm và mô hình được đề xuất của chúng tôi chỉ cần mạng nơ-ron được đào tạo trước trọng số để phân loại từng ảnh một. Ngoài ra, của chúng tôi độ chính xác khi thử nghiệm mang lại 78,73%, sử dụng cùng một tập dữ liệu.

Bài báo cáo này trình bày một phương pháp phát hiện sử dụng các bước bổ sung sau thuật toán phân loại. Như một Convolutional thông thường mạng nơ-ron phân loại toàn bộ hình ảnh, các phương pháp phân đoạn như vậy sử dụng phân loại pixelwise, cuối cùng, áp dụng lớp không hợp lệ để phân loại từng pixel một bởi một và cuối cùng tách các đối tượng khác nhau trong một hình ảnh, xương là những thứ phổ biến nhất cho những người đã đề cập nhiệm vụ. Trong một nghiên cứu khác nhằm tiến hành phát hiện sớm không phải cho bệnh viêm phổi mà cho bệnh lồng ngực qua hàng tuần phân loại với mạng nơ-ron tích tụ. Bài báo cáo này phát hiện thành công các mẫu cho bệnh nhân có lồng ngực bệnh hoặc một trong những có thể có bệnh đã đề cập. Chưa, không có hoạt động về phân loại viêm phổi. Trong nghiên cứu này, chúng tôi trình bày một phương pháp mới để phân loại sự tồn tại của bệnh viêm phổi trong hình ảnh chụp X-quang. Chúng tôi đề xuất một hai bước xử lý hình ảnh trước khi đào tạo mô hình học sâu của chúng tôi, trong để làm cho các đặc điểm của hình ảnh X quang rõ ràng hơn và rõ ràng để giảm bớt quá trình phân loại. Chúng tôi, sau đó, thực hiện một mạng nơ-ron phức hợp theo sau bởi một mạng nơ-ron dư mạng cho quá trình phân loại.

Bài báo cáo của chúng tôi trình bày một phương pháp phát hiện các mẫu tổn thương trong khu vực phổi trên nền ảnh X-Rays bằng cách sử dụng kiến trúc Chex\_Net kết hợp phân đoạn tổn thương được xử lý bởi Unet đồng thời được S.O.T.A cùng nhiều kiến khác để tiến hành đo đạc kết quả . Mô hình được đề xuất hiệu quả hơn so với so với kiến trúc Chex\_Net do được sử đào tạo trên kích thước ảnh gốc và được preprocessing rất kỹ [4, 8. 14, 17]. Mô hình này cũng có tiềm năng áp dụng trong các hệ thống nhỏ thực tế.

1. **METHODOLOGY**
2. Proposed Framework:

Cho {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>t</mi><mo>&#xA0;</mo><mo>&#x2208;</mo><mi>T</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"} là các phoneme đầu vào và {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi><mo>&#xA0;</mo><mo>&#x2208;</mo><mi>X</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"} là một mel-spectrogram tham chiếu tùy ý, mục tiêu của chúng tôi là huấn luyện một hệ thống tạo ra mel-spectrogram {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi>x</mi><mo>~</mo></mover><mo>&#x2208;</mo><mi>X</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"} tương ứng với giọng điệu của t và phản ánh các phong cách nói chung của {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"}. Phong cách nói chung được định nghĩa là bất kỳ đặc điểm nào trong âm thanh tham chiếu {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"} ngoại trừ nội dung phổ âm (16), bao gồm nhưng không giới hạn là mô hình âm vị, căng cơ, chuyển đổi formants, tốc độ nói và danh tính người nói. Khung cảnh của chúng tôi bao gồm tám module có thể được chia thành ba loại chính: (i) các module tạo ra giọng nói bao gồm **Text Encoder**, **Style Encoder** và **Decoder**, (ii) các module dự đoán TTS bao gồm **Duration** và **Prosody Prediction**. (iii) các module tiện ích chỉ được sử dụng trong quá trình đào tạo bao gồm **Discriminator**, **Text Aligner** và **Pitch Extractor**.

Chúng tôi cung cấp một tổng quan về kiến trúc của chúng tôi trong Hình 1. Chúng tôi chi tiết từng module dưới đây.

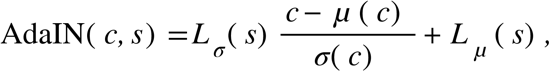
**Text Encoder**. Bộ mã văn bản T chuyển đổi các phoneme ***t*** thành một biểu diễn ẩn ***h***text = *T*(***t***)*.* Nó bao gồm một CNN 3 lớp theo sau là một LSTM hai chiều [36].

**Text Aligner**. Bộ làm đồng bộ văn bản ***A*** tạo ra một sự phù hợp ***d***align giữa mel-spectrogram và phoneme. Chúng tôi huấn luyện một bộ làm đồng bộ văn bản ***A*** cùng với giải mã ***G*** trong giai đoạn tái tạo. Mô phỏng theo giải mã của **Tacotron 2** với chú ý, ***A*** được huấn luyện ban đầu trong nhiệm vụ nhận dạng giọng nói tự động (ASR) bằng cách sử dụng tập dữ liệu **LibriSpeech** [37] và sau đó được điều chỉnh đồng thời với giải mã của chúng tôi. Chúng tôi gọi một bộ làm đồng bộ văn bản với thiết lập này (được huấn luyện trước trên các bộ dữ liệu lớn và được điều chỉnh cho các nhiệm vụ TTS) là **Transferable Monotonic Aligner** (TMA).

**Style Encoder**. Cho một mel-spectrogram đầu vào {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"}, bộ mã của chúng tôi tạo ra một vectơ phong cách {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>s</mi><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mi>E</mi><mfenced><mi>x</mi></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}. Với nhiều âm thanh tham chiếu khác nhau, *E* có thể tạo ra các biểu diễn phong cách đa dạng, cho phép giải mã *G* tạo ra giọng nói phản ánh phong cách *s* của một âm thanh tham chiếu {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"}. *E* bao gồm bốn khối dư dội [38] theo sau là một lớp lọc trung bình theo trục thời gian.

**Pitch Extractor**. Như trong **FastPitch** [11], chúng tôi trích xuất tần số F0 trực tiếp ở Hertz mà không cần xử lý thêm, mang lại biểu diễn đơn giản hóa hơn và cho phép kiểm soát nâng cao của tần số nói. Thay vì sử dụng phương pháp phát hiện chu kỳ âm thanh [39] được sử dụng trong **FastPitch** để ước lượng tần số thực tế, chúng tôi huấn luyện một trích xuất tần số *F* end-to-end với giải mã của chúng tôi để có ước lượng chính xác hơn. Trích xuất tần số của chúng tôi là một mạng **JDC** [40], được huấn luyện trước trên **LibriSpeech** với F0 thực tế được ước lượng bằng **YIN** [41]. Trích xuất này được điều chỉnh lại với giải mã để dự đoán tần số {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>p</mi><mi>x</mi></msub><mo>=</mo><mi>F</mi><mfenced><mi>x</mi></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}cho việc tái tạo của {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"}.

**Decoder**. Giải mã của chúng tôi *G* được huấn luyện để tái tạo mel-spectrogram đầu vào {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"}, được biểu diễn bởi {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi>x</mi><mo>&#x2227;</mo></mover><mo>=</mo><mo>&#xA0;</mo><mi>G</mi><mfenced><mrow><msub><mi>h</mi><mrow><mi>t</mi><mi>e</mi><mi>x</mi><mi>t</mi></mrow></msub><mo>.</mo><msub><mi>d</mi><mrow><mi>a</mi><mi>l</mi><mi>i</mi><mi>g</mi><mi>n</mi></mrow></msub><mo>,</mo><mi mathvariant=\"bold-italic\">s</mi><mo mathvariant=\"bold\">,</mo><msub><mi>p</mi><mi>x</mi></msub><mo>,</mo><msub><mi>n</mi><mi>x</mi></msub></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}. Ở đây, *h*text · *d*align là biểu diễn ẩn được làm phẳng của các phoneme đã được làm đồng bộ, {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>s</mi><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><mi>E</mi><mfenced><mi>s</mi></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}là style vector của {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"}, *px* là đường đi tần số của {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"} và *nx* là năng lượng (được biểu diễn bằng norm log) của {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>x</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"} mỗi frame. Giải mã bao gồm bảy khối dư dội với **AdaIN** [19].



Trong đó, *c* là một kênh đơn của các bản đồ đặc trưng, *s* là style vector, *µ*(·) và *σ*(·) biểu thị giá trị trung bình và độ lệch chuẩn của kênh và *Lσ* và *Lµ* là các chiếu chiến lược tuyến tính đã được học để tính toán adaptive gain và bias bằng cách sử dụng style vector *s*. Việc sử dụng AdaIN là một trong những khác biệt lớn giữa mô hình của chúng tôi và các mô hình TTS khác có bộ **Style Encoder** như [13] và [16]. **Các ưu điểm của AdaIN trong tổng hợp giọng nói đa dạng được thảo luận thêm trong Phụ lục A-B.**

Để ngăn chất lượng của các đặc trưng quan trọng bị giảm, chúng tôi nối tần số *px*, năng lượng *nx* và các đặc trưng nguyên thể của phoneme cuối cùng *hres* và đưa chúng vào các khối nguyên thể sau AdaIN. Quy trình này được mô tả chi tiết hơn trong Bảng , và độ hiệu quả của nó được thảo luận trong Phần IV-D.

**Discriminator**. Chúng tôi bao gồm một bộ phân biệt ***D*** để hỗ trợ việc huấn luyện giải mã để có chất lượng âm thanh tốt hơn [1]. Bộ phân biệt chia sẻ kiến trúc giống như bộ mã phong cách của chúng tôi.

**Duration Predictor**. Duration Predictor của chúng tôi bao gồm một LSTM hai chiều 3 lớp *R* với module chuẩn hóa lớp được điều chỉnh theo tầng động [42] sau là một linear projection *L*. Duration Predictor được chia sẻ với Prosody Predictor *P* thông qua *hprosody* = *R*(*htext*) được đưa vào *P*.

**Prosody Predictor**. Prosody Predictor *P* của chúng tôi dự đoán cả tần số {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mover><mi>p</mi><mo>&#x2227;</mo></mover><mi>x</mi></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"} và năng lượng {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mover><mi>n</mi><mo>&#x2227;</mo></mover><mi>x</mi></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"} với văn bản và style vector đã cho. Biểu diễn chia sẻ đã được làm đồng bộ {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"bold-italic\">h</mi><mrow><mi>p</mi><mi>r</mi><mi>o</mi><mi>s</mi><mi>o</mi><mi>d</mi><mi>y</mi></mrow></msub><mo mathvariant=\"bold\">&#xA0;</mo><mo mathvariant=\"bold\">&#xB7;</mo><mi mathvariant=\"bold-italic\">a</mi></mstyle></math>","origin":"MathType for Microsoft Add-in"} được xử lý thông qua một lớp LSTM hai chiều được chia sẻ để tạo ra hprosody, sau đó được đưa vào hai bộ ba khối dư dội với AdaIN và một lớp linear projection, một cho đầu ra tần số và một cho đầu ra năng lượng (xem Phụ lục D để biết chi tiết).

1. Training Objectives:

A diagram of a process flow

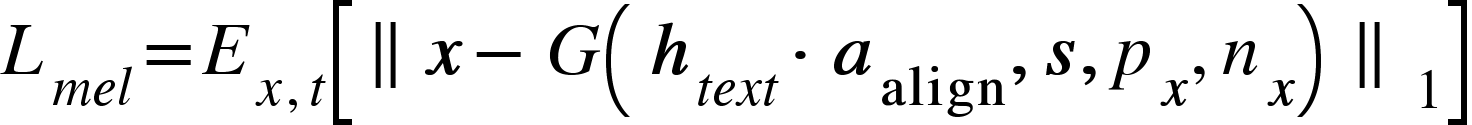
Description automatically generated

Hình 1 Quá trình train và inference của mô hình

Quá trình đào tạo của mô hình được chia thành hai giai đoạn để cho phép tích hợp kỹ thuật gia tăng dữ liệu không đổi thời gian, một đóng góp quan trọng của công việc. Trong giai đoạn đầu tiên, mô hình học tái tạo mel-spectrogram từ văn bản, tần số, năng lượng và phong cách. Giai đoạn thứ hai giữ nguyên tất cả các module ngoại trừ bộ dự đoán thời gian và nhịp điệu, mà được huấn luyện để dự đoán thời lượng, tần số và năng lượng từ văn bản đã cho.

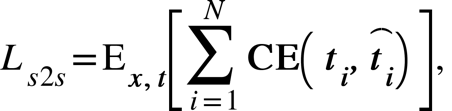
* 1. First Stage Objectives:

**Mel reconstruction.** Với một mel-spectrogram *x* ∈ *X* và văn bản tương ứng *t* ∈ *T*, chúng tôi huấn luyện encoder theo việc tái tạo lại hàm mất mát *L*1

 (2)

Với ***h****text* = *T*(***t***) là biểu diễn ẩn của phoneme đã được mã hóa, ***a***align = *A*(***x***, ***t***) là sự căn chỉnh chú ý từ **Text Aligner**, *s* = *E*(***x***) là vectơ phong cách của ***x***, *p****x*** = *F*(***x***) là tần số F0 của ***x*** và *n****x*** là năng lượng của ***x***. Đối với việc huấn luyện end-to-end (E2E) với **Decoder** và **Text Aligner**, áp dụng một chiến lược mới 50%-50%: một nửa thời gian, sử dụng đầu ra chú ý gốc như là sự làm đồng bộ, cho phép truyền ngược gradient thông qua bộ làm đồng bộ văn bản; trong nửa còn lại, chúng tôi sử dụng phiên bản làm đồng bộ không khác biệt của sự làm đồng bộ chú ý thông qua các thuật toán lập kế hoạch động [9] để huấn luyện giải mã tạo ra giọng nói thông minh từ sự làm đồng bộ cứng theo chiều động trong suốt triển khai. Phương pháp tiếp cận sáng tạo này hiệu quả trong việc điều chỉnh bộ làm đồng bộ văn bản được huấn luyện trước để tạo ra sự làm đồng bộ tối ưu cho tái tạo giọng nói, từ đó nâng cao chất lượng mẫu của giọng nói được tạo ra. Hiệu quả của chiến lược này được phân tích trong phần IV-D.

**TMA objectives.** tôi điều chỉnh lại **Text Aligner** của mình bằng sequence-to-sequence ASR objectives *Ls2s* gốc để đảm bảo sự làm đồng bộ chú ý đúng đắn được duy trì trong quá trình huấn luyện E2E:



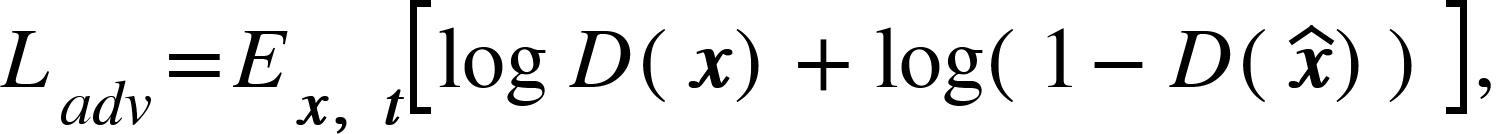
Với N là số lượng phoneme trong ***t***, ***t****i* là biểu tượng phoneme thứ *i* của ***t***, {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><msub><mi mathvariant=\"bold-italic\">t</mi><mi mathvariant=\"bold\">i</mi></msub><mo>&#x23DC;</mo></mover></mstyle></math>","origin":"MathType for Microsoft Add-in"}

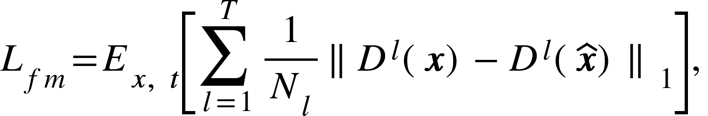
là biểu tượng phoneme được dự đoán thứ *i* và **CE**(·) là hàm Cross-Entropy. Vì sự làm đồng bộ này không nhất thiết phải là một dãy đơn điệu, chúng tôi sử dụng một mất mát L-1 đơn giản *Lmono* để đảm bảo đồng bộ giữa Soft Attention Alignment và phiên bản Monotonic không khác biệt:

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>L</mi><mrow><mi>m</mi><mi>o</mi><mi>n</mi><mi>o</mi></mrow></msub><mo>=</mo><msub><mi>E</mi><mrow><mi mathvariant=\"bold-italic\">x</mi><mo mathvariant=\"bold\">,</mo><mi mathvariant=\"bold-italic\">t</mi></mrow></msub><mfenced open=\"[\" close=\"]\"><mrow><mo>&#x2225;</mo><msub><mi mathvariant=\"bold-italic\">a</mi><mrow><mi>a</mi><mi>l</mi><mi>i</mi><mi>g</mi><mi>n</mi></mrow></msub><mo>-</mo><msub><mi mathvariant=\"bold-italic\">a</mi><mrow><mi>h</mi><mi>a</mi><mi>r</mi><mi>d</mi></mrow></msub><msub><mo>&#x2225;</mo><mn>1</mn></msub></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

nơi *aalign* = *A*(*x,t*) làm đồng bộ chú ý và *ahard* làm đồng bộ cứng đơn điệu được thu được thông qua các thuật toán quy hoạch động (xem Phụ lục A-A để biết chi tiết).

**Adversarial objectives**. Chúng tôi sử dụng hai hàm mất mát Adversarial, hàm mất mát cross-entropy gốc *Ladv* để đào tạo và bổ sung thêm hàm mất mát feature-matching [43] *Lfm* để cải thiện chất lượng âm thanh của mel-spectrogram được tái tạo:





Với {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi mathvariant=\"bold-italic\">x</mi><mo>^</mo></mover></mstyle></math>","origin":"MathType for Microsoft Add-in"} là mel-spectrogram tái tạo bởi *G*, *T* là tổng số lớp trong *D* và *Dl* chỉ đến bản đồ tính năng đầu ra của lớp thứ l với *Nl* Feature. Hàm mất mát phù hợp tính năng có thể được xem xét như một mất mát tái tạo các đặc trưng ẩn của giọng nói thực và được đánh giá bởi bộ phân biệt.

First stage full objectives. có thể tóm tắt như sau với các siêu tham số *λs2s, λmono*, *λadv* và *λfm*:

Second Stage Objectives:

**Duration Predictor**: tôi sử dụng mất mát L-1 để đào tạo mô hình dự đoán thời lượng:

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>L</mi><mrow><mi>d</mi><mi>u</mi><mi>r</mi></mrow></msub><mo>=</mo><msub><mi>E</mi><mi>d</mi></msub><mfenced open=\"[\" close=\"]\"><mrow><mo>|</mo><mo>|</mo><mi mathvariant=\"bold-italic\">d</mi><mo mathvariant=\"bold\">&#xA0;</mo><mo mathvariant=\"bold\">-</mo><mo mathvariant=\"bold\">&#xA0;</mo><msub><mi mathvariant=\"bold-italic\">d</mi><mrow><mi mathvariant=\"bold\">p</mi><mi mathvariant=\"bold\">r</mi><mi mathvariant=\"bold\">e</mi><mi mathvariant=\"bold\">d</mi></mrow></msub><mo>|</mo><msub><mo>|</mo><mn>1</mn></msub></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

trong đó *d* là thời lượng thực tế thu được bằng cách cộng *aalign* dọc theo trục khung mel. {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"bold-italic\">d</mi><mrow><mi mathvariant=\"bold\">p</mi><mi mathvariant=\"bold\">r</mi><mi mathvariant=\"bold\">e</mi><mi mathvariant=\"bold\">d</mi></mrow></msub><mo mathvariant=\"bold\">=</mo><mi>L</mi><mfenced><mrow><mi>R</mi><mfenced><mrow><msub><mi mathvariant=\"bold-italic\">h</mi><mrow><mi>t</mi><mi>e</mi><mi>x</mi><mi>t</mi></mrow></msub><mo mathvariant=\"bold\">,</mo><mi mathvariant=\"bold-italic\">s</mi></mrow></mfenced></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"} là thời lượng dự đoán dưới Style Vector ***s***.

Prosody Predictor: tôi huấn luyện mô hình dự đoán điệu của chúng tôi thông qua một phương pháp tăng dữ liệu độc đáo. Do mô hình dự đoán thời lượng được huấn luyện độc lập từ các module khác (chỉ sử dụng *Ldur*), các dự đoán thời lượng mà nó tạo ra không luôn luôn là tối ưu hoặc tương thích với mô hình dự đoán điệu. Để làm cho mô hình dự đoán điệu chống lại những dự đoán thời lượng có thể không lý tưởng, tăng dữ liệu để giới thiệu tính không biến với điệu trong suốt thời lượng.

Cụ thể hơn, thay vì sử dụng sự đồng bộ giữa thực tế và dự đoán, tần số, và năng lượng của mel-spectrogram gốc, áp dụng một phép nội suy tuyến tính 1D để kéo dài hoặc co melspectrogram trong thời gian và thu được mẫu tăng cường {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi>x</mi><mo>~</mo></mover></mstyle></math>","origin":"MathType for Microsoft Add-in"}. Do đó, tốc độ nói thay đổi, nhưng đường cong tần số và năng lượng vẫn giữ nguyên tính nhất quán. Theo cách này, mô hình dự đoán điệu học cách duy trì tính không biến với tần số và năng lượng, bất kể thời lượng của đoạn nói. Phương pháp này giúp giảm thiểu vấn đề với điệu không tự nhiên khi thời lượng dự đoán là không chính xác.

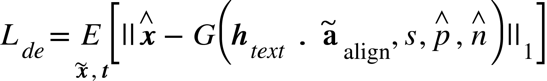
Chúng tôi sử dụng ​{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>L</mi><mrow><mi>f</mi><mn>0</mn></mrow></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"} và {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>L</mi><mi>n</mi></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"} , tương ứng với mất mát tái tạo F0 và năng lượng:

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>L</mi><mrow><mi>f</mi><mn>0</mn></mrow></msub><mo>=</mo><msub><mi>E</mi><mrow><mi>p</mi><mover><mi>x</mi><mo>~</mo></mover></mrow></msub><mfenced open=\"[\" close=\"]\"><mrow><mo>|</mo><mo>|</mo><msub><mi>p</mi><mover><mi>x</mi><mo>~</mo></mover></msub><mo>-</mo><msub><mi>P</mi><mi>p</mi></msub><mfenced><mrow><mi>S</mi><mfenced><mrow><msub><mi mathvariant=\"bold-italic\">h</mi><mrow><mi mathvariant=\"bold\">t</mi><mi mathvariant=\"bold\">e</mi><mi mathvariant=\"bold\">x</mi><mi mathvariant=\"bold\">t</mi></mrow></msub><mo>,</mo><mi>s</mi></mrow></mfenced><mo>&#xB7;</mo><mover><msub><mi mathvariant=\"bold-italic\">a</mi><mrow><mi mathvariant=\"bold\">a</mi><mi mathvariant=\"bold\">l</mi><mi mathvariant=\"bold\">i</mi><mi mathvariant=\"bold\">g</mi><mi mathvariant=\"bold\">n</mi></mrow></msub><mo>~</mo></mover></mrow></mfenced><mo>|</mo><msub><mo>|</mo><mn>1</mn></msub></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>L</mi><mi>n</mi></msub><mo>=</mo><msub><mi>E</mi><mover><mi>x</mi><mo>~</mo></mover></msub><mfenced open=\"[\" close=\"]\"><mrow><mo>|</mo><mo>|</mo><msub><mi>n</mi><mover><mi>x</mi><mo>~</mo></mover></msub><mo>-</mo><msub><mi>P</mi><mi>n</mi></msub><mfenced><mrow><mi>S</mi><mfenced><mrow><msub><mi mathvariant=\"bold-italic\">h</mi><mrow><mi>t</mi><mi>e</mi><mi>x</mi><mi>t</mi></mrow></msub><mo>,</mo><mi>s</mi></mrow></mfenced><mo>&#xB7;</mo><mover><msub><mi mathvariant=\"bold-italic\">a</mi><mrow><mi mathvariant=\"bold\">a</mi><mi mathvariant=\"bold\">l</mi><mi mathvariant=\"bold\">i</mi><mi mathvariant=\"bold\">g</mi><mi mathvariant=\"bold\">n</mi></mrow></msub><mo>~</mo></mover></mrow></mfenced><mo>|</mo><msub><mo>|</mo><mn>1</mn></msub></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}

Trong đó, {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi>p</mi><mover><mi>x</mi><mo>~</mo></mover></msub><mo>,</mo><msub><mi>n</mi><mover><mi>x</mi><mo>~</mo></mover></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"} và {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mover><mi>a</mi><mo>~</mo></mover><mrow><mi>a</mi><mi>l</mi><mi>i</mi><mi>g</mi><mi>n</mi></mrow></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"} là tần số, năng lượng và sự đồng bộ của {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi mathvariant=\"bold-italic\">x</mi><mo mathvariant=\"bold\">~</mo></mover><mo>&#xA0;</mo><mo>&#x2208;</mo><mover><mi>X</mi><mo>~</mo></mover></mstyle></math>","origin":"MathType for Microsoft Add-in"} tập dữ liệu được tăng cường. *Pp* đại diện cho đầu ra của tần số từ mô hình Prosody Prediction và *Pn* là đâu ra của năng lượng.

**Decoder reconstruction**: Cuối cùng, để đảm bảo rằng tần số và năng lượng dự đoán có thể được sử dụng hiệu quả bởi giải mã. Cho mel-spectrogram được kéo dài hoặc co lại trong quá trình gia tăng dữ liệu, việc sử dụng chúng như làm đồng bộ có thể dẫn đến các hiện tượng không mong muốn trong việc tái tạo điệu. Thay vào đó, huấn luyện mô hình dự đoán điệu để tạo ra dự đoán tần số và năng lượng có thể hiệu quả khi được sử dụng bởi giải mã:



Trong đó, {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi>x</mi><mo>&#x2227;</mo></mover><mo>=</mo><mi>G</mi><mfenced><mrow><msub><mi mathvariant=\"bold-italic\">h</mi><mrow><mi>t</mi><mi>e</mi><mi>x</mi><mi>t</mi></mrow></msub><mo mathvariant=\"bold\">&#xA0;</mo><mo mathvariant=\"bold\">.</mo><mo mathvariant=\"bold\">&#xA0;</mo><msub><mover><mi mathvariant=\"bold\">a</mi><mo mathvariant=\"bold\">~</mo></mover><mrow><mi mathvariant=\"bold\">a</mi><mi mathvariant=\"bold\">l</mi><mi mathvariant=\"bold\">i</mi><mi mathvariant=\"bold\">g</mi><mi mathvariant=\"bold\">n</mi></mrow></msub><mo mathvariant=\"bold\">,</mo><mi>s</mi><mo>,</mo><mover><mi>p</mi><mo>~</mo></mover><mo>,</mo><mo>|</mo><mo>|</mo><mover><mi mathvariant=\"bold-italic\">x</mi><mo mathvariant=\"bold\">~</mo></mover><mo>|</mo><mo>|</mo></mrow></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"} là tái tạo của {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi mathvariant=\"bold-italic\">x</mi><mo>~</mo></mover><mo>&#xA0;</mo><mo>&#x2208;</mo><mo>&#xA0;</mo><mover><mi>X</mi><mo>~</mo></mover><mo>,</mo><mo>&#xA0;</mo><mover><mi>p</mi><mo>&#x2227;</mo></mover><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><msub><mi>P</mi><mi>p</mi></msub><mo>&#xA0;</mo><mo>(</mo><mi>S</mi><mo>&#xA0;</mo><mo>(</mo><msub><mi mathvariant=\"bold-italic\">h</mi><mrow><mi>t</mi><mi>e</mi><mi>x</mi><mi>t</mi></mrow></msub><mo>,</mo><mo>&#xA0;</mo><mi>s</mi><mo>)</mo><mo>&#xA0;</mo><mo>&#xB7;</mo><mo>&#xA0;</mo><msub><mover><mi mathvariant=\"bold-italic\">a</mi><mo>~</mo></mover><mrow><mi>a</mi><mi>l</mi><mi>i</mi><mi>g</mi><mi>n</mi></mrow></msub><mo>)</mo></mstyle></math>","origin":"MathType for Microsoft Add-in"} là tần số dự đoán và {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi>n</mi><mo>&#x2227;</mo></mover><mo>&#xA0;</mo><mo>=</mo><mo>&#xA0;</mo><msub><mi>P</mi><mi>n</mi></msub><mo>&#xA0;</mo><mo>(</mo><mi>S</mi><mo>&#xA0;</mo><mo>(</mo><msub><mi mathvariant=\"bold-italic\">h</mi><mrow><mi>t</mi><mi>e</mi><mi>x</mi><mi>t</mi></mrow></msub><mo>,</mo><mo>&#xA0;</mo><mi mathvariant=\"bold-italic\">s</mi><mo>)</mo><mo>&#xA0;</mo><mo>&#xB7;</mo><mo>&#xA0;</mo><msub><mover><mi>a</mi><mo>~</mo></mover><mrow><mi>a</mi><mi>l</mi><mi>i</mi><mi>g</mi><mi>n</mi></mrow></msub><mo>)</mo></mstyle></math>","origin":"MathType for Microsoft Add-in"} là năng lượng dự đoán.

Mục tiêu đầy đủ giai đoạn thứ hai có thể tóm tắt như sau với các siêu tham số *λdur*, *λf0* và *λn*:

{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><munder><mrow><mi>m</mi><mi>i</mi><mi>n</mi></mrow><mrow><mi>S</mi><mo>,</mo><mo>&#xA0;</mo><mi>L</mi><mo>,</mo><mo>&#xA0;</mo><mi>P</mi></mrow></munder><mo>&#xA0;</mo><msub><mi>L</mi><mrow><mi>d</mi><mi>e</mi></mrow></msub><mo>+</mo><msub><mi>&#x3BB;</mi><mrow><mi>d</mi><mi>u</mi><mi>r</mi></mrow></msub><msub><mi>L</mi><mrow><mi>d</mi><mi>u</mi><mi>r</mi></mrow></msub><mo>+</mo><msub><mi>&#x3BB;</mi><mrow><mi>f</mi><mn>0</mn></mrow></msub><msub><mi>L</mi><mrow><mi>f</mi><mn>0</mn></mrow></msub><mo>+</mo><msub><mi>&#x3BB;</mi><mi>n</mi></msub><msub><mi>L</mi><mi>n</mi></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"}

1. **EXPERIMENT AND RESULT**
2. Datasets:

Chúng tôi đã tiến hành một chuỗi thí nghiệm đa dạng trên ba bộ dữ liệu khác nhau để đánh giá hiệu suất của mô hình của chúng tôi. Trước hết, chúng tôi đã tập trung vào việc huấn luyện một mô hình chỉ có một người nói trên bộ dữ liệu LJSpeech [44]. Bộ dữ liệu này bao gồm 13,100 đoạn âm thanh ngắn, tổng cộng khoảng 24 giờ, mang đến một thách thức lớn đối với mô hình trong việc học và tái tạo các đặc điểm độc nhất của mỗi người nói. Chúng tôi đã áp dụng phương pháp chia tách dữ liệu giống như trong mô hình VITS, với bộ huấn luyện chứa 12,500 mẫu, bộ xác thực 100 mẫu và bộ kiểm tra 500 mẫu.

Bên cạnh đó, chúng tôi đã tiếp tục nghiên cứu bằng cách huấn luyện một mô hình nhiều người nói trên bộ dữ liệu LibriTTS [45]. Phần con train-clean-460 của LibriTTS là một nguồn tài nguyên đáng kể với khoảng 245 giờ âm thanh từ 1,151 người nói. Để đảm bảo tính đa dạng, chúng tôi loại bỏ các đoạn nói có thời lượng lớn hơn 30 giây hoặc ngắn hơn một giây. Bộ con train-clean-460 được chia ngẫu nhiên thành tập huấn luyện (98%), tập xác thực (1%), và tập kiểm tra (1%), theo phương pháp trong [32]. Sử dụng tập kiểm tra để đánh giá, chúng tôi thu được thông tin chi tiết về khả năng tổng quát hóa và hiệu suất của mô hình.

Chúng tôi cũng đã kiểm tra khả năng thích ứng của mô hình đối với người nói mới mà không cần huấn luyện lại, sử dụng bộ dữ liệu VCTK [46]. Chúng tôi giữ nguyên phương pháp chia tách người nói trong [47], với 88 người nói được sử dụng để huấn luyện và 20 người còn lại được dành cho kiểm tra.

Đối với thách thức của việc tái tạo tiếng nói mang tính cảm, chúng tôi đã huấn luyện một mô hình nhiều người nói trên bộ dữ liệu tiếng nói có tính cảm (ESD) [48]. Bộ dữ liệu này bao gồm 10 người nói tiếng Trung và 10 người nói tiếng Anh đọc cùng 400 câu ngắn trong năm tình trạng cảm xúc khác nhau. Chúng tôi đã tăng mẫu âm thanh huấn luyện lên 24 kHz để phù hợp với LibriTTS và chuyển đổi văn bản thành âm vị sử dụng công cụ mã nguồn mở. Bằng cách này, chúng tôi có thể nghiên cứu khả năng của mô hình trong việc tạo ra âm thanh có đặc tính prosody đa dạng, một yếu tố quan trọng trong việc truyền đạt cảm xúc trong tiếng nói. Chúng tôi cũng sử dụng các kỹ thuật trích xuất mel-spectrogram tiên tiến để bảo đảm chất lượng và độ chính xác của quá trình tổng hợp âm thanh.

A screenshot of a graph

Description automatically generated

Hình 2

|  |  |
| --- | --- |
| Model | MOS-N (CI) |
| Ground Truth | 4.32 (± 0.04) |
| Tacotron 2 + HiFi-GAN | 3.01 (± 0.06) |
| FastSpeech 2 + HiFi-GAN | 2.97 (± 0.06) |
| VITS | 3.78 (± 0.06) |
| Our Proposal (StyleTTS + HiFi-GAN) | **4.01 (± 0.05)** |

**BẢNG I.**

**SO SÁNH ĐIỂM MOS ĐƯỢC ĐÁNH GIÁ VỚI KHOẢNG TIN CẬY 95% (CI) TRÊN BỘ DỮ LIỆU LJSPEECH.**

|  |  |  |
| --- | --- | --- |
| Model | MOS-N (CI) | MOS-S (CI) |
| Ground Truth | 4.35 (± 0.04) | 3.90 (± 0.07) |
| FastSpeech 2 + HiFi-GAN | 3.00 (± 0.06) | 3.51 (± 0.07) |
| VITS | 3.62 (± 0.06) | 3.70 (± 0.07) |
| Our Proposal (StyleTTS + HiFi-GAN) | **4.03 (± 0.05)** | **3.79 (± 0.07)** |

**BẢNG II.**

**SO SÁNH ĐIỂM MOS ĐƯỢC ĐÁNH GIÁ VỚI KHOẢNG TIN CẬY 95% (CI) TRÊN BỘ DỮ LIỆU LIBRITTS.**

**A collage of images of different types of sound waves

Description automatically generated**

**Hình 3 Spectrogram của các audio minh hoạ**

Hình trên đang thể hiện Spectrogram của âm thanh tham chiếu và tiếng nói được tạo ra tương ứng đọc câu "How much variation is there? Let’s find it out." từ mô hình nói đơn được huấn luyện trên bộ dữ liệu LJSpeech. Đường nét nhấn tượng định của giọng đã được biểu diễn bằng các điểm màu trắng. Phía trái trên: Âm thanh tham chiếu của một câu hỏi, "Did England let nature take her course?". Chú ý rằng độ cao của giọng thường tăng lên ở cuối mỗi từ. Phía trái dưới: Tiếng nói được tổng hợp. Mẫu giọng tương tự với độ cao tăng lên ở cuối các từ. Phía phải trên: Âm thanh tham chiếu của một bài nói ngạc nhiên nói "It’s true! I am shocked! My dreams!". Chú ý rằng giọng đi lên trước rồi đi xuống cho mỗi từ. Phía phải dưới: Tiếng nói được tổng hợp với mẫu giọng tương tự của độ cao đi lên và đi xuống cho phần lớn các từ.

1. Data pre-processing và huấn luyện mô hình:

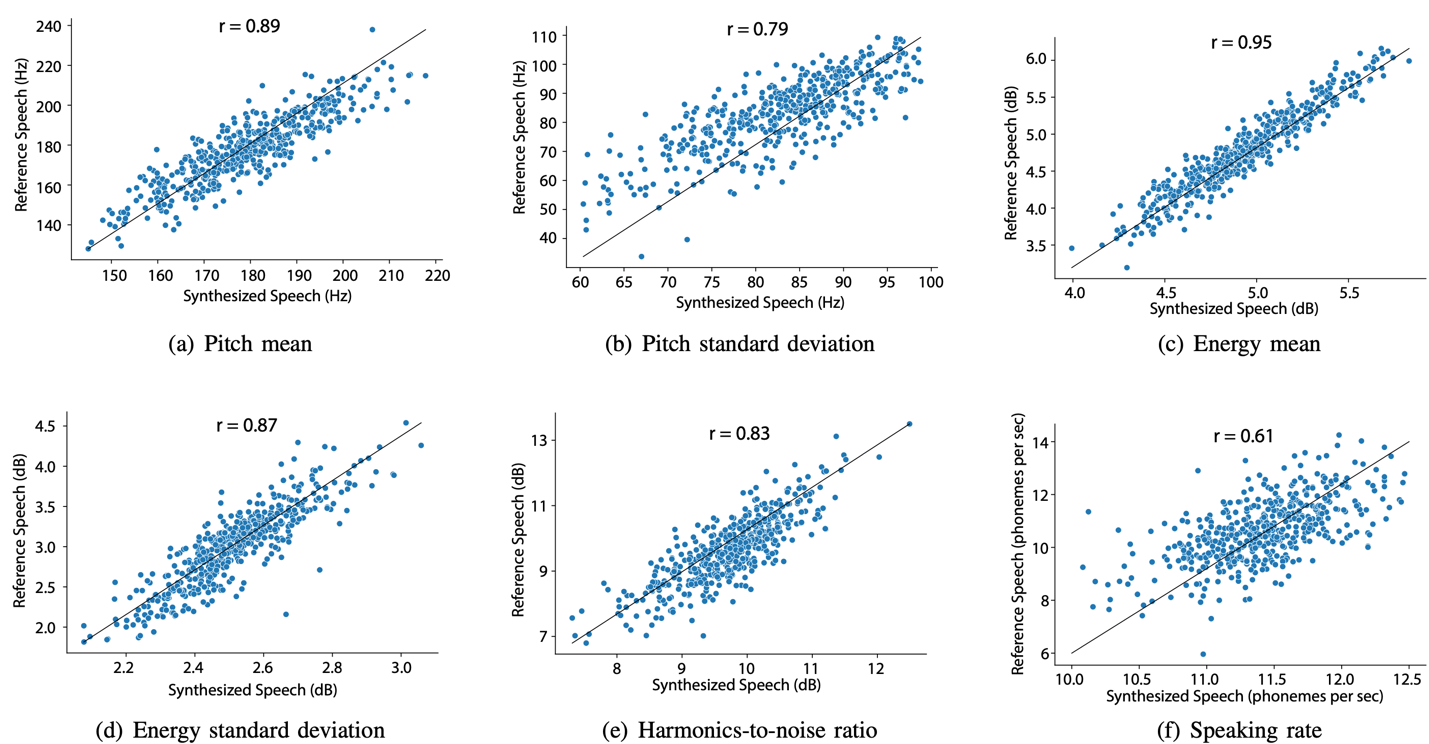
Đối với cả hai giai đoạn, chúng tôi đã đào tạo tất cả các mô hình trong 200 epochs cách sử dụng hàm optimizer AdamW với β1 = 0, β2 = 0.99, weight decay λ = 10−4, learning rate γ = 10−4 và batch size là 64. Chúng tôi thiết lập λs2s = 0.2, λadv = 1, λmono = 5, λfm = 0.2, λdur = 1, λf0 = 0.1 và λn = 1. Việc cài đặt các Hyperparameter này đảm bảo rằng tất cả các giá trị loss đều có cùng tỷ lệ và quá trình training không nhạy cảm với các siêu tham số này. Hệ scale nằm trong khoảng từ 0,75 đến 1,25 để tăng cường dữ liệu. Chúng tôi chia ngẫu nhiên các mel-spectrogram thành các đoạn có độ dài ngắn nhất trong lô. Việc đào tạo và huấn luyện mô hình được thực hiện trên 6 GPU NVIDIA A5000.

1. Implementation and Metrics:

Chúng tôi đã thực hiện hai đánh giá chủ quan: điểm ý kiến trung bình về độ tự nhiên (MOS-N) để đo độ tự nhiên của giọng nói tổng hợp và điểm ý kiến trung bình về độ tương tự (MOS-S) để đánh giá mức độ tương tự giữa giọng nói tổng hợp và tham chiếu cho mô hình nhiều người nói...Trong mọi thử nghiệm, chúng tôi chọn ngẫu nhiên 100 mẫu văn bản từ bộ thử nghiệm. Đối với mỗi văn bản, chúng tôi tổng hợp lời nói bằng cách sử dụng mô hình của mình và các mô hình cơ sở, đồng thời đưa thông tin cơ bản vào để so sánh. Các mô hình cơ sở bao gồm Tacorton 2 [51], FastSpeech 2 [11] và VITS [1]. Đối với các thử nghiệm zero-shot speaker, chúng tôi đã so sánh mô hình của mình với StyleSpeech [32] và YourTTS [47]. Tất cả các mô hình cơ sở đều được đào tạo trước và công khai. Các biểu đồ mel-spectrogram được tạo ra đã được chuyển đổi thành dạng sóng bằng cách sử dụng bộ encoder Hifi-GAN [52] cho tất cả mô hình. Để so sánh công bằng, chúng tôi đã giảm tần số lấy mẫu âm thanh tổng hợp của mình xuống 22 kHz để khớp với âm thanh từ các mẫu cơ sở. Chúng tôi đã sử dụng các tham chiếu ngẫu nhiên khi tổng hợp giọng nói cho các thử nghiệm điều chỉnh single-shot speaker và zero-shot speaker. Đối với các trường hợp multi-speaker, vì quá trình đào tạo của chúng tôi không yêu cầu label của speaker nên để so sánh công bằng với các mẫu khác sử dụng tính năng embedded speaker rõ ràng trong quá trình đào tạo, chúng tôi đã tính trung bình các style vectơ bằng cách sử dụng tất cả các mẫu trong tập huấn luyện từ cùng một speaker làm tham chiếu tới style vector.

Khi đánh giá từng bộ, chúng tôi hoán đổi ngẫu nhiên thứ tự của các mô hình và hướng dẫn các đối tượng nghe và đánh giá chúng mà không cho họ biết nhãn mô hình. Nó tương tự như nhiều kích thích có tham chiếu và neo ẩn (MUSHRA), cho phép các đối tượng so sánh sự khác biệt tinh tế giữa các mô hình. Chúng tôi đã sử dụng sự thật cơ bản làm công cụ kiểm tra sự chú ý ẩn: những người xếp hạng sẽ bị loại khỏi phân tích nếu MOS của sự thật cơ bản không được xếp hạng hai trong số tất cả các mô hình.

Chúng tôi cũng tiến hành đánh giá khách quan bằng cách sử dụng số liệu ASR. Chúng tôi đã đánh giá độ tin cậy của các mô hình đối với các độ dài khác nhau của kiểu nhập văn bản. Chúng tôi đã tạo bốn bộ thử nghiệm có độ dài văn bản lần lượt là L < 10, 10 < L < 50, 50 < L < 100 và 100 < L. Mỗi bộ chứa 100 văn bản được lấy mẫu từ bộ dữ liệu WSJ0 [53]. Chúng tôi đã tính toán tỷ lệ lỗi từ của giọng nói tổng hợp từ cả mô hình một speaker và nhiều speaker bằng cách sử dụng mô hình ASR được đào tạo trước từ ESPnet [54]. Để đo tốc độ suy luận, chúng tôi đã tính toán hệ số thời gian thực (RFT), biểu thị thời gian (tính bằng giây) cần thiết để mô hình tổng hợp dạng sóng một giây. RFT được đo trên máy chủ có 3 GPU NVIDIA RTX A5000 và batch size là 6. Ngoài ra, chúng tôi đã tiến hành phân tích tương tự về mối tương quan của các đặc điểm âm thanh liên quan đến cảm xúc giữa giọng nói tham chiếu và giọng nói tổng hợp bằng cách sử dụng bốn mô hình nhiều speaker. Vì không có kiểu trong FastSpeech 2 và VITS, chúng tôi đã sử dụng mô hình X-vector được đào tạo trước [55] từ Kaldi [56] để trích xuất phần nhúng của loa làm vectơ tham chiếu.



Hình 4 Hệ số tương quan Pearson của sáu đặc điểm âm thanh liên quan đến cảm xúc giữa lời nói tham chiếu và lời nói tổng hợp trên bộ dữ liệu LJ Speech.

Bảng III SO SÁNH MOS ĐƯỢC ĐÁNH GIÁ VỚI KHOẢNG ĐỘ Tin cậy (CI) 95% TRÊN BỘ DỮ LIỆU VCTK ĐỂ ĐIỀU CHỈNH Unseen Speaker.

A table with numbers and symbols

Description automatically generated

1. Result:

A. Hiệu suất mô hình:  
Bảng I, II và III trình bày kết quả đánh giá chủ quan của con người trên bộ dữ liệu LJSpeech và LibriTTS. Khi được đánh giá về độ tự nhiên (MOS-N) và độ tương tự (MOS-S), StyleTTS rõ ràng vượt trội so với các mẫu khác, thể hiện hiệu suất vượt trội trong cả single speaker, multi-speaker và zero-shot speaker. Các mô hình của chúng tôi mạnh mẽ hơn so với các mô hình khác (Bảng IV), đặc biệt đối với các văn bản đầu vào dài. Vì chúng tôi không sử dụng các luồng yêu cầu tính toán Jacobian nghịch đảo nên mô hình của chúng tôi nhanh hơn VITS [1], mặc dù nó không được đào tạo từ đầu đến cuối như VITS (Bảng VI).

Chúng tôi lưu ý rằng kết quả đánh giá của chúng tôi khác với kết quả được báo cáo trong các mô hình cơ sở, đặc biệt đối với VITS. Mô hình VITS đã được báo cáo là mang lại kết quả rất gần với thực tế [1]. Tuy nhiên, theo đánh giá của chúng tôi, VITS được nhận thấy là không đạt được mức hiệu suất thực tế cơ bản. Yếu tố chính dẫn đến sự khác biệt này là sự khác biệt trong phương pháp đánh giá. Trong các thử nghiệm VITS, phương pháp đánh giá Điểm ý kiến trung bình (MOS) truyền thống đã được sử dụng, trong đó người đánh giá đánh giá từng mô-đun riêng lẻ mà không có bất kỳ tài liệu tham khảo nào. Việc sử dụng điểm tham chiếu trong đánh giá giống như MUSHRA của chúng tôi mang lại điểm neo cho xếp hạng, đặc biệt là sự thật cơ bản làm tham chiếu, điều này có khả năng làm giảm điểm của các mô hình khác. Một sự khác biệt tương tự đã được báo cáo trong một nghiên cứu gần đây nhằm kiểm tra tác động của các phương pháp đánh giá đối với kết quả MOS [57] và đánh giá VITS của chúng tôi có thể so sánh với các nghiên cứu khác đã cố gắng tái tạo nó trên cả bộ dữ liệu LJSpeech và LibriTTS [58 ], [59], [60], [61].

B. Trực quan hoá Style Vector:

Để xác minh rằng mô hình của chúng tôi có thể tìm hiểu các biểu diễn style có ý nghĩa, chúng tôi chiếu các style vector được trích xuất từ âm thanh tham chiếu vào mặt phẳng 2-D để hiển thị bằng cách sử dụng t-SNE [62]. Chúng tôi đã chọn 50 mẫu của từng cảm xúc từ một speaker duy nhất trong ESD và chiếu style vectơ của từng âm thanh vào không gian 2-D. Có thể thấy trong Hình 2(a) rằng vectơ phong cách của chúng tôi mã hóa rõ ràng các sắc thái cảm xúc của các câu tham chiếu mặc dù quá trình đào tạo không sử dụng nhãn cảm xúc. Chúng tôi cũng tính toán các vectơ kiểu bằng cách sử dụng các mẫu giọng nói từ cùng một loa với mô hình loa đơn của chúng tôi. Mô hình này chỉ được đào tạo trên tập dữ liệu LJSpeech và do đó chưa bao giờ thấy diễn giả được chọn từ ESD trong quá trình đào tạo. Tuy nhiên, trong Hình 2(b), chúng ta thấy rằng mô hình của chúng tôi vẫn có thể nắm bắt rõ ràng âm sắc cảm xúc của các câu, cho thấy rằng ngay cả khi âm thanh tham chiếu đến từ một loa khác với loa đơn được thấy trong quá trình đào tạo, nó vẫn có thể tổng hợp lời nói với giọng điệu cảm xúc chính xác. Điều này cho thấy mô hình của chúng tôi có thể trích xuất ngầm cảm xúc từ tập dữ liệu không được gắn nhãn theo cách tự giám sát. Cuối cùng, chúng tôi hiển thị các vectơ kiểu dự kiến từ 10 loa VCTK không nhìn thấy, mỗi loa có 50 mẫu trong Hình 2(c). Các loa khác nhau được tách biệt hoàn hảo với nhau trong hình chiếu 2-D. Điều này chỉ ra rằng mô hình của chúng tôi có thể tìm hiểu danh tính người nói mà không cần nhãn người nói rõ ràng và do đó thực hiện điều chỉnh người nói không cần chụp.

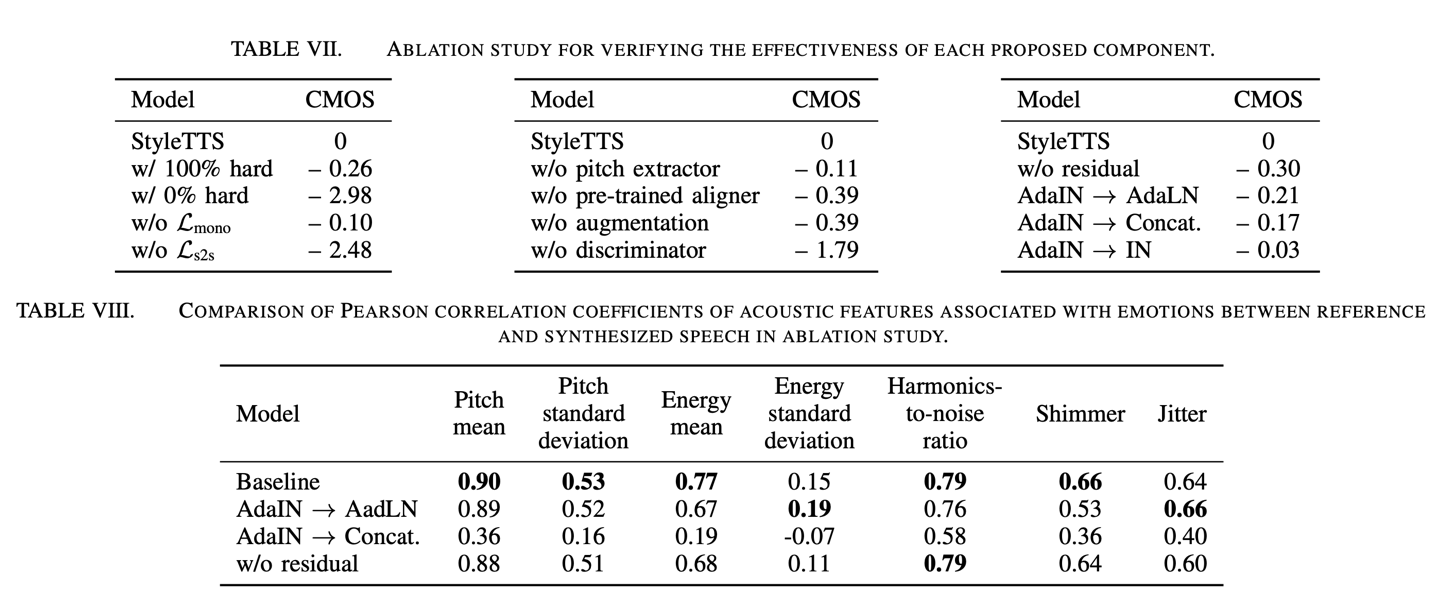
C Tổng hợp giọng nói đa dạng được hỗ trợ theo phong cách

Để chỉ ra rằng các style vectơ đã học thực sự cho phép tổng hợp giọng nói đa dạng, chúng tôi cung cấp một ví dụ về giọng nói tổng hợp với hai âm thanh tham chiếu khác nhau bằng mô hình đơn speaker được đào tạo trên tập dữ liệu LJSpeech trong Hình 3. Có thể thấy rõ rằng giọng nói tổng hợp thu được các khía cạnh khác nhau của bài phát biểu tham chiếu, bao gồm cao độ, nhịp điệu, khoảng dừng và chuyển đổi hình thức. Để định lượng một cách có hệ thống hiệu ứng này, chúng tôi đã vẽ sáu biểu đồ phân tán cho thấy mối tương quan giữa giọng nói tổng hợp và giọng nói tham chiếu trong các đặc điểm âm thanh thường được sử dụng để nhận dạng cảm xúc giọng nói (Hình 4). Sáu đặc điểm này là trung bình cao độ, độ lệch chuẩn cao độ, trung bình năng lượng, độ lệch chuẩn năng lượng, tỷ lệ hài hoà trên tạp âm và tốc độ nói [63]. Tất cả sáu đặc điểm đều thể hiện mối tương quan đáng kể giữa giọng nói tổng hợp và giọng nói tham chiếu (p < 0,001) với các hệ số tương quan đều trên 0,6. Mô hình của chúng tôi cũng vượt trội hơn các mô hình khác trên bộ dữ liệu nhiều loa về mối tương quan đặc điểm âm thanh (Bảng V). Kết quả chỉ ra rằng nhiều khía cạnh của giọng nói tổng hợp được khớp với tham chiếu, cho phép kiểm soát linh hoạt giọng nói tổng hợp chỉ bằng cách chọn âm thanh tham chiếu thích hợp. Vì các mô hình của chúng tôi cũng cho phép kiểm soát hoàn toàn cao độ, năng lượng và thời lượng nên cách tiếp cận của chúng tôi là một trong những mô hình linh hoạt nhất về khả năng kiểm soát tổng hợp giọng nói.

D Nghiên cứu tối giản

Chúng tôi tiếp tục tiến hành một nghiên cứu cắt bỏ để xác minh tính hiệu quả của từng thành phần trong mô hình của chúng tôi bằng các thử nghiệm đánh giá chủ quan của con người. Chúng tôi đã hướng dẫn các đối tượng so sánh mẫu đơn speaker của chúng tôi với mẫu speaker đã bị loại bỏ một thành phần. Chúng tôi đã chuyển đổi xếp hạng thành điểm số ý kiến ​​trung bình so sánh (CMOS) bằng cách lấy sự khác biệt giữa MOS của mô hình cơ sở và mô hình bị loại bỏ. Kết quả được trình bày trong bảng VII và thông tin chi tiết hơn được trình bày trong Phụ lục A. Bảng ngoài cùng bên trái trình bày các kết quả liên quan đến Transferable Monotonic Aligner (TMA) được đề xuất. Khi huấn luyện bao gồm 100% hard alignment sao cho không có độ dốc bị truyền ngược đến các tham số của bộ căn chỉnh văn bản (tương đương với việc sử dụng bộ căn chỉnh bên ngoài chẳng hạn như trong FastSpeech 2), MOS định mức sẽ giảm −0,26. Điều này là do sự thay đổi đồng biến giữa dữ liệu trước đào tạo (LibriSpeech) và dữ liệu TTS (LJ Speech). Một ví dụ về căn chỉnh kém của bộ căn chỉnh bên ngoài được huấn luyện trước được hiển thị trong Hình 5. Điều này cho thấy việc tinh chỉnh bộ căn chỉnh có hiệu quả trong việc cải thiện chất lượng giọng nói tổng hợp. Tuy nhiên, khi sử dụng hard alignment 0% (softe attention 100%), mô hình sẽ bị trang bị quá mức để tái tạo lại giọng nói với căn chỉnh mềm và không thể tạo ra giọng nói có thể nghe được bằng cách căn chỉnh cứng trong khi suy luận (−2,98 CMOS). Chúng tôi cũng thấy rằng cả hai mục tiêu TMA (phương trình 3 và 4) đều quan trọng để tổng hợp giọng nói chất lượng cao.

Bảng ở giữa cho thấy tác động của việc loại bỏ các kỹ thuật và thành phần đào tạo khác nhau. Việc sử dụng bộ tách cao độ bên ngoài (chẳng hạn như các phương pháp dựa trên âm thanh) sẽ làm giảm MOS xuống −0,11. Điều này có thể xảy ra do phương pháp trích xuất cao độ dựa trên âm thanh đôi khi không trích xuất được đường cong F0 chính xác và việc tinh chỉnh bộ trích xuất cao độ cùng với bộ giải mã làm cho mô hình học cách biểu diễn cao độ tốt hơn (xem Phụ lục A-C). Nếu không có bộ căn chỉnh văn bản được huấn luyện trước (chẳng hạn như VITS), MOS được xếp hạng sẽ giảm −0,39. Điều này cho thấy rằng việc học chuyển giao của chúng tôi rất hữu ích trong việc giảm thiểu các vấn đề về trang bị quá mức khi đào tạo bộ chỉnh răng nội bộ với một tập dữ liệu tương đối nhỏ. Việc loại bỏ tính năng tăng cường dữ liệu bất biến theo thời lượng mới của chúng tôi cũng làm giảm hiệu suất. Cuối cùng, việc đào tạo không có yếu tố phân biệt ảnh hưởng đáng kể đến chất lượng âm thanh cảm nhận được. Bảng ngoài cùng bên phải hiển thị các thay đổi về kiến trúc bằng cách loại bỏ các tính năng còn lại và thay thế các thành phần AdaIN trong bộ giải mã và bộ dự đoán bằng chuẩn hóa phiên bản (IN), AdaLN và nối tính năng đơn giản (Concat). Tác dụng của chúng đối với việc phản ánh phong cách cũng được thể hiện trong Bảng VIII. Việc loại bỏ các đặc điểm còn sót lại trong bộ giải mã sẽ làm giảm cả tính tự nhiên và mối tương quan giữa giọng nói tổng hợp và giọng nói tham chiếu. Chuẩn hóa lớp cũng kém hơn IN đối với cả hai số liệu. Việc ghép các kiểu thay cho AdaIN làm giảm đáng kể mối tương quan và giảm độ tự nhiên được xếp hạng, xác nhận quan sát của chúng tôi rằng tất cả các phương pháp trước đây sử dụng phép nối để kết hợp thông tin kiểu ([1], [9], [64], [12], [13], [15], [16]) không hiệu quả bằng AdaIN do thiếu khả năng điều chế theo thời gian (xem Phụ lục A-B). Cuối cùng, chúng tôi thấy rằng việc thay thế AdaIN bằng IN không ảnh hưởng đáng kể đến độ tự nhiên được xếp hạng, cho thấy rằng độ tự nhiên được cải thiện không phải do sự ra đời của các kiểu mà là do những cải tiến kỹ thuật mới của chúng tôi bao gồm TMA, tăng cường dữ liệu, sử dụng IN, trình trích xuất cao độ và phần dư đặc trưng. Tuy nhiên, phong cách cho phép tổng hợp giọng nói đa dạng mà các mô hình không có phong cách không thể làm được.



**CONCLUSION**

StyleTTS, một phương pháp tổng hợp văn bản thành giọng nói (TTS) mới lạ và tự nhiên. Nghiên cứu của chúng tôi thực hiện một bước tiến đặc biệt trong việc tận dụng điểm mạnh của các hệ thống TTS song song với một số cấu trúc mới bao gồm transferable monotonic aligner (TMA) độc đáo đồng thời tích hợp thông tin về kiểu dáng thông qua AdaIN. Chúng tôi đã chứng minh rằng phương pháp này có thể phản ánh hiệu quả các đặc điểm phong cách từ âm thanh tham chiếu. Hơn nữa, Style vectơ từ mô hình của chúng tôi mã hóa một tập hợp thông tin phong phú có trong âm thanh tham chiếu, bao gồm cao độ, năng lượng, tốc độ nói, chuyển tiếp biểu mẫu và nhận dạng người nói. Điều này cho phép dễ dàng kiểm soát các mẫu nhịp điệu và tông màu cảm xúc của giọng nói tổng hợp bằng cách chọn kiểu tham chiếu thích hợp đồng thời hưởng lợi từ việc tổng hợp giọng nói mạnh mẽ và nhanh chóng của các hệ thống TTS song song. Nói chung, chúng cho phép tổng hợp giọng nói tự nhiên với các phong cách giọng nói đa dạng vượt xa những gì đã đạt được trong các hệ thống TTS trước đây.

1. **REFERENCE**

**[1].** J. Kim, J. Kong, and J. Son, “Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech,” in Proceedings of the 38th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139, 18–24 Jul 2021, pp. 5530–5540.

**[2].** Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, “Png bert: augmented bert on phonemes and graphemes for neural tts,” arXiv preprint arXiv:2103.15060, 2021.

**[3].**  X. Tan, J. Chen, H. Liu, J. Cong, C. Zhang, Y. Liu, X. Wang, Y. Leng, Y. Yi, L. He et al., “Naturalspeech: End-to-end text to speech synthesis with human-level quality,” arXiv preprint arXiv:2205.04421, 2022

**[4].**  X. Tan, T. Qin, F. Soong, and T.-Y. Liu, “A survey on neural speech synthesis,” arXiv preprint arXiv:2106.15561, 2021.

**[5].**  W.-N. Hsu, Y. Zhang, R. Weiss, H. Zen, Y. Wu, Y. Cao, and Y. Wang, “Hierarchical generative modeling for controllable speech synthesis,” in International Conference on Learning Representations, 2019.

**[6].**  Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, “Learning latent representa- tions for style control and transfer in end-to-end speech synthesis,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 6945–6949.

**[7].**  Y. Lee, J. Shin, and K. Jung, “Bidirectional variational inference for non- autoregressive text-to-speech,” in International Conference on Learning Representations, 2020.

**[8].**  R. Valle, K. J. Shih, R. Prenger, and B. Catanzaro, “Flowtron: an au- toregressive flow-based generative network for text-to-speech synthesis,” in International Conference on Learning Representations, 2020.

**[9].** J. Kim, S. Kim, J. Kong, and S. Yoon, “Glow-tts: A generative flow for text-to-speech via monotonic alignment search,” Advances in Neural Information Processing Systems, vol. 33, 2020.

**[10].** R. Valle, J. Li, R. Prenger, and B. Catanzaro, “Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens,” in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 6189–6193.

**[11].**  Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Fastspeech 2: Fast and high-quality end-to-end text to speech,” in International Conference on Learning Representations, 2021.

**[12].**  R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, “Towards end-to-end prosody transfer for expressive speech synthesis with tacotron,” in international conference on machine learning. PMLR, 2018, pp. 4693–4702.

**[13].**  Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, “Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis,” in International Conference on Machine Learning. PMLR, 2018, pp. 5180–5189.

**[14].** L. Chen, Y. Deng, X. Wang, F. K. Soong, and L. He, “Speech bert embedding for improving prosody in neural tts,” in ICASSP 2021- 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 6563–6567.

**[15].** G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, and Y. Wu, “Fully- hierarchical fine-grained prosody modeling for interpretable speech synthesis,” in ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2020, pp. 6264–6268.

**[16].** R. Liu, B. Sisman, G. lai Gao, and H. Li, “Expressive tts training with frame and style reconstruction loss,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021.

**[17].** Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Fastspeech: fast, robust and controllable text to speech,” in Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019, pp. 3171–3180.

**[18].** A. Łan ́cucki, “Fastpitch: Parallel text-to-speech with pitch prediction,” in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 6588–6592.

**[19].**  X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.

**[20].**  X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, “Multimodal unsu- pervised image-to-image translation,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 172–189.

**[21].**  M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz, “Few-shot unsupervised image-to-image translation,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 10 551–10 560.

**[22].**  Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, “Stargan v2: Diverse image synthesis for multiple domains,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8188–8197.

**[23].**  T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4401–4410.

**[24].**  T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “An- alyzing and improving the image quality of stylegan,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8110–8119.

**[25].**  T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, “Alias-free generative adversarial networks,” Advances in Neural Information Processing Systems, vol. 34, 2021.

**[26].**  C.-H. Lee, Z. Liu, L. Wu, and P. Luo, “Maskgan: Towards diverse and interactive facial image manipulation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5549–5558.

**[27].** P. Zhu, R. Abdal, Y. Qin, and P. Wonka, “Sean: Image synthesis with semantic region-adaptive normalization,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5104–5113.

**[28].** J. C. Chou and H.-Y. Lee, “One-Shot Voice Conversion by Separating Speaker and Content Representations with Instance Normalization,” in Proc. Interspeech 2019, 2019, pp. 664–668.

**[29].** Y.-H. Chen, D.-Y. Wu, T.-H. Wu, and H.-y. Lee, “Again-vc: A one- shot voice conversion using activation guidance and adaptive instance normalization,” in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 5954–5958.

**[30].** Y. A. Li, A. Zare, and N. Mesgarani, “StarGANv2-VC: A Diverse, Unsupervised, Non-Parallel Framework for Natural-Sounding Voice Conversion,” in Proc. Interspeech 2021, 2021, pp. 1349–1353.

**[31].** M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T.-Y. Liu, “Adaspeech: Adaptive text to speech for custom voice,” arXiv preprint arXiv:2103.00993, 2021.

**[32].** D. Min, D. B. Lee, E. Yang, and S. J. Hwang, “Meta-stylespeech: Multi-speaker adaptive text-to-speech generation,” arXiv preprint arXiv:2106.03153, 2021.

**[33].** M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using kaldi.” in Interspeech, vol. 2017, 2017, pp. 498–502.

**[34].** C. Miao, L. Shuang, Z. Liu, C. Minchuan, J. Ma, S. Wang, and J. Xiao, “Efficienttts: An efficient and high-quality text-to-speech architecture,” in International Conference on Machine Learning. PMLR, 2021, pp. 7700–7709.

**[35].** I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Skerry-Ryan, and Y. Wu, “Parallel tacotron 2: A non-autoregressive neural tts model with differentiable duration modeling,” arXiv preprint arXiv:2103.14574, 2021.

**[36].** M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” 9 IEEE transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

**[37].** V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015, pp. 5206–5210.

**[38].** K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

**[39].** P. Boersma et al., “Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound,” in Proceedings of the institute of phonetic sciences, vol. 17, no. 1193. Citeseer, 1993, pp. 97–110.

**[40].** S. Kum and J. Nam, “Joint detection and classification of singing voice melody using convolutional recurrent neural networks,” Applied Sciences, vol. 9, no. 7, p. 1324, 2019.

**[41].** A. De Cheveigné and H. Kawahara, “Yin, a fundamental frequency estimator for speech and music,” The Journal of the Acoustical Society of America, vol. 111, no. 4, pp. 1917–1930, 2002.

**[42].** J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” arXiv preprint arXiv:1607.06450, 2016.

**[43].** T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” Advances in neural information processing systems, vol. 29, pp. 2234–2242, 2016.

**[44].** K. Ito and L. Johnson, “The lj speech dataset,” https://keithito.com/ LJ-Speech-Dataset/, 2017.

**[45].** H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, “Libritts: A corpus derived from librispeech for text-to-speech,” arXiv preprint arXiv:1904.02882, 2019.

**[46].** J. Yamagishi, C. Veaux, K. MacDonald et al., “Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92),” 2019.

**[47].** E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti, “Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone,” in International Conference on Machine Learning. PMLR, 2022, pp. 2709–2720.

**[48].** K. Zhou, B. Sisman, R. Liu, and H. Li, “Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset,” in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 920–924.

**[49].** Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, “Torchaudio: Building blocks for audio and speech processing,” arXiv preprint arXiv:2110.15018, 2021.

**[50].** I. Loshchilov and F. Hutter, “Fixing weight decay regularization in adam,” 2018. [Online]. Available: https://openreview.net/forum?id=rk6qdGgCZ

**[51].** J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan et al., “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 4779–4783.

**[52].** J. Kong, J. Kim, and J. Bae, “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis,” Advances in Neural Information Processing Systems, vol. 33, 2020.

**[53].** J. Garofolo, D. Graff, D. Paul, and D. Pallett, “Csr-i (wsj0) complete ldc93s6a,” Web Download. Philadelphia: Linguistic Data Consortium, vol. 83, 1993.

**[54].** S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, “ESPnet: End-to-end speech processing toolkit,” in Proceedings of Interspeech, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

**[55].** D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X- vectors: Robust dnn embeddings for speaker recognition,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5329–5333

**[56].** D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stem- mer, and K. Vesely, “The kaldi speech recognition toolkit,” in IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW- USB.

**[57].** C.-H. Chiang, W.-P. Huang, and H.-y. Lee, “Why we should report the details in subjective evaluation of tts more rigorously,” arXiv preprint arXiv:2306.02044, 2023.

**[58].** T. Hayashi, R. Yamamoto, T. Yoshimura, P. Wu, J. Shi, T. Saeki, Y. Ju, Y. Yasuda, S. Takamichi, and S. Watanabe, “Espnet2-tts: Extending the edge of tts research,” arXiv preprint arXiv:2110.07840, 2021.

**[59].** Y. Lei, S. Yang, J. Cong, L. Xie, and D. Su, “Glow-wavegan 2: high-quality zero-shot text-to-speech synthesis and any-to-any voice conversion,” arXiv preprint arXiv:2207.01832, 2022.

**[60].** D. Lim, S. Jung, and E. Kim, “Jets: Jointly training fastspeech2 and hifi-gan for end to end text to speech,” arXiv preprint arXiv:2203.16852, 2022.

**[61].** Z. Liu, Y. Guo, and K. Yu, “Diffvoice: Text-to-speech with latent diffusion,” in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023, pp. 1–5.

**[62].** L. Van der Maaten and G. Hinton, “Visualizing data using t-sne.” Journal of machine learning research, vol. 9, no. 11, 2008.

**[63].** C. Busso, M. Bulut, S. Narayanan, J. Gratch, and S. Marsella, “Toward effective automatic recognition systems of emotion in speech,” Social emotions in nature and artifact: emotions in human and human-computer interaction, J. Gratch and S. Marsella, Eds, pp. 110–127, 2013. [53]. J. Garofolo, D. Graff, D. Paul, and D. Pallett, “Csr-i (wsj0) complete ldc93s6a,” Web Download. Philadelphia: Linguistic Data Consortium, vol. 83, 1993.

**[64].** M. Chen, X. Tan, Y. Ren, J. Xu, H. Sun, S. Zhao, T. Qin, and T.-Y. Liu, “Multispeech: Multi-speaker text to speech with transformer,” in Proc. Interspeech, 2020, p. pages 4024–4028.

**[65].** Y. A. Li, C. Han, and N. Mesgarani, “Styletts-vc: One-shot voice conversion by knowledge transfer from style-based tts models,” in 2022 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2023, pp. 920–927.

**[66].** S.-w. Park, D.-y. Kim, and M.-c. Joe, “Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data,” arXiv preprint arXiv:2005.03295, 2020.

**[67].** J. An, S. Huang, Y. Song, D. Dou, W. Liu, and J. Luo, “Artflow: Unbiased image style transfer via reversible neural flows,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 862–871.

**[68].** T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” arXiv preprint arXiv:1802.05957, 2018.

**[69].** T. Salimans and D. P. Kingma, “Weight normalization: A simple reparameterization to accelerate training of deep neural networks,” Advances in neural information processing systems, vol. 29, pp. 901–909, 2016.