**ABSTRACT**

Chúng tôi giải quyết nhiệm vụ tạo ra âm nhạc có điều kiện. Chúng tôi giới thiệu MUSICGEN, một Mô hình Ngôn ngữ (LM) đơn lẻ hoạt động trên nhiều luồng biểu diễn âm nhạc rời rạc được nén, tức là các token. Khác với các nghiên cứu trước đây, MUSICGEN bao gồm một LM transformer giai đoạn đơn lẻ cùng với các mẫu xen kẽ token hiệu quả, loại bỏ nhu cầu kết hợp nhiều mô hình, ví dụ như theo cấp cao hoặc tăng mẫu. Chúng tôi chứng minh cách MUSICGEN có thể tạo ra các mẫu chất lượng cao, cả đơn và đa kênh, trong khi được điều kiện hóa bằng mô tả văn bản hoặc đặc trưng nhạc lý, cho phép kiểm soát tốt hơn trên kết quả tạo ra. Chúng tôi tiến hành đánh giá kinh nghiệm chi tiết, bao gồm cả nghiên cứu tự động và con người, chứng minh phương pháp đề xuất vượt trội so với các tiêu chuẩn cơ sở trên một bài kiểm tra văn bản-âm nhạc tiêu chuẩn. Thông qua các nghiên cứu phân tích, chúng tôi làm sáng tỏ tầm quan trọng của từng thành phần trong MUSICGEN

**INTRODUCTION**

Text-to-music là quá trình tạo ra các bản nhạc dựa trên mô tả văn bản, ví dụ như "bài hát rock thập kỷ 90 với giai điệu guitar". Tạo ra nhạc là một nhiệm vụ khó khăn vì nó yêu cầu mô hình hóa các chuỗi dài. Khác với lời nói, âm nhạc yêu cầu sử dụng toàn bộ phổ tần số, điều này đòi hỏi tốc độ lấy mẫu cao hơn. Âm nhạc còn chứa các âm đồng âm và giai điệu từ các nhạc cụ khác nhau, tạo nên cấu trúc phức tạp. Người nghe nhạc rất nhạy cảm với sự không hòa âm [Fedorenko et al., 2012, Norman-Haignere et al., 2019], do đó việc tạo ra nhạc không để lại nhiều không gian cho các lỗi về giai điệu. Cuối cùng, khả năng kiểm soát quá trình tạo ra trong nhiều phương pháp khác nhau như khóa, nhạc cụ, giai điệu, thể loại, v.v. là rất quan trọng đối với những người sáng tạo âm nhạc.

Những tiến bộ gần đây trong việc học biểu diễn âm thanh tự học [Balestriero et al., 2023], mô hình tuần tự [Touvron et al., 2023], và tổng hợp âm thanh [Tan et al., 2021] cung cấp điều kiện để phát triển các mô hình như vậy. Để làm cho việc mô hình hóa âm thanh dễ dàng hơn, các nghiên cứu gần đây đã đề xuất biểu diễn tín hiệu âm thanh dưới dạng nhiều luồng mã thông báo rời rạc đại diện cho cùng một tín hiệu [Défossez et al., 2022]. Điều này cho phép tạo ra âm thanh chất lượng cao và mô hình hóa âm thanh hiệu quả, tuy nhiên, điều này đồng nghĩa với việc mô hình hóa đồng thời nhiều luồng phụ thuộc song song.

Kharitonov và đồng nghiệp [2022], Kreuk và đồng nghiệp [2022] đã đề xuất mô hình hóa nhiều luồng mã thông báo âm thanh song song theo phương pháp trễ, tức là giới thiệu các độ trễ giữa các luồng khác nhau. Agostinelli và đồng nghiệp [2023] đề xuất biểu diễn các đoạn nhạc bằng nhiều chuỗi mã thông báo rời rạc ở mức độ khác nhau và mô hình chúng bằng một hệ thống các mô hình tự hồi quy theo tầng. Song song với đó, Donahue và đồng nghiệp [2023] áp dụng phương pháp tương tự nhưng cho nhiệm vụ tạo ra bản nhạc hòa âm cho việc hát. Gần đây, Wang và đồng nghiệp [2023] đề xuất giải quyết vấn đề này theo hai giai đoạn: (i) mô hình hóa giai đoạn đầu tiên stream of tokens only; (ii) then, apply a post-network to jointly model the rest of the streams in a non-autoregressive manner.

Trong nghiên cứu này, chúng tôi giới thiệu MUSIGEN, một mô hình tạo ra âm nhạc đơn giản và có thể điều khiển, có khả năng tạo ra âm nhạc chất lượng cao dựa trên mô tả văn bản. Chúng tôi đề xuất một khung công việc tổng quát để mô hình hóa nhiều luồng song song của các đơn vị âm thanh, là một sự tổng quát hóa của các nghiên cứu trước đó. Chúng tôi cũng thực hiện một đánh giá chi tiết về MUSIGEN và cho thấy phương pháp đề xuất vượt trội so với các phương pháp cơ sở, với một đánh giá chủ quan là 84,8 trên 100 cho MUSIGEN so với 80,5 cho phương pháp cơ sở tốt nhất.

Đóng góp của chúng tôi là: (i) Chúng tôi giới thiệu một mô hình đơn giản và hiệu quả để tạo ra âm nhạc chất lượng cao với tần số 32 kHz. Chúng tôi chứng minh rằng MUSICGEN có thể tạo ra âm nhạc nhất quán bằng một mô hình ngôn ngữ giai đoạn duy nhất thông qua một chiến lược xen kẽ mã code hiệu quả. (ii) Chúng tôi trình bày một mô hình duy nhất để thực hiện cả tạo âm dựa trên văn bản và giai điệu, và chứng minh rằng âm thanh được tạo ra phù hợp với giai điệu đã cho và trung thành với thông tin điều kiện văn bản. (iii) Chúng tôi cung cấp đánh giá mục tiêu và đánh giá của con người về các lựa chọn thiết kế quan trọng trong phương pháp của chúng tôi.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Nhiều màu sắc

Mô tả được tạo tự động

Hình 1: Mô hình tương tác mã sách mã hóa được trình bày trong Phần 2.2. Mỗi bước thời gian t1, t2, ..., tn bao gồm 4 giá trị đã được định lượng (tương ứng với k1, ..., k4). Khi thực hiện mô hình tự hồi quy, chúng ta có thể làm phẳng hoặc xen kẽ chúng theo nhiều cách khác nhau, tạo ra một chuỗi mới với 4 luồng song song và các bước s1, s2, ..., sm. Tổng số bước chuỗi S phụ thuộc vào mẫu và số bước ban đầu T. 0 là một mã thông báo đặc biệt chỉ định các vị trí trống trong mẫu.

**II** **METHOD**

MUSICGEN là một mô hình decoder dựa trên transformer autoregressive [Vaswani et al., 2017], được điều kiện dựa trên biểu diễn văn bản hoặc giai điệu. Mô hình này hoạt động trên các đơn vị đã được mã hóa từ EnCodec [Défossez et al., 2022], một công cụ phân tích âm thanh, để tái tạo âm thanh chất lượng cao từ biểu diễn rời rạc với tốc độ khung hình thấp. Các mô hình nén như [Défossez et al., 2022, Zeghidour et al., 2021] sử dụng Residual Vector Quantization (RVQ) và tạo ra nhiều luồng song song, mỗi luồng bao gồm các mã thông báo rời rạc từ các codebook đã học.

AUDIO TOKENIZATION

Trong quá trình mã hóa âm thanh, chúng tôi sử dụng EnCodec, một mô hình tự mã hóa tích chập với không gian ẩn được lượng tử hóa bằng phương pháp Residual Vector Quantization (RVQ) và mất mát tái tạo đối địch. Đầu vào là một biến ngẫu nhiên âm thanh tham chiếu X với độ dài d và tốc độ mẫu fs. EnCodec mã hóa nó thành một tensor liên tục với tốc độ khung fr nhỏ hơn fs, sau đó lượng tử hóa thành Q với K chuỗi mã và M kích thước codebook. Sau lượng tử hóa, chúng ta có K chuỗi mã rời rạc song song, mỗi chuỗi có độ dài T đại diện cho mẫu âm thanh. Trong RVQ, mỗi bộ mã hóa mã hóa lỗi lượng tử hóa còn lại từ bộ mã hóa trước đó, do đó các giá trị lượng tử hóa cho các codebook khác nhau không độc lập và codebook đầu tiên quan trọng nhất.

**Codebook interleaving patterns**

**Exact flattened autoregressive decomposition.**Phân rã tự hồi quy phẳng chính xác: Mô hình tự hồi quy yêu cầu một chuỗi ngẫu nhiên rời rạc U ∈ {1, . . . , N }S với S là độ dài chuỗi. Chúng ta có thể xây dựng một chuỗi thứ hai của biến ngẫu nhiên ̃U bằng cách sử dụng mật độ tự hồi quy p, và nếu chúng ta có thể đạt được một ước lượng hoàn hảo ˆp của p bằng mô hình học sâu, thì chúng ta có thể khớp chính xác phân phối của U. Một cách khác để giải quyết vấn đề với biểu diễn Q là làm phẳng Q, tuy nhiên điều này tăng độ phức tạp và mất một phần của tốc độ mẫu thấp fr.

**Arbitrary codebook interleaving patterns** Mô hình các mẫu xen kẽ codebook tùy ý được giới thiệu để nghiên cứu các phân rã khác nhau và đo chính xác tác động của việc sử dụng một phân rã không chính xác. Các mẫu codebook là một chuỗi các tập hợp các cặp bước thời gian và chỉ số codebook, được chia thành các phần tử con. Chúng ta có thể định nghĩa nhiều phân rã khác nhau, ví dụ như mẫu "song song" hoặc mẫu "trễ", và thông qua đánh giá thực nghiệm, chúng ta có thể hiểu được lợi ích và hạn chế của các mẫu codebook khác nhau và sự quan trọng của việc mô hình chính xác các chuỗi codebook song song.

MODEL CONDITIONING

**Text conditioning** Trong việc tạo âm thanh có điều kiện, chúng ta sử dụng một tensor điều kiện C được tính toán từ mô tả văn bản tương ứng với âm thanh đầu vào X. Có ba phương pháp chính để biểu diễn văn bản cho việc tạo âm thanh có điều kiện, bao gồm việc sử dụng bộ mã hóa văn bản đã được huấn luyện trước như T5, sử dụng các mô hình ngôn ngữ dựa trên hướng dẫn, và sử dụng biểu diễn chung văn bản-âm thanh như CLAP. Trong nghiên cứu này, chúng tôi thử nghiệm cả ba phương pháp trên: bộ mã hóa T5, FLAN-T5 và CLAP.

**Melody conditioning** là một phương pháp tiếp cận tự nhiên hơn trong mô hình tạo ra âm nhạc có điều kiện, so với việc sử dụng văn bản. Phương pháp này cho phép điều chỉnh cấu trúc giai điệu bằng cách kết hợp thông tin từ một bản nhạc khác hoặc cả việc huýt sáo hoặc hát. Để giảm hiện tượng overfitting, chúng tôi sử dụng một cách tiếp cận không giám sát bằng cách chọn tần số thời gian quan trọng nhất trong mỗi bước thời gian của đầu vào.

MODEL ARCHITECTURE **Codebook projection and positional embedding** Trong mô hình, khi có một mẫu codebook, chỉ một số codebook xuất hiện ở mỗi bước mẫu. Chúng ta lấy các giá trị tương ứng từ Q dựa trên các chỉ số trong mẫu. Nếu codebook xuất hiện, chúng ta sử dụng một bảng nhúng đã học với N mục nhập và kích thước D để biểu diễn giá trị liên quan từ Q. Nếu không, chúng ta sử dụng một mã thông báo đặc biệt chỉ ra sự vắng mặt của nó. Sau quá trình biến đổi này, chúng ta tổng hợp đóng góp từ mỗi codebook. Với P0 = ∅, đầu vào đầu tiên luôn là tổng của tất cả các mã thông báo đặc biệt. Cuối cùng, chúng ta tổng hợp một nhúng hình sin để mã hóa bước hiện tại s [Vaswani et al., 2017].

**Transformer decoder**. Trình giải mã Transformer nhận đầu vào từ một mạng Transformer với L lớp và kích thước D. Mỗi lớp bao gồm một khối tự-chú ý theo hướng gây ra tác động. Sau đó, chúng ta sử dụng một khối chú ý chéo được cung cấp với tín hiệu điều kiện C. Khi sử dụng điều kiện giai điệu, chúng ta thay vào đó cung cấp tensor điều kiện C như một tiền tố cho đầu vào của Transformer. Lớp kết thúc với một khối kết nối đầy đủ bao gồm một lớp tuyến tính từ D đến kênh 4·D, một ReLU, và một lớp tuyến tính trở lại kênh D. Các khối chú ý và kết nối đầy đủ được bao bọc bằng một kết nối trượt dư. Chuẩn hóa lớp [Ba et al., 2016] được áp dụng cho mỗi khối trước khi được tổng hợp với kết nối trượt dư ("pre-norm").

**Logits prediction** Ở bước Ps, đầu ra từ bộ giải mã transformer được chuyển đổi thành dự đoán logits cho các giá trị của Q tại các chỉ số được cho bởi Ps+1. Mỗi codebook chỉ xuất hiện tối đa một lần trong Ps+1. Nếu một codebook xuất hiện, dự đoán logits được thu được bằng cách áp dụng một lớp tuyến tính cụ thể cho codebook từ D kênh đến N kênh.

**III. EXPERIMENT SETUP**

MODELS AND HYPERPARAMETERS

**Audio tokenization model** Mô hình mã hóa âm thanh EnCodec không gây ảnh hưởng ngược với năm lớp không gian cho âm thanh đơn âm 32 kHz, với bước nhảy là 640, tạo ra tốc độ khung hình là 50 Hz và kích thước ẩn ban đầu là 64, tăng gấp đôi ở mỗi lớp của mô hình. Các nhúng được rời rạc hóa bằng một RVQ với bốn bộ lượng tử, mỗi bộ có kích thước bảng mã là 2048. Chúng tôi tuân theo phương pháp của Défossez et al. [2022] để huấn luyện mô hình trên các đoạn âm thanh một giây được cắt ngẫu nhiên trong chuỗi âm thanh.

**Transformer model** Mô hình Transformer được huấn luyện với các kích thước khác nhau: 300 triệu, 1.5 tỷ và 3.3 tỷ tham số. Chúng sử dụng một phương pháp chú ý hiệu quả về bộ nhớ gọi là Flash attention từ gói xFormers để cải thiện tốc độ và sử dụng bộ nhớ tốt hơn với các chuỗi dài. Mô hình được huấn luyện trên các đoạn âm thanh kéo dài 30 giây được lấy mẫu ngẫu nhiên từ bản gốc và sử dụng các kỹ thuật tối ưu hóa như AdamW và D-Adaptation để cải thiện hiệu suất và độ ổn định.

**Text processing** Trong quá trình tiền xử lý văn bản, Kreuk và đồng nghiệp (2022) đã đề xuất một phương pháp chuẩn hóa văn bản, trong đó các từ dừng được loại bỏ và phần còn lại của văn bản được lemmatized. Phương pháp này được gọi là "text-normalization". Khi xem xét các tập dữ liệu âm nhạc, thông thường có sẵn các nhãn chú thích bổ sung như khóa âm nhạc, nhịp độ, loại nhạc cụ, vv. Chúng tôi cũng thử nghiệm việc ghép các nhãn chú thích này vào mô tả văn bản. Phương pháp này được gọi là "condition-merging". Cuối cùng, chúng tôi khám phá việc sử dụng "word dropout" như một chiến lược tăng cường văn bản khác. Đối với các mô hình cuối cùng, chúng tôi sử dụng "condition-merging" với xác suất là 0.25. Sau khi ghép, chúng tôi áp dụng "text description dropout" với xác suất là 0.5. Chúng tôi sử dụng "word dropout" với xác suất là 0.3 trên văn bản kết quả. Một so sánh đầy đủ về các chiến lược tiền xử lý văn bản khác nhau có thể được tìm thấy trong Phụ lục A.2.

**Codebook patterns and conditioning.** Trong phần này, chúng ta sử dụng mô hình mã hóa văn bản T5 để điều kiện văn bản, có thể kết hợp với việc điều kiện theo giai điệu. Chúng ta cũng thử nghiệm với các mô hình mã hóa văn bản khác như FLAN-T5 và CLAP để so sánh hiệu suất của mô hình MUSICGEN. Đối với điều kiện theo giai điệu, chúng ta tính toán chromagram với kích thước cửa sổ là 214 và bước nhảy là 212. Chúng ta sử dụng một cách tiếp cận tương tự như Kreuk et al. để thực hiện việc lấy mẫu từ logits của mô hình. Cụ thể, trong quá trình huấn luyện, chúng ta loại bỏ điều kiện với xác suất là 0.2 và trong quá trình suy luận, chúng ta sử dụng một tỷ lệ hướng dẫn là 3.0.

DATASETS

**Training datasets.** Để huấn luyện MUSICGEN, chúng ta sử dụng 20.000 giờ âm nhạc được cấp phép. Cụ thể, chúng ta sử dụng một bộ dữ liệu nội bộ gồm 10.000 bản nhạc chất lượng cao và các bộ sưu tập âm nhạc ShutterStock và Pond5 với lần lượt 25.000 và 365.000 bản nhạc chỉ chứa nhạc cụ. Các bộ dữ liệu này bao gồm âm nhạc đầy đủ với tần số lấy mẫu 32 kHz và có thông tin mô tả văn bản như thể loại, BPM và các từ khóa. Chúng ta chuyển đổi âm thanh thành đơn âm nếu không có yêu cầu khác.

**Evaluation datasets** Để đánh giá phương pháp đề xuất và so sánh với các công trình trước đây, chúng tôi sử dụng bộ kiểm tra MusicCaps [Agostinelli et al., 2023]. MusicCaps bao gồm 5.5K mẫu (mỗi mẫu dài 10 giây) được chuẩn bị bởi các nhạc sĩ chuyên gia và một tập con 1K cân bằng theo thể loại. Chúng tôi báo cáo các chỉ số mục tiêu trên tập không cân bằng, trong khi chúng tôi lấy mẫu các ví dụ từ tập cân bằng theo thể loại để đánh giá âm nhạc từ văn bản. Về nghiên cứu các yếu tố như giai điệu, tác giả sử dụng một tập dữ liệu chứa 528 bản nhạc không trùng lặp với tập dữ liệu huấn luyện. Các mẫu âm nhạc được sử dụng để đánh giá chất lượng tổng thể và sự phù hợp với văn bản đầu vào.

**Bảng 1:** Tạo âm nhạc từ văn bản. Chúng tôi so sánh các chỉ số khách quan và chủ quan cho MUSICGEN so với một số cơ sở. Chúng tôi báo cáo cả điểm trung bình và khoảng tin cậy CI95. Mô hình Mousai được đào tạo lại trên cùng tập dữ liệu, trong khi đối với MusicLM, chúng tôi sử dụng API công khai cho các nghiên cứu về con người. Chúng tôi báo cáo FAD ban đầu trên MusicCaps cho Noise2Music và MusicLM. "MUSICGEN w. random melody" đề cập đến MUSICGEN được đào tạo với chromagram và văn bản. Khi đánh giá, chúng tôi lấy mẫu chromagram ngẫu nhiên từ một tập dữ liệu giữ lại.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

EVALUATION

**Baseline** Trong việc so sánh với hai mô hình cơ sở (Riffusion và Mousai) cho việc tạo âm nhạc từ văn bản, MUSICGEN được đào tạo trên tập dữ liệu riêng để có sự so sánh công bằng. Ngoài ra, khi có thể, nó cũng được so sánh với MusicLM và Noise2Music.

**Evaluation Metrics** Chúng tôi đánh giá phương pháp đề xuất bằng cách sử dụng các chỉ số mục tiêu và chủ quan. Đối với các phương pháp mục tiêu, chúng tôi sử dụng ba chỉ số: Khoảng cách âm thanh Fréchet (FAD), Độ lệch Kullback-Leiber (KL) và điểm CLAP (CLAP). Chúng tôi báo cáo điểm FAD sử dụng mô hình VGGish trong Tensorflow. Điểm FAD thấp cho thấy âm thanh được tạo ra là hợp lý. Đồng thời, chúng tôi sử dụng mô hình phân loại âm thanh tiên tiến để tính toán độ lệch KL giữa các nhãn của âm nhạc gốc và âm nhạc được tạo ra. Điểm CLAP được tính toán để đo sự phù hợp giữa mô tả bài hát và âm thanh được tạo ra. Đối với các nghiên cứu về con người, chúng tôi yêu cầu các đánh giá viên đánh giá chất lượng tổng thể và sự liên quan đến nội dung văn bản của các mẫu âm thanh.

Bảng 2: Chúng tôi báo cáo độ tương đồng cosine giữa giai điệu tham chiếu và giai điệu được tạo ra (SIM.) và các chỉ số chủ quan bao gồm sự phù hợp với giai điệu (MEL.). Tất cả kết quả được báo cáo cho MUSICGEN 1.5B.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

**IV RESULT**

Bài viết trình bày kết quả của phương pháp đề xuất trong việc tạo ra âm nhạc từ văn bản và so sánh với các nghiên cứu trước đó trong lĩnh vực này. Tiếp theo, phương pháp đề xuất được đánh giá về khả năng tạo ra âm nhạc dựa trên đặc điểm giai điệu. Bài viết cũng đề cập đến việc mở rộng mẫu mã âm thanh cho việc tạo ra âm thanh stereo và kết luận với một nghiên cứu về tác động của các yếu tố cụ thể.

COMPARISION WITH THE BASELINES

Bảng 1 trình bày sự so sánh giữa phương pháp đề xuất và các phương pháp Mousai, Riffusion, MusicLM và Noise2Music. Do không có phiên bản chính thức của Noise2Music hoặc mô hình đã được huấn luyện trước, chúng tôi chỉ báo cáo FAD trên MusicCaps, được báo cáo trong tài liệu Noise2Music. Tương tự, việc triển khai MusicLM không công khai. Kết quả cho thấy MUSICGEN hoạt động tốt hơn so với các phương pháp cơ sở được đánh giá bởi người nghe, cả về chất lượng âm thanh và sự tuân thủ với mô tả văn bản được cung cấp. Noise2Music cho kết quả tốt nhất về FAD trên MusicCaps, tiếp theo là MUSICGEN được huấn luyện với điều kiện văn bản.

MELODY EVALUATION

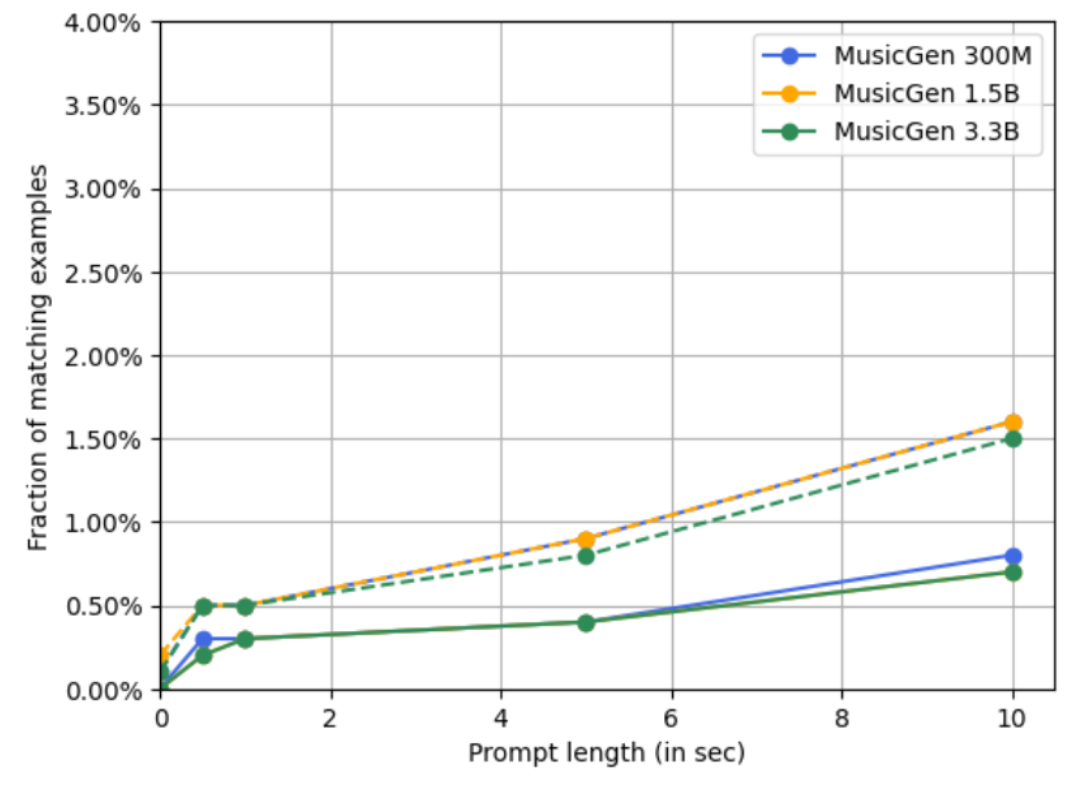
Chúng tôi đánh giá MUSICGEN, được điều kiện chung trên biểu diễn văn bản và giai điệu, bằng cách sử dụng các chỉ số mục tiêu và chủ quan trên tập dữ liệu đánh giá. Đối với đánh giá mục tiêu, chúng tôi giới thiệu một chỉ số mới: độ tương đồng cosine của chroma, đo lường sự tương đồng trung bình giữa các khung tương ứng với cùng các bước thời gian, được lấy từ chroma đã được định lượng của mẫu tham chiếu và mẫu được tạo ra. Kết quả cho thấy rằng MUSICGEN được đào tạo với điều kiện chromagram thành công trong việc tạo ra âm nhạc theo giai điệu đã cho, đồng thời cho phép kiểm soát tốt hơn về đầu ra được tạo ra. Đáng chú ý, MUSICGEN có tính ổn định khi loại bỏ chroma trong quá trình suy luận, với cả OVL. và REL. không thay đổi đáng kể.

Hình 2: Codebook stereo (trái) và kết quả ghi nhớ (phải) là một hình ảnh trong bài viết, hiển thị các codebook stereo và kết quả ghi nhớ. Codebook stereo là một tập hợp các mẫu âm thanh được sử dụng trong mô hình stereo, trong khi kết quả ghi nhớ đo lường khả năng ghi nhớ của mô hình trên các mẫu âm thanh.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Nhiều màu sắc

Mô tả được tạo tự động

(a) Hình ảnh hóa các mẫu codebook cho các mô hình stereo với hai cách xen kẽ có thể của codebook trái (màu xanh) và codebook phải (màu đỏ). Đối với một chỉ số codebook cụ thể, mẫu "stereo delay" sử dụng các độ trễ khác nhau cho các kênh trái và phải, trong khi mẫu "stereo partial delay" dự đoán hai kênh song song nhau.



(b) Kết quả ghi nhớ cho các mã codebook đầu tiên trong việc tạo ra âm thanh trong 5 giây, xem xét sự khớp chính xác (đường liền) và sự khớp một phần 80% (đường đứt) khi được yêu cầu với các đoạn trích có độ dài khác nhau từ tập dữ liệu huấn luyện.

Bảng 3: Tạo âm nhạc từ văn bản đa âm. EnCodec xử lý riêng lẻ các kênh trái và phải, cho chúng ta 8 bộ mã thay vì 4. Chúng tôi thử nghiệm hai mẫu mã codebook, được miêu tả trong Hình 2a. Chúng tôi cũng đo lường một trong các mô hình âm nhạc đa âm sau khi được trộn xuống mono. Chúng tôi sử dụng một mô hình MUSICGEN 1.5B được điều kiện chỉ bằng văn bản.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

FINE-TUNING FOR STEREOPHONIC GENERATION

Chúng tôi thực hiện thí nghiệm để mở rộng khả năng tạo ra âm thanh stereo. Chúng tôi sử dụng cùng một công cụ mã hóa EnCodec, được áp dụng độc lập cho các kênh trái và phải, cung cấp 8 codebook cho mỗi khung hình. Chúng tôi tinh chỉnh mô hình MUSICGEN đã được huấn luyện trước đó cho âm thanh stereo và sử dụng mẫu mã "delay" để tạo ra âm thanh stereo mà không tốn thêm chi phí tính toán. Kết quả đánh giá cho thấy âm thanh stereo được đánh giá cao hơn so với âm thanh mono và mẫu mã "stereo partial delay" có hiệu suất tốt hơn so với mẫu mã "stereo delay" cả về chất lượng tổng thể và liên quan đến văn bản.

ABALATION

Phần này cung cấp một nghiên cứu về các mẫu codebook khác nhau, kết quả cho các quy mô mô hình và một nghiên cứu về việc ghi nhớ. Ngoài ra, chúng tôi trình bày kết quả cho các chiến lược tăng cường văn bản khác nhau, bộ mã hóa văn bản và các mô hình mã hóa âm thanh khác nhau trong Phụ lục A.2. Các thí nghiệm được thực hiện trên 1.000 mẫu trong 30 giây, được lấy ngẫu nhiên từ tập đánh giá.

**The effect of the codebook interleaving patterns** Hiểu nôm na, trong đoạn văn này, tác giả đang nói về việc đánh giá và so sánh các mẫu mã codebook khác nhau trong quá trình tạo âm thanh stereo. Các mẫu mã này bao gồm "delay", "partial delay", "parallel", "coarse first" và "flattening". Bảng 4 và bảng 5 trong đoạn văn cung cấp kết quả đánh giá mục tiêu và chủ quan cho từng mẫu mã codebook.

Bảng 4 so sánh các mẫu xen kẽ codebook khác nhau trên các chuỗi âm thanh kéo dài 30 giây. Mẫu "flattening" đạt được điểm số tốt nhất, trong khi mẫu "delay" và "partial flattening" đạt điểm số tương đương, còn mẫu "parallel" đạt điểm số kém hơn.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Bảng 5: Quy mô mô hình. Chúng tôi so sánh 3 quy mô cho phương pháp của chúng tôi và đánh giá nó trên một bộ kiểm tra nội bộ để giới hạn tác động của các vấn đề dự đoán ngoài lĩnh vực mà chúng tôi quan sát được với MusicCaps. Theo các chỉ số mục tiêu, chúng tôi quan sát được sự cải thiện liên tục của các chỉ số, mặc dù chất lượng chủ quan không còn cải thiện ở 1.5 tỷ tham số. Tuy nhiên, mô hình 3.3 tỷ tham số có vẻ phù hợp hơn với yêu cầu văn bản.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

**The effect of model size** Hiệu ứng của kích thước mô hình. Bảng 5 báo cáo kết quả cho các kích thước mô hình khác nhau, bao gồm 300 triệu, 1.5 tỷ và 3.3 tỷ tham số. Như dự đoán, tăng kích thước mô hình dẫn đến điểm số tốt hơn, nhưng điều này đồng nghĩa với thời gian đào tạo và suy luận lâu hơn. Đối với đánh giá chủ quan, chất lượng tổng thể tốt nhất ở mức 1.5 tỷ, nhưng một mô hình lớn hơn có thể hiểu tốt hơn yêu cầu văn bản.

**Memorization experiment** Thực hiện thí nghiệm ghi nhớ, theo [Agostinelli et al., 2023], chúng tôi phân tích khả năng ghi nhớ của MUSICGEN. Chúng tôi chỉ xem xét luồng đầu tiên của các cuốn sách mã từ MUSICGEN vì nó chứa thông tin tổng quát hơn. Chúng tôi chọn ngẫu nhiên N = 20.000 ví dụ từ tập huấn luyện và cho mỗi ví dụ, chúng tôi cung cấp mô hình một đầu vào là các cuốn sách mã EnCodec tương ứng với âm thanh gốc và thông tin điều kiện. Chúng tôi tạo ra một phần tiếp theo gồm 250 mã âm thanh (âm thanh dài 5 giây) bằng cách sử dụng giải mã theo cách tham lam. Chúng tôi báo cáo các kết quả trùng khớp chính xác và trùng khớp một phần cho các kích thước mô hình khác nhau khi thay đổi độ dài của đầu vào âm thanh.