UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

PEDRO LUCAS TOMAZ NEVES

# Geração e análise de músicas com base em dados de classificação emocional usando aprendizado de máquinas

Campinas

2022

Pedro Lucas Tomaz Neves

# Geração e análise de músicas com base em dados de classificação emocional usando aprendizado de máquinas

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Matemática Aplicada.

Orientador: João Batista Florindo

Coorientador: Jose Eduardo Fornari Novo Junior

Este trabalho corresponde à versão final da Dissertação defendida pelo aluno Pedro Lucas Tomaz Neves e orientada pelo Prof. Dr. João Batista Florindo.

Campinas

2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Silvania Renata de Jesus Ribeiro - CRB 8/6592

Neves, Pedro Lucas Tomaz, 1998-
N414g Geração e análise de músicas com base em dados de classificação emocional usando aprendizado de máquinas / Pedro Lucas Tomaz Neves. – Campinas, SP : [s.n.], 2022.

Orientador: João Batista Florindo.
Coorientador: Jose Eduardo Fornari Novo Junior.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Aprendizado de máquina. 2. Redes adversárias generativas. 3. Composição (Música). I. Florindo, João Batista, 1984-. II. Novo Junior, Jose Eduardo Fornari, 1966-. III. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. IV. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Music generation and analysis based on emotional classification data using machine learning
**Palavras-chave em inglês:**
Machine learning
Generative adversarial networks
Composition (Music)
**Área de concentração:** Matemática Aplicada
**Titulação:** Mestre em Matemática Aplicada
**Banca examinadora:**
João Batista Florindo [Orientador]
Marcos Eduardo Ribeiro do Valle Mesquita
Tiago Fernandes Tavares
**Data de defesa:** 21-03-2022
**Programa de Pós-Graduação:** Matemática Aplicada

**Identificação e informações acadêmicas do(a) aluno(a)**
- ORCID do autor: https://orcid.org/0000-0002-8902-5743
- Currículo Lattes do autor: http://lattes.cnpq.br/9767316862287321

**Dissertação de Mestrado defendida em 21 de março de 2022 e aprovada pela banca examinadora composta pelos Profs. Drs.**

**Prof(a). Dr(a). JOÃO BATISTA FLORINDO**

**Prof(a). Dr(a). MARCOS EDUARDO RIBEIRO DO VALLE MESQUITA**

**Prof(a). Dr(a). TIAGO FERNANDES TAVARES**

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

# Agradecimentos

Ao meu orientador, Prof. Dr. João Batista Florindo e co-orientador, Dr. José Eduardo Fornari Novo Junior, pelo apoio, sugestões, correções e encorajamento ao longo da pesquisa.

À minha família, pelo imprescindível apoio no decorrer de toda a minha vida.

Aos meus amigos, pelas discussões valiosas e incontáveis boas memórias.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo financiamento do projeto (processo 132641/2020-8).

*"A Inteligência Artificial tem a capacidade de compor uma boa música, mas não uma grande música." (Nick Cave)*

# Resumo

As Redes Generativas Adversárias (GANs) se tornaram conhecidas dentro do *Deep Learning* graças aos resultados impressionantes que alcançaram na área de modelagem generativa. Essas constituem uma estrutura na qual duas redes neurais interagem entre si de forma a otimizar um modelo gerador que produz dados sintéticos semelhantes a algum tipo de conteúdo real. É possível guiar o processo de criação através de informação condicional, escolhendo a qual dos modos da distribuição original devem pertencer as amostras sintetizadas.

Ao longo deste trabalho, foram desenvolvidas e avaliadas GANs voltadas para a produção automática de música. O objetivo principal da pesquisa foi a construção de um modelo capaz de sintetizar trechos musicais condicionados por estados emocionais específicos de acordo com sua representação nas dimensões de *Valence* e *Arousal*. Durante o projeto, no entanto, surgiram algumas perspectivas que viabilizaram o desenvolvimento de outros sistemas generativos. Os modelos foram avaliados por meio de métricas automáticas disponíveis na literatura, e o último deles também contou com a avaliação humana. Os experimentos executados deixam claro que o uso de GANs pode trazer resultados promissores para a criação de modelos computacionais de composição musical automática.

**Palavras-chave**: Aprendizado de Máquinas. Redes Generativas Adversárias. Composição Musical Automática.

# Abstract

Generative Adversarial Networks (GANs) have become known within Deep Learning thanks to the impressive results that they achieve in the field of generative modeling. GANs are a framework in which two neural networks interact with each other in order to optimize a generative model that produces synthetic data similar to some form of real content. It is possible to guide the generative process via conditional information, which is equivalent to choosing from which modes of the original distribution the synthesized samples should come.

This work focused on the development and evaluation of music generating GANs. The main objective was to construct a model capable of synthesizing musical excerpts conditioned by specific emotional states according to their representations on the Valence and Arousal Dimensions. However, some results obtained during the research allowed for the development of other generative systems. The models were evaluated via automatic metrics avalilable in the literature, and the last one also recieved human evaluation. These evaluations suggest that promising results can be achieved when GANs are applied to the task of automatic music generation.

**Keywords**: Machine Learning. Generative Adversarial Networks. Automatic Music Composition.

# Lista de ilustrações

# Lista de tabelas

# Lista de abreviaturas e siglas

| | |
|---|---|
| GAN | General Adversarial Network |
| MIR | Music Information Retrieval |
| CMAP | Composição Musical Assistida por Computador |
| CMA | Composição Musical Automática |
| RNN | Recurrent Neural Network |
| LSTM | Long-Short Term Memory |
| CNN | Convolutional Neural Network |
| IS | Inception Score |
| FID | Frechét Inception Distance |
| MLP | Multi Layer Perceptron |

# Lista de símbolos

| | |
|---|---|
| $\|\cdot\|$ | Norma-p |
| $\nabla f$ | Gradiente da função $f$ |
| $\mathbb{E}_{x\sim p(x)} f(x)$ | Valor esperado de $f$ em relação a $x$ retirado da distribuição $p(x)$ |

# Lista de Algoritmos

# Sumário

# Introdução

O Aprendizado de Máquinas possibilita a resolução automática de alguns problemas que anteriormente requisitavam considerável trabalho humano, como segmentação de imagens médicas (TAJBAKHSH et al., 2020), classificação de imagens (HE et al., 2015), reconhecimento de fala (RAVANELLI et al., 2020), e muitos outros. De forma geral, essa área consiste em um amplo conjunto de técnicas e algoritmos, como por exemplo as redes neurais (BISHOP, 2006), que "aprendem" a reconhecer determinados padrões a partir da experiência de um agente externo (normalmente um especialista da área de aplicação) ou da própria estrutura dos dados, e que através disso tentam solucionar um problema desejado. Dentre os modelos utilizados nessa área, destacam-se aqui aqueles baseados em algoritmos de Aprendizado Profundo ("*Deep Learning*"), modelos estes tipicamente compostos por um grande número de camadas de aprendizagem, correspondendo, geralmente, a níveis diferentes de abstração.

Uma das classes de algoritmos pertencentes ao ramo do Aprendizado de Máquinas são os Modelos Generativos, que buscam capturar uma distribuição de dados, sendo capazes de produzir novos membros desta distribuição. Redes Generativas Adversárias (GANs, de Generative Adversarial Networks) (GOODFELLOW et al., 2014) constituem uma estrutura algorítmica em que duas redes neurais competem de modo a otimizar um modelo generativo. Enquanto uma das redes, denominada Geradora, sintetiza exemplos que se assemelhem o máximo possível com aqueles vindos de uma determinada distribuição de dados reais, a outra, chamada de Discriminatória, cumpre o papel de discernir entre os elementos reais a aqueles produzidos pela outra rede. A analogia normalmente invocada para explicar este processo é a de um falsificador (a rede Geradora) que tenta enganar a polícia (a rede Discriminatória), convencendo-a de que suas notas falsas são legítimas, e esta segunda tenta impedir a primeira de executar o crime.

As áreas de Computação Musical Automática (CMA) (NIERHAUS, 2009) e de Composição Musical Auxiliada por Computador (CMAC) (ASSAYAG et al., 1999), esta última que consiste no uso de processo computacionais no processo de composição, vêm sendo exploradas há muitas décadas. Um grande leque de estratégias já foi utilizado para desempenhá-las, partindo de sistemas baseados em regras matemáticas rígidas ou processos estocásticos àqueles criados a partir de modelos genéticos. Dentro dessas áreas, modelos de aprendizagem, ou seja, algoritmos de Aprendizado de Máquinas possuem grande relevância, e naturalmente a popularização dos sistemas de *Deep Learning* trouxe um avanço expressivo para a área.

De forma geral, no contexto do *Deep Learning*, a informação musical é repre-

sentada de uma das duas maneiras: a partir de *waveforms* retiradas do áudio digital, como de arquivos .wav e .mp3, ou através de representações simbólicas derivadas de arquivos MIDI. Portanto, os modelos que trabalham com música dividem-se em dois grupos de acordo com a representação escolhida. Para áudio digital, utilizam-se redes semelhantes àquelas empregadas em processamento de fala, como Wavenets (OORD et al., 2016). Já no caso das representações simbólicas, opta-se por arquiteturas comumente aplicadas na área de Processamento de Linguagem Natural, como Transformers (VASWANI et al., 2017) e LSTMs (HOCHREITER; SCHMIDHUBER, 1997). GANs, assim como outros algoritmos de Aprendizado de Máquinas, a princípio são capazes de trabalhar com quaisquer tipos de dados. No entanto, até o momento do desenvolvimento deste trabalho, a maior parte dos esforços na área foi voltada para a Visão Computacional, com muitos modelos já sendo capazes de sintetizar imagens naturais ou faces humanas muito semelhantes a exemplos reais (BROCK; DONAHUE; SIMONYAN, 2018; KARRAS et al., 2020). Portanto, independentemente da representação escolhida para a música, seja ela simbólica ou acústica, ainda são muitas as dificuldades envolvidas na modelagem generativa musical com o uso de GANs.

De fato, modelar e gerar áudio no domínio *waveform* é um desafio que apenas recentemente passou a ser computacionalmente viável (DIELEMAN; OORD; SIMONYAN, 2018; BRIOT; HADJERES; PACHET, 2019), e ainda não há muitos modelos baseados em GANs que realizam tais tarefas. O motivo principal por trás desse fato é o enorme número de amostras por segundo necessárias para a composição do áudio digital, fato esse que limita o campo receptivo de modelos trabalhando com dados dessa natureza.

Além do mais, apesar de já existir uma grande quantidade de modelos que trabalham com representações simbólicas de música, poucos desses são baseados em Redes Generativas Adversárias, já que existem dificuldades teóricas relacionadas à adaptação de GANs para dados discretos (YU et al., 2017; JANG; GU; POOLE, 2017).

Dado isso, e levando-se também em consideração o sucesso das GANs em outras áreas, principalmente a da Computação Visual, há uma série de resultados positivos que ainda podem ser alcançados através do uso dessa estrutura nos contextos de Computação Musical Assistida por Computador e de Computação Musical.

## Objetivos

O objetivo principal deste trabalho foi o desenvolvimento de um modelo de aprendizado de máquinas baseado em GANs e capaz de gerar excertos musicais representando estados emocionais de acordo com o modelo circumplexo de Russell (RUSSELL, 1980). Para o treinamento do modelo, foram usadas passagens musicais das quais foram tomados dados de *valence* e *arousal*, de acordo com a percepção de humanos que escutaram

essas amostras (*ground truth*). Dessa forma, os modelos tornaram-se capazes de realizar geração musical condicionada, em que uma rede generativa produz conteúdo musical com base em dados fornecidos por um usuário. Adicionalmente, durante a execução do trabalho, foram explorados também modelos generativos de áudio - um deles capaz de sintetizar passagens musicais curtas condicionadas por instrumento, assim como um modelo capaz de transformar áudio para uma representação intermediária altamente comprimida mas semanticamente densa. Essas etapas serviram como base para os desenvolvimentos finais do projeto.

A análise dos modelos foi realizada em duas frentes: em primeiro lugar, aplicaram-se métricas quantitativas disponíveis na literatura de MIR (*Music Information Retrieval)* para que fosse averiguada a qualidade das amostras produzidas pelos modelos. Esta análise foi feita para os três modelos propostos. Posteriormente, para o último modelo, foi feita uma pesquisa onde participantes responderam a perguntas relacionadas tanto à qualidade quanto à capacidade das amostras de expressarem o conteúdo emocional fornecido condicionalmente.

Levando-se em consideração os pontos expostos acima, do ponto de vista computacional/matemático, o projeto propiciou mais possibilidades de aplicações de GANs na área de Áudio e Música. Para que o trabalho fosse bem sucedido, foi necessária uma ampla exploração das técnicas já disponíveis na literatura de MIR, assim como a adaptação de técnicas advindas de outras áreas e o desenvolvimento de novas heurísticas baseadas em conhecimento prévio sobre estrutura musical.

Já do ponto de vista musical e artístico, buscou-se a criação de novas ferramentas para o uso da tecnologia no processo criativo, especificamente na forma de Composição Musical Auxiliada por Computador (ASSAYAG et al., 1999) ou Composição Automática. Portanto, foram explorados processos automáticos de ressignificação da estética musical, criando assim a possibilidade de musificação condicionada por valores que quantificam as classificações emocionais realizadas por seres humanos. Os frutos do trabalho oferecem várias possíveis direções de pesquisa futuras, como musicoterapia e composição automática conduzida por narrativa.

Ao longo do trabalho, serão apresentados três modelos baseados na estrutura de Redes Generativas Adversárias. Assim, o restante do trabalho será organizado da seguinte forma: No Capítulo 2, será apresentado um sistema baseado no Mecanismo de Auto-Atenção voltado para a geração de pequenos trechos sonoros condicionados por instrumento musical. No Capítulo 3, será estudado um modelo VQGAN, composto pela combinação de um *Vector Quantized Variational Autoencoder* com uma GAN. O papel do modelo é codificar informação sonora para um espaço latente de forma a reduzir os custos relacionados à modelagem generativa. O último modelo a ser discutido, proposto no Capítulo 4, é um híbrido de Transformer com GAN, que gera informação musical no

domínio simbólico. A rede produz passagens condicionadas por categorias que representam regiões de estados afetivos.

# 1 Revisão da Literatura

## 1.1 Redes Generativas Adversárias

Redes Generativas Adversárias, ou simplesmente GANs (*Generative Adversarial Networks*, em inglês), constituem uma estrutura teórica para o treinamento de sistemas de aprendizado de máquinas na qual duas redes neurais competem entre si, culminando no refinamento das capacidades de cada uma. Uma das redes, conhecida como Generativa, tenta mapear vetores aleatoriamente selecionados pertencentes a uma distribuição *a priori* (geralmente $\mathcal{N}(0,1)$) a membros de uma determinada distribuição de dados reais que se deseja modelar, como imagens de faces humanas ou animais de uma determinada espécie. Já a outra, chamada de Discriminatória, fica encarregada de discernir entre os exemplos reais e aqueles produzidos pela rede Generativa, maximizando a atribuição de rótulos corretos. A distribuição da rede Geradora, $p_g$ sobre os dados $x$, é descoberta a partir de uma distribuição a priori $p(z)$, definida sobre variáveis de *noise*. Então, uma função diferenciável $G$, representada por uma rede neural parametrizada por $\theta_g$, é definida de forma a criar-se um mapeamento $G(\boldsymbol{z};\theta_g)$ para o espaço de dados. Uma outra rede neural, $D(\boldsymbol{x};\theta_d)$, também é definida. Esta rede gera um escalar $D(\boldsymbol{x})$, este que representa a probabilidade $\boldsymbol{x}$ ter vindo dos dados reais e não da distribuição sintética. Então, $D$ é treinada para maximizar a probabilidade de atribuição correta de rótulos tanto para os dados reais quanto para aqueles produzidos por $G$. Ao mesmo tempo, $G$ enganar $D$, fazendo-a classificar dados falsos como reais.

Matematicamente, essa interação é expressa por meio de um jogo min-max:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))], \tag{1.1}$$

onde $G$ é a rede generativa e $D$ a Discriminatória, $p_{data}(\boldsymbol{x})$ é a distribuição de dados reais, e $p_z(\boldsymbol{z})$ a distribuição de dados *a priori*. O Algoritmo 1 apresenta uma iteração de

treinamento de uma GAN.

**Algoritmo 1** – Algoritmo original da GAN. A rede Discriminatória $D$ tem parâmetros $w$, enquanto a Geradora $G$ tem parâmetros $\theta$.

**Input:** $\alpha_g$, taxa de aprendizado da rede Geradora, $\alpha_d$, taxa de aprendizado da rede Discriminatória. $m$, o tamanho de *minibatch*. $n_{disc}$, o número de iterações de $D$ por iteração de $G$. Opt é o otimizador.

**enquanto** $\theta$ não convergir **faça**

  **para** $j$ em $0, ..., n_{disc}$ **faça**

    Amostre $\{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$ do conjunto de dados real.

    Amostre $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m}$ de $p_g(\boldsymbol{z})$

    $g_w \leftarrow \nabla_w \left[ -\frac{1}{m} \sum_{i=1}^{m} \log D(\boldsymbol{x}^{(i)}) - \log(1 - D(G(\boldsymbol{z}^{(i)}))) \right]$

    $w \leftarrow w - \alpha_d \cdot \mathrm{Opt}_D(w, g_w)$

  **fim**

  Amostre $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m}$ de $\mathcal{N}(0, I)$.

  $g_\theta \leftarrow \nabla_\theta \left[ \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(\boldsymbol{z}^{(i)}))) \right]$

  $\theta \leftarrow \theta - \alpha_g \cdot \mathrm{Opt}_G(\theta, g_\theta)$

**fim**

É possível provar (GOODFELLOW et al., 2014) que para um $G$ fixo, a rede Discriminatória $D$ ótima é dada por:

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}. \tag{1.2}$$

Em termos de G, pode-se então redefinir a Equação 1.1 como

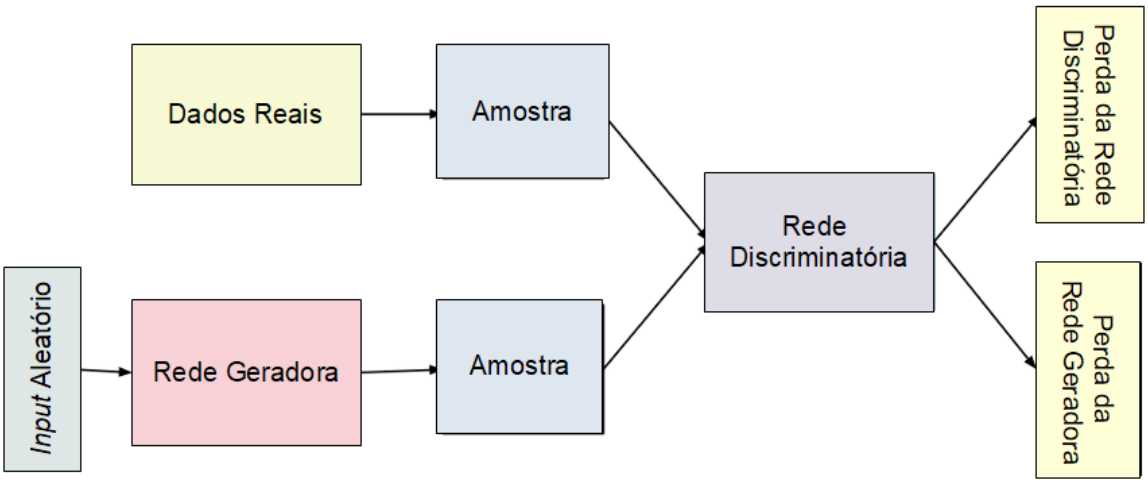
$$\begin{aligned} C(G) &= \max_D \ V(D, G) && (1.3)\\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{data}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_z}[\log(1 - D_G^*(G(\boldsymbol{z})))] && (1.4)\\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{data}}[\log D_G^*(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x} \sim p_g}[\log(1 - D_G^*(\boldsymbol{x}))] && (1.5)\\ &= \mathbb{E}_{\boldsymbol{x} \sim p_{data}} \left[ \log \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \right] + \mathbb{E}_{\boldsymbol{x} \sim p_g} \left[ \log \frac{p_g(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})} \right]. && (1.6) \end{aligned}$$

A expressão acima define um critério de treinamento para $G$, dado um $D$ ótimo. O mínimo global deste critério é atingido se, e somente se, $p_g = p_{\text{data}}$, ou seja, quando as distribuições sintética e real são iguais. Além disso, quando $D$ e $G$ possuem capacidade o suficiente[1] (no limite não paramétrico) se $D$ pode atingir a otimalidade em cada passo do Algoritmo 1 dado um $G$ fixo, e quando $p_g$ é atualizado de forma a melhorar o critério

$$D_G^*(\boldsymbol{x}) = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}, \tag{1.7}$$

[1] A capacidade de uma rede neural relaciona-se ao nível de complexidade dos padrões que a rede é capaz de expressar.

Figura 1 – Redes Generativas Adversárias.




Fonte: Adaptada de Google Developers [2]

então $p_g$ converge a $p_{\text{data}}$. Apesar do sucesso das GANs e das garantias teóricas dadas acima, a definição de $G$ restringe a família de distribuições $p_g$ às quais se tem acesso, e quando uma rede neural é utilizada para definir $G$, vários pontos críticos são introduzidos no espaço de parâmetros (GOODFELLOW et al., 2014). Um diagrama detalhando o funcionamento das GANs é dado na Figura 1.

Também é possível guiar a geração da rede através de informações condicionais (MIRZA; OSINDERO, 2014). Essa técnica, além de facilitar o treinamento das redes, possibilita, no momento da inferência, que o usuário do modelo escolha que tipo de dado deseja receber. No caso de uma rede geradora de dígitos numéricos, por exemplo, pode-se fornecer a informação adicional de qual dígito deve ser gerado. Nesse caso, a Equação 1.1 transforma-se em:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))], \quad (1.8)$$

em que $y$ é o rótulo fornecido.

O treinamento de uma GAN é inerentemente instável, pois, ao contrário do que ocorre na maior parte dos algoritmos de *Machine Learning*, a função *loss* global de uma GAN, representada pela estimativa do erro calculada por $D$, não é fixa. Muitas vezes, o algoritmo não converge; em outras, a rede Geradora, em uma tentativa de maximizar sua capacidade de "enganar" a Discriminatória, acaba por produzir uma quantidade limitada de amostras realistas, sacrificando a variedade no processo - um fenômeno conhecido como Colapso de Modos. Dessa forma, uma das direções de pesquisa mais frutíferas

[2] Disponível em: <https://developers.google.com/machine-learning/gan/gan_structure>. Acesso em: 18 abr. 2020

dentro do ramo das GANs é o desenvolvimento de estratégias de objetivo, regularização e normalização que estabilizem o treino das redes e melhorem sua convergência.

A Wasserstein GAN, ou simplesmente WGAN (ARJOVSKY; CHINTALA; BOTTOU, 2017), é uma GAN treinada com um objetivo modificado, a distância de Wasserstein, cujo propósito é servir como medida da distância entre a distribuição original e aquela produzida por $G$. Além de evitar o problema de saturação das GANs, essa distância expressa de maneira mais clara o que se deseja da rede Geradora.

Seja $\mathcal{X}$ um espaço métrico compacto. Uma $\sigma$-álgebra em $\mathcal{X}$ é uma coleção $\mathcal{F}$ de subconjuntos de $\mathcal{X}$ satisfazendo as seguintes condições:

(a) $\mathcal{X} \in \mathcal{F}$.

(b) se $B \in \mathcal{F}$, então, sendo $B^c$ seu complemento, também vale que $B^c \in \mathcal{F}$.

(c) Se $A_n$ é uma sequência de elementos em $\mathcal{F}$, então a união $\bigcup_{n=1}^{\infty} A_n$ é tal que $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$.

Dada uma coleção $\mathcal{C}$ de subconjuntos de $\mathcal{X}$, $\mathcal{C} \in P(\mathcal{X})$, a menor $\sigma$-álgebra contendo $\mathcal{C}$, correspondente à intersecção entre todas as $\sigma$-álgebras contendo $\mathcal{C}$, é conhecida como a $\sigma$-álgebra gerada por $\mathcal{C}$. Uma $\sigma$-álgebra de Borel, finalmente, é a $\sigma$-álgebra gerada pelos conjuntos abertos de $\mathcal{X}$, e um conjunto de Borel é um elemento de uma $\sigma$-álgebra de Borel.

Assim, sendo $\Sigma$ o conjunto de todas os subconjuntos de Borel de $\mathcal{X}$, e $\text{Prob}(\mathcal{X})$ o espaço de medidas de probabilidade definidas em $\mathcal{X}$, temos que a Distância de Wasserstein-1 entre duas distribuições $\mathbb{P}_r, \mathbb{P}_g \in \text{Prob}(\mathcal{X})$ é dada por

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|], \tag{1.9}$$

onde $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ é o conjunto de todas as distribuições conjuntas $\gamma(x, y)$ cujas marginais são, respectivamente, $\mathbb{P}_r$ e $\mathbb{P}_g$. Essa medida indica, intuitivamente, a quantidade de massa que deve ser transportada de $x$ a $y$ para que a distribuição $\mathbb{P}_r$ seja transformada na distribuição $\mathbb{P}_g$. A distâcia de Wasserstein-1, portanto, é o custo do plano de transporte ótimo. A dualidade de Kantorovich-Rubinstein (VILLANI, 2009) diz que:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leqslant 1} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r}[f(\boldsymbol{x})] - \mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_\theta}[f(\hat{\boldsymbol{x}})], \tag{1.10}$$

onde o supremo é tomado sobre todas as funções 1-Lipshitz contínuas $f : \mathcal{X} \to \mathbb{R}$. Uma função $f : \mathcal{X} \to \mathbb{R}$ é chamada de $K$-Lipschitz contínua se existir constante $K > 0$ tal que :

$$|f(x) - f(y)| \leqslant K|x - y|, \quad \forall x, y \in \mathcal{X}. \tag{1.11}$$

Ainda em (ARJOVSKY; CHINTALA; BOTTOU, 2017), os autores demonstram que, sendo $\mathbb{P}_r$ uma distribuição qualquer e $\mathbb{P}_\theta$ a distribuição definida implicitamente pela rede Geradora, $g_\theta(z) = \hat{\boldsymbol{x}}$, $\boldsymbol{z} \sim p(\boldsymbol{z})$ a partir de variáveis aleatórias com densidade p, então existe uma solução $f : \mathbb{X} \to \mathbb{R}$ para o problema

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \max_{\|f\|_L \leqslant 1} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r}[f(\boldsymbol{x})] - \mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_\theta}[f(\hat{\boldsymbol{x}})] \tag{1.12}$$

A partir disso, o jogo minmax que define a interação entre as GANs torna-se

$$\min_G \max_{\|D\|_L \leqslant 1} \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_r}[f(\boldsymbol{x})] - \mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_\theta}[f(\hat{\boldsymbol{x}})]. \tag{1.13}$$

Ou seja, $D$ tenta afastar as distribuições, enquanto $G$ tenta aproximá-las.

Cabe notar que aqui, a rede Discriminatória é substituída por uma rede chamada de Crítico que, em vez de servir como adversária de $G$, atua como um guia que a auxilia a melhorar a qualidade das amostras geradas. Da análise acima, segue que as equações que definem as *losses* de ambas as redes estão dadas nas Equações 1.14 e 1.15, respectivamente:

$$\mathcal{L}_{WGAN,D} = -\left[\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[D(G(\boldsymbol{z}))]\right] \tag{1.14}$$

$$\mathcal{L}_{WGAN,G} = -\mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[D(G(\boldsymbol{z}))]. \tag{1.15}$$

Um dos pré-requisitos para o funcionamento da WGAN é o de que o Crítico seja uma função Lipschitz contínua. Originalmente, a solução proposta para garantir essa propriedade consistia em estabelecer um limitante superior para os parâmetros da rede e diminuir para o número escolhido quaisquer valores que ultrapassem esse teto. Essa estratégia impõe uma limitação muito forte, e por vezes impossibilita a convergência ou a geração de amostras com boa qualidade. Como solução para os problemas mencionados, em (GULRAJANI et al., 2017), propõe-se uma penalidade de gradiente incorporada como um fator adicional na função *loss* do modelo:

$$\mathcal{L}_{GP} = \mathbb{E}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\| - 1)^2], \tag{1.16}$$

onde $\mathbb{P}_{\hat{\boldsymbol{x}}}$ é definida implicitamente por meio da amostragem uniforme ao longo de linhas retas entre pares de pontos amostrados das distribuições real, $\mathbb{P}_r$, e a do gerador, $\mathbb{P}_\theta$. Essa amostragem se dá pois, conforme demonstrado em (GULRAJANI et al., 2017), o crítico ótimo possui norma de gradiente 1 quase em todo lugar sob $\mathbb{P}_r$ e $\mathbb{P}_\theta$. Além disso, os autores do artigo, por meio de experimentos, mostram que reforçar a restrição de gradiente com norma 1 apenas ao longo dessas linhas retas, em vez de em todo o espaço é usualmente suficiente para garantir bons resultados.

Uma outra função *loss* sugerida para sanar os problemas relacionados ao algoritmo original da GAN é a *Relativistic Standard loss* (RSGAN) (JOLICOEUR-MARTINEAU, 2019), que força a rede Discriminatória a estimar a probabilidade de os dados originais serem mais realistas do que os sintéticos, em vez de estimar a probabilidade de os dados recebidos serem reais. Os objetivos para essa versão da GAN correspondentes às redes Discriminatória e Geradora, respectivamente, estão dados abaixo nas Equações 1.17 e 1.18.

$$\mathcal{L}_{RSGAN,D} = -\mathbb{E}_{(\boldsymbol{x}_r,\boldsymbol{x}_f)\sim(\mathbb{P}_r,\mathbb{P}_\theta)}[\log(\text{sigmoid}(D(\boldsymbol{x}_r) - D(\boldsymbol{x}_f)))] \tag{1.17}$$

$$\mathcal{L}_{RSGAN,G} = -\mathbb{E}_{(\boldsymbol{x}_r,\boldsymbol{x}_f)\sim(\mathbb{P}_r,\mathbb{P}_\theta)}[\log(\text{sigmoid}(D(\boldsymbol{x}_f) - D(\boldsymbol{x}_r)))], \tag{1.18}$$

onde $\mathbb{P}_r$ e $\mathbb{P}_\theta$ são as distribuições real e da rede Geradora, respectivamente. A função sigmoid, por sua vez, é dada por $\phi(z) = \dfrac{1}{1+\exp(-z)}$.

A Hinge Loss (LIM; YE, 2017) encoraja a rede Discriminatória a aprender um hiperplano que separa entre as amostras falsas e as verdadeiras e a maximizar a distância entre essas, enquanto a rede Geradora tenta aproximá-las. As funções *loss* para $D$ e $G$, em respectivo, são:

$$\mathcal{L}_{Hinge,D} = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[\max(0, 1 - D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\max(0, 1 + D(G(\boldsymbol{z})))\right] \tag{1.19}$$

$$\mathcal{L}_{Hinge,G} = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[-D(G(\boldsymbol{z}))]. \tag{1.20}$$

A *Hinge loss* canonicamente é utilizada no treinamento de *Support Vector Machines* (SVMs) (GOODFELLOW; BENGIO; COURVILLE, 2016), e para esta, penaliza classificações nas quais o dado considerado é posto no lado errado da fronteira de decisão, ou muito próximo desta, e não penaliza classificações nas quais o modelo insere o dado no lado correto da fronteira de decisão e distante dessa. No caso da *Hinge Loss* para GANs, observa-se que a rede Discriminatória é encorajada a aproximar $D(\boldsymbol{x})$ de 1, não obtendo penalização também quando esses valores estão ao lado direito de (ou sejam maiores que) 1, e ao mesmo tempo aproxima $D(G(\boldsymbol{z}))$ de $-1$, não obtendo penalização caso os valores estejam ao lado esquerdo de (ou sejam menores que) $-1$. Já a rede Geradora deve aproximar $D(G(\boldsymbol{z}))$ da região positiva da reta, ou seja, aproximando esses valores dos dados reais.

A normalização espectral (MIYATO et al., 2018) é uma técnica que introduz uma condição de regularidade na derivada de $D$ por meio do controle da constante de Lipschitz da rede através da limitação da norma espectral de cada camada $g : \mathbf{h}_{in} \mapsto \mathbf{h}_{out}$ da mesma. A norma de Lipschitz $||g||_{Lip}$ é igual a $\sup_h \sigma(\nabla g(\mathbf{h}))$, onde $\sigma(A)$ é a norma espectral da matriz $A$. Esta, por sua vez, é o maior valor singular de A, ou seja, a raiz

quadrada do maior auto-valor da matriz $A^*A$, onde $A^*$ denota o conjugado transposto de A. Este valor também é equivalente à norma-2 de A, ou seja:

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^*A)} = \sigma_{\max}(A) = \max_{|x|_2 \neq 0} \frac{|Ax|_2}{|x|_2}. \tag{1.21}$$

Assim, limitando-se esse valor, restringe-se também o gradiente de $g$, o que evita o fenômeno de explosão do gradiente e auxilia na estabilidade do treino. Além de trazer um considerável ganho tanto em se tratando de performance quanto de estabilidade, o método é computacionalmente leve e de fácil implementação, e portanto foi utilizado em muitos dos modelos de GAN do estado-da-arte (BROCK; DONAHUE; SIMONYAN, 2018; KARRAS et al., 2020).

### 1.1.1 Métricas

A tarefa de geração artificial é uma das mais complexas no ramo do *Machine Learning*. Para que seja realizada de maneira satisfatória, é necessário que haja modelos capazes não apenas de entender uma distribuição de dados, mas também de replicá-la e gerar membros não observados de tal distribuição. Dada essa complexidade, a análise quantitativa da performance de modelos generativos ainda é um desafio, e a determinação da qualidade dos modelos é sempre dependente da avaliação por parte de especialistas ou de membros do público geral. No entanto, existe uma área de pesquisa especialmente voltada para o desenvolvimento de métricas que auxiliem os pesquisadores a avaliar e comparar o desempenho de modelos generativos. No caso das GANs, há duas métricas que se destacam e que têm sido frequentemente utilizadas, o *Inception Score* (IS) (SALIMANS et al., 2016) e a *Frechét Inception Distance* (FID) (HEUSEL et al., 2017).

O *Inception Score* (IS) é uma métrica que se propõe a avaliar tanto o realismo quanto a variedade do conteúdo produzido pela rede generativa. Aplicando-se o modelo *Inception-V3* (SZEGEDY et al., 2015), uma rede convolucional treinada para classificação de imagens, às amostras geradas, calcula-se a distribuição condicional por rótulo $p(\boldsymbol{y}|\boldsymbol{x})$. De forma mais específica, as duas propriedades que o *Inception Score* se propõe a medir são as seguintes (BARRATT; SHARMA, 2018):

- Entropia da distribuição condicional: Amostras devem conter objetos bem-definidos, e portanto devem ter uma distribuição condicional $p(y|\boldsymbol{x})$ com entropia baixa.
- Entropia da distribuição marginal: Para que a rede gere amostras variadas, a distribuição marginal $\int p(\boldsymbol{y}|\boldsymbol{x} = G(\boldsymbol{z}))d\boldsymbol{z}$ deve ter entropia alta.

A métrica que combina essas duas características é a exponencial da divergência de Kullback-leibler, ou simplesmente divergência KL, entre as distribuições $p(y|\boldsymbol{x})$ e $p(y)$, esta última que é:

$$D_{KL}(p(y|\boldsymbol{x})\|p(y)) = \sum_i p(y_i|\boldsymbol{x}_i)\log\frac{p(y_i|\boldsymbol{x}_i)}{p(y_i)}, \tag{1.22}$$

onde cada $y_i$ ou $x_i$ é um ponto extraído das distribuições. Esta divergência serve como um análogo probabilístico à distância entre as distribuições, ainda que não seja, matematicamente falando, uma distância. Quando o modelo generativo apresenta as duas características desejadas, a divergência entre as distribuições é alta, e o valor da IS é grande (BARRATT; SHARMA, 2018).

A FID, por sua vez, compara a estatística das amostras reais com a das amostras geradas.

Um conjunto mensurável é um conjunto pertencente a um espaço mensurável. Um conjunto $X$ é dito espaço mensurável se é dotado de uma $\sigma$-álgebra $M$ (vide acima para uma definição de $\sigma$-àlgebra).

Dadas duas distribuições $p(.)$ e $p_w(.)$, pode-se afirmar que $p(.) = p_w(.)$ em um conjunto mensurável se, e somente se, $\int p(.)f(x)dx = \int p_w(.)f(x)dx$ para uma base $f(.)$ definida em todo o espaço onde $p(.)$ e $p_w(.)$ estão definidos ().

Essas igualdades são utilizadas na descrição de distribuições por momentos, onde os $f(x)$ são polinômios (HEUSEL et al., 2017). Momentos de uma distribuição de probabilidades são números que nos permitem descrever o formato daquela distribuição, descartando certas informações em troca de simplicidade. Em particular, o primeiro e segundo momentos de uma determinada distribuição de probabilidades são, respectivamente, a média (ou valor esperado) e a variância daquela distribuição. Formalmente, dada uma distribuição contínua com densidade probabilística $p(.)$, o $n$-ésimo momento associado à distribuição é $\langle x^n\rangle = \int_{-\infty}^{\infty} p(x)x^n dx$.

Para o cálculo do FID, utilizam-se, no lugar da variável $x$, as camadas do modelo *Inception-V3*, e no lugar de $f(.)$ os primeiros dois momentos que descrevem a distribuição (média e variância). Levando-se em consideração o fato de que a Gaussiana é a distribuição de máxima entropia dada uma certa média e variância, e portanto assumindo-se que as camadas do modelo seguem uma distribuição Gaussiana multidimensional, mede-se a distância entre a Gaussiana real e a sintética através da distância de Frechét, dada na equação 1.23.

$$d^2((\mathbf{m},\mathbf{C}),(\mathbf{m}_w,\mathbf{C}_w)) = \|\mathbf{m}-\mathbf{m}_w\|_2^2 + \mathrm{Tr}(\mathbf{C}+\mathbf{C}_w - 2(\mathbf{C}\mathbf{C}_w)^{\frac{1}{2}}), \tag{1.23}$$

em que $(\mathbf{m}, \mathbf{C})$ e $(\mathbf{m}_w, \mathbf{C}_w)$ representam a média e covariância da distribuição real e sintética, respetivamente.

Ainda que essas métricas sejam calculadas a partir de *features* retiradas de uma rede classificadora de imagens, as mesmas podem ser adaptadas a outros domínios, desde que existam modelos suficientemente poderosos treinados em tais domínios. Dada essa condição, podem-se extrair as *features* desses modelos e usá-las no lugar daquelas advindas do modelo *Inception-V3*. As *features* originadas de modelos de *deep learning* são ferramentas de grande utilidade, pois em uma rede bem treinada elas tendem a representar características importantes do *dataset* para o qual são otimizadas (ZHANG et al., 2018) e de *datasets* com características semelhantes.

### 1.1.2 GANs para sequências discretas

GANs podem ser adaptadas para lidar com diversos tipos de dados. No entanto, até então, elas obtiveram um sucesso muito maior com dados contínuos, ou seja, dados compostos por pontos que podem teoricamente ocupar qualquer valor em um intervalo contínuo (ainda que, na prática, estes valores sejam quantizados, graças à estrutura dos bits de computador), como imagens e vídeos, do que com dados discretos, como texto ou outros tipos de informação sequencial. Isso se dá pois para que uma GAN seja treinada, é necessário que gradientes se propaguem da Rede Discriminatória para a Rede Generativa, ou seja, todas as operações envolvidas no processo devem ser diferenciáveis. No caso de uma distribuição discreta ou categórica, no entanto, as amostras são produzidas por meio de uma operação $\arg\max$ aplicada na distribuição de probabilidades gerada após a última camada de rede, e essa operação não é diferenciável. Para contornar essa dificuldade, algumas técnicas já foram desenvolvidas. Uma delas consiste em formular o problema como uma tarefa de Aprendizagem por Reforço (YU et al., 2017), usando o algoritmo REINFORCE (WILLIAMS, 1992) durante o treino. Uma outra consiste em aproximar a distribuição categórica através da distribuição Gumbel-Softmax (JANG; GU; POOLE, 2017; KUSNER; HERNáNDEZ-LOBATO, 2016). Especificamente, sendo $z$ uma variável categórica com probabilidades $\pi_1, ..., \pi_k$, podemos tomar amostras dessa distribuição através de:

$$y = \text{one\_hot}\left(\arg\max_i [g_i + \log \pi_i]\right), \tag{1.24}$$

em que $g_i$ são amostrados de uma distribuição Gumbel$(0, 1)$ (local 0 e escala 1), a função argmax seleciona o argumento de valor máximo, e a função one_hot transforma uma linha matricial contendo números inteiros em uma matriz onde cada posição equivale a uma coluna com 1 na linha correspondente ao número inteiro (partindo de zero), como no

exemplo abaixo:

$$\text{one\_hot}([0,1,3,2]) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{1.25}$$

A expressão 1.24 não é diferenciável graças à função $\arg\max$. No entanto, pode-se obter uma aproximação continuamente diferenciável dessa operação em termos da função softmax $\left(\text{softmax}(x)_i = \dfrac{\exp(x_i)}{\sum_{j=1}^{K}\exp(x_i)}\right)$:

$$y = \text{softmax}((1/\tau)(\pi + g)), \tag{1.26}$$

em que $\tau$ é um parâmetro de temperatura inversa. Para valores pequenos de $\tau$, a distribuição gerada se assemelha à categórica gerada em 1.24, porém a variância dos gradientes é baixa. Já para valores grandes, aproxima-se da distribuição uniforme, e a variância é alta. Em geral, o parâmetro progride de valores altos para próximos de zero durante o treinamento, pois espera-se que a confiança do modelo nas amostras produzidas aumente ao longo do processo.

## 1.2 Modelos sequenciais

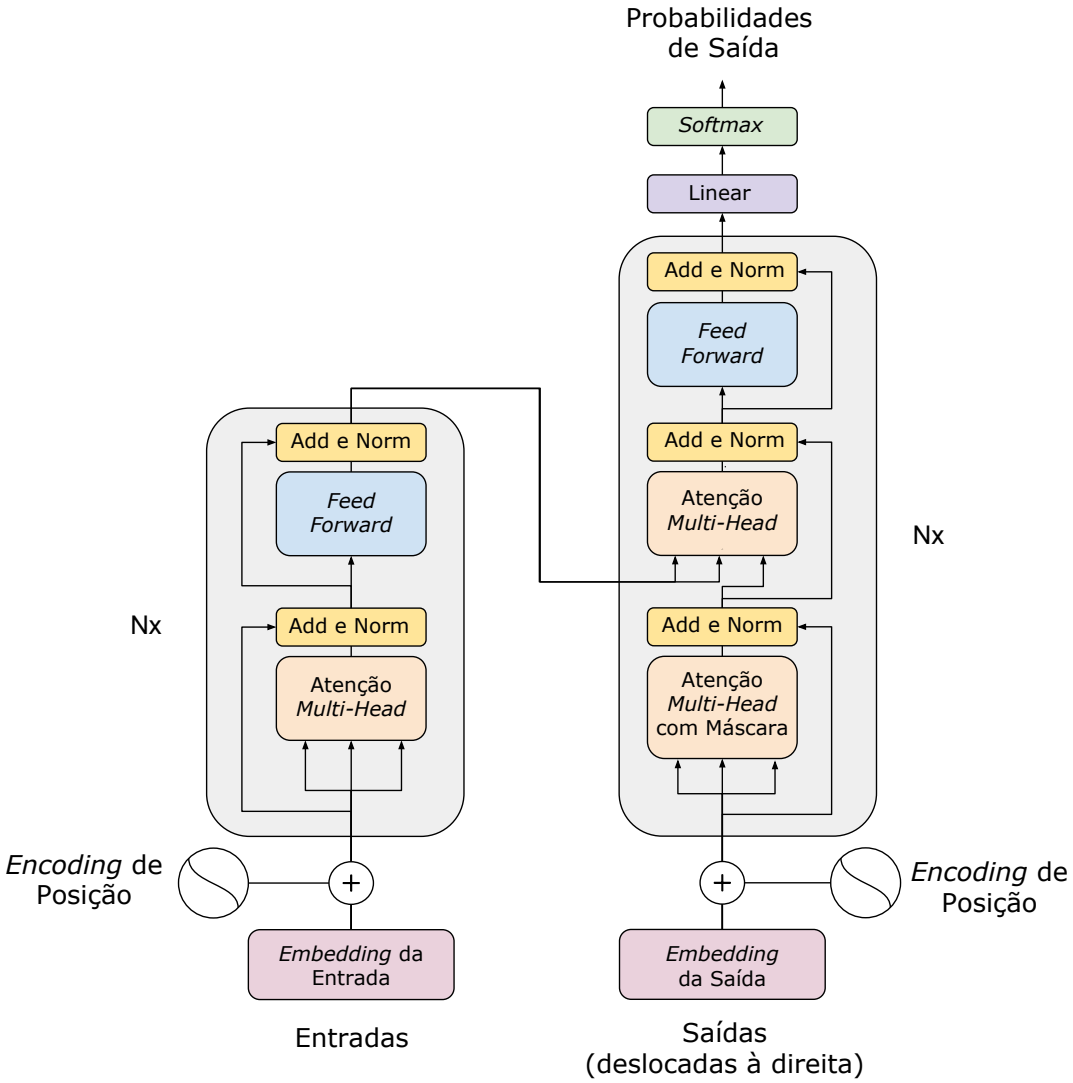
Modelos sequenciais são caracterizados por receber ou gerar dados em forma de sequência, como texto ou clipes de áudio, sendo, assim, capazes de trabalhar com dados que evoluem no tempo, ou seja, séries temporais.

Redes Neurais Recorrentes (RNNs) dividem o processamento de uma sequência em várias iterações, isto é, várias execuções do algoritmo, onde em cada uma de tais iterações a rede recebe a informação gerada no passo anterior, retendo informação sobre essas diversas passagens. Redes *Long Short Term Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) são um tipo consideravelmente popular de RNNs capazes de aprender dependências de longo prazo através de *gates* que controlam o fluxo de informação pela célula e de estados que carregam informações sobre iterações passadas.

Pouco tempo após sua apresentação, o modelo Transformer (VASWANI et al., 2017) atingiu grande relevância no campo do Processamento Natural de Linguagem (NLP). Esse sucesso se deve, em grande parte, ao mecanismo de Atenção[3], que permite a modelagem da relação entre todos os elementos de uma sequência ou entre os elementos

[3] Existem diversas versões do mecanismo de Atenção. As duas mais comuns são a aditiva e a multiplicativa. No contexto deste trabalho, usaremos o termo "mecanismo de Atenção" para nos referir à versão multiplicativa ou "de produto interno", que é a apresentada.

Figura 2 – Modelo Transformer.




Fonte: (NEVES, 2022)

de várias sequências (no caso em que apenas uma sequência é utilizada, o mecanismo também pode ser chamado de "Auto-Atenção"). O mecanismo recebe três matrizes, $Q$, $K$, e $V$, correspondendo, respectivamente a *queries*, *keys*, e *values* construídos através de transformações lineares aplicadas à sequência que recebem. O resultado da operação é uma matriz em que cada linha é uma combinação linear dos *values* e os coeficientes da combinação são calculados por uma função de similaridade entre as *keys* e as *queries*. Formalmente:

$$A(Q, K, V)_i = \frac{\sum_{j=1}^{N} \text{sim}(Q_i, K_j) V_j}{\sum_{j=1}^{N} \text{sim}(Q_i, K_j)}. \tag{1.27}$$

Originalmente, a função de similaridade utilizada no Transformer é a exponencial do produto interno entre uma *query* e uma *key* escalada por um fator dimensional $1/\sqrt{(D)}$, onde $D$ é a dimensionalidade de cada vetor de *features* nas matrizes $Q, K, V$. Essa expressão também é equivalente à função softmax entre as *queries* e *keys*:

$$A(Q,K,V)_i = \frac{\sum_{j=1}^{N} \exp(\frac{Q_i^T K_j}{\sqrt{D}}) V_j}{\sum_{j=1}^{N} \exp(\frac{Q_i^T K_j}{\sqrt{D}})} = \sum_{j=1}^{N} \text{softmax}\left(\frac{Q_i^T K_j}{\sqrt{D}}\right) V_j. \tag{1.28}$$

No entanto, o cálculo dessa função demanda recursos computacionais que crescem quadraticamente com o tamanho das sequências recebidas pelo modelo. Em decorrência desse fato, diversos esforços de pesquisa têm sido direcionados para a criação de versões do mecanismo de Atenção que necessitem de uma quantidade menor de computação e memória (CHILD et al., 2019; KATHAROPOULOS et al., 2020; KITAEV; KAISER; LEVSKAYA, 2020; CHOROMANSKI et al., 2021). As soluções encontradas incluem a restrição do cálculo da matriz a uma região local (LIU* et al., 2018), o uso de padrões esparsos na matriz de Atenção (CHILD et al., 2019), o desenvolvimento de *kernels* especiais que permitam uma fatorização do cálculo da matriz de Atenção de modo a reduzir o número de operações envolvidas (CHOROMANSKI et al., 2021; KATHAROPOULOS et al., 2020), entre outras. Especificamente, como proposto em (KATHAROPOULOS et al., 2020), assumindo que o *kernel* possua uma representação $\phi(x)$, a equação 1.27 torna-se:

$$A(Q,K,V)_i = \frac{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j) V_j}{\sum_{j=1}^{N} \phi(Q_i)^T \phi(K_j)} = \frac{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j) V_j}{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j)}. \tag{1.29}$$

Perceba que, desta forma, pode-se calcular o fator dentro da soma apenas uma vez e reutilizar os resultados para o cálculo de todas as *queries*.

Também nesse trabalho, uma analogia é feita entre os modelos *Transformers* e as RNNs. Partimos da versão causal linearizada do mecanismo de atenção, na qual o cálculo de uma determinada posição $i$ é influenciado por uma posição $j$ apenas se $i \leqslant j$ (o cálculo de um elemento posição não pode ser influenciado pelos elementos anteriores):

$$A(Q,K,V)_i = \frac{\phi(Q_i)^T \sum_{j=1}^{i} \phi(K_j) V_j}{\phi(Q_i)^T \sum_{j=1}^{i} \phi(K_j)}. \tag{1.30}$$

Escrevendo $S_i = \sum_{j=1}^{i} \phi(K_j) V_j$ e $Z_i = \sum_{j=1}^{i} \phi(K_j)$, temos:

$$A(Q,K,V)_i = \frac{\phi(Q_i)^T S_i}{\phi(Q_i)^T Z_i}. \tag{1.31}$$

Pode-se, então, considerar $Z_i$ e $S_i$ como análogos aos estados presentes em RNNs, que são atualizados a cada iteração da rede e carregam informações sobre as passagens anteriores. Assim, obtém-se:

$$\begin{aligned} s_0 &= 0 \\ z_0 &= 0 \\ s_i &= s_{i-1} + \phi(x_i W_K)(x_i W_V)^T \\ z_i &= z_{i-1} + \phi(x_i W_K) \\ y_i &= f_l\left(\frac{\phi(x_i W_Q)^T s_i}{\phi(x_i W_Q)^T z_i} + x_i\right), \end{aligned} \tag{1.32}$$

onde $x_i$ é a $i$-ésima entrada e $y_i$ a i-ésima sáida após a aplicação da camada de *feedforward* (KATHAROPOULOS et al., 2020).

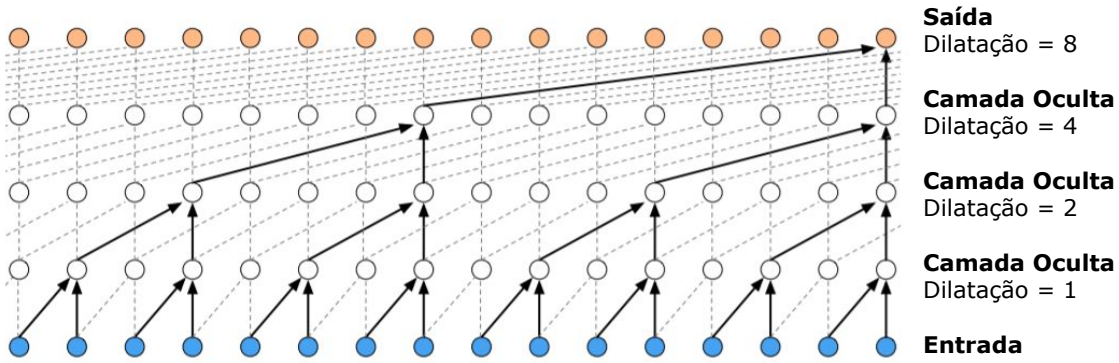
O Wavenet (OORD et al., 2016) é um dos primeiros e mais populares modelos criados especificamente para lidar com áudio. Ele trabalha de forma autorregressiva, ou seja, considerando o que já foi gerado para criar a próxima amostra de áudio. O modelo é uma rede neural totalmente convolucional com camadas que possuem vários fatores de dilatação, permitindo que seu campo receptivo cresça exponencialmente com a profundidade. Como pode ser visto na imagem 3, isso permite que cada amostra gerada tenha acesso a informações sobre outras temporalmente distantes, e o esquema de dilatação faz com que nem todos os pontos sejam considerados nos cálculos, reduzindo os custos computacionais.

Outro modelo que gera áudio amostra por amostra é o SampleRNN (MEHRI et al., 2016). Ele consiste de uma hierarquia de RNNs em que cada uma de tais redes captura uma escala diferente de áudio, e as informações advindas dos níveis que trabalham com as estruturas em larga escala condicionam a geração dos submodelos que atuam em escalas menores.

Embora o Wavenet e os trabalhos que inspirou tenham sido aplicados com sucesso à tarefa de geração musical, o custo computacional envolvido na modelagem autorregressiva ainda permanece um desafio, apesar da existência de técnicas que atenuam esse problema, como a descrita em (OORD et al., 2017), que utiliza destilação de densidade paralela, fluxos autorregressivos inversos (IAFs) e um esquema de treinamento professor-aluno para aumentar a velocidade de amostragem do Wavenet em várias ordens de magnitude.

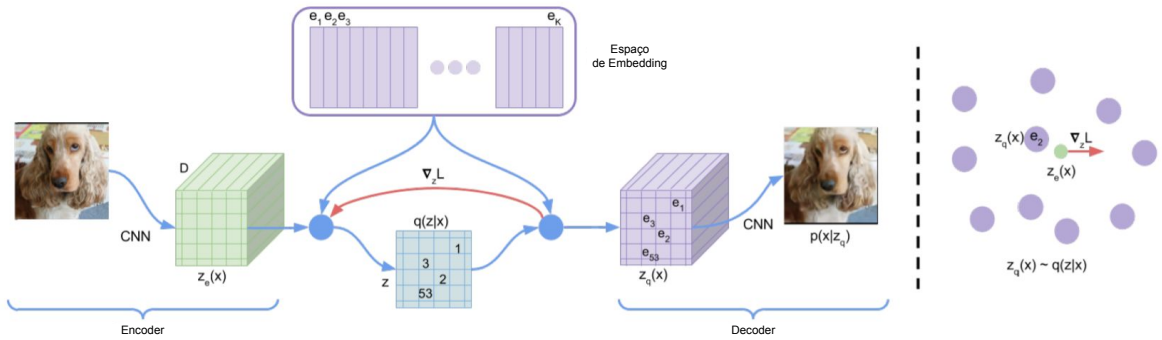
O *Variational Autoencoder*, ou VAE (KINGMA; WELLING, 2014), é um sistema generativo consistindo em um *Encoder* $q_\phi(\mathbf{z}, \mathbf{x})$ que parametriza uma distribuição *a posteriori*, mapeando pontos originados de uma distribuição $q_D(\mathbf{x})$, comumente difícil de modelar, para uma distribuição $q(\mathbf{z})$ cuja estrutura é mais simples. A parte generativa

Figura 3 – Convoluções dilatadas em um modelo Wavenet.




Fonte: Adaptado de (OORD et al., 2016)

Figura 4 – Modelo VQ-VAE.




Fonte: Adaptado de (OORD; VINYALS; KAVUKCUOGLU, 2017)

de rede aprende uma distribuição $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$, onde o primeiro fator é uma distribuição *a priori* sobre o espaço latente, e a segunda é o *Decoder* (KINGMA; WELLING, 2019). A otimização é feita através da maximização do Limite Inferior da Evidência (ELBO):

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]. \tag{1.33}$$

Essa quantia aproxima $\log p_\theta(\mathbf{x})$.

O *Vector Quantized Variational Autoencoder* (VQ-VAE) (OORD; VINYALS; KAVUKCUOGLU, 2017) é uma variação do VAE original que substitui as distribuições *a priori* e *a posteriori* por variáveis latentes discretas. Um Encoder $E$ recebe um *input* $x$ e o projeta para um conjunto de códigos $z_e(x)$. Para que seja feita a construção das variáveis latentes discretas $z_q$, um modelo chamado de *Vector Quantizer* (VQ) mapeia cada variável para seu vizinho mais próximo dentro de um conjunto de vetores $\mathcal{C} = \{e_k\}_{k=1}^K$, que fica conhecido como *codebook*, e que é aprendido durante o treinamento para condensar

informações relevantes sobre os dados recebidos. Então, um modelo autorregressivo é treinado sobre essas representações, gerando diretamente no espaço latente, ou seja, produzindo novas variáveis latentes que representam as informações compactadas pelo Encoder. O Decoder $D$ recebe essa representação discretizada e tenta reconstruir o material original a partir dela, perdendo a menor quantidade de informação possível no caminho. Formalmente,

$$z_q(x) = e_k, \quad \text{onde} \quad k = \arg\min_j \|z_e(x) - e_j\|_2. \tag{1.34}$$

A função *loss* do VQVAE total é:

$$\mathcal{L}_{VQVAE} = \log p(x|z_q(x)) + \|sg[z_e(x)] - z_q(x)\|_2^2 + \beta\|z_e(x) - sg[z_q(x)]\|_2^2. \tag{1.35}$$

Na equação, o primeiro termo é a perda de reconstrução. O segundo termo, que caracteriza o objetivo do Vector Quantizer, treina os vetores de *embedding* $z_q(x)$ do *codebook* para se aproximarem dos *outputs* do Decoder, $z_e(x)$, por meio da perda $l_2$. Note que $sg[.]$ indica a operação de *stop-gradient*, responsável por parar o fluxo de gradiente naquela variável. Por fim, o terceiro termo, conhecido como "perda de comprometimento", assegura que os *outputs* $z_e(x)$ do Decoder se comprometam aos elementos do *codebook*, não trocando arbitrariamente de um a outro elemento ao longo do treino (OORD; VINYALS; KAVUKCUOGLU, 2017). O termo $\beta$ serve como hiperparâmetro que regula o efeito dessa perda.

Como uma extensão desse trabalho, o VQ-VAE2 (RAZAVI; OORD; VINYALS, 2019) utiliza vários conjuntos de variáveis organizados de maneira hierárquica, representando um viés indutivo onde níveis mais altos capturam informação local, enquanto os mais baixos priorizam aspectos globais.

Mesmo que o VAE e o VQVAE não sejam voltados estritamente para a modelagem de sequências, diversas variações deles já foram aplicadas na geração musical (ENGEL et al., 2017; DHARIWAL et al., 2020; DIELEMAN; OORD; SIMONYAN, 2018).

## 1.3 Quantificando Emoções

Para que seja possível o desenvolvimento de modelos generativos de música condicionados por estados afetivos, é necessária uma maneira de se quantificarem esses estados. Alguns modelos para o entendimento das emoções humanas propõem uma estrutura multidimensional, no qual cada emoção varia independentemente das outras. No entanto, algumas evidências (RUSSELL, 1980) sugerem uma organização espacial bidimensional, formada por eixos denominados *Arousal* e *Valence*. Essa caracterização da afetividade humana é conhecida como o modelo circumplexo de Russell (RUSSELL, 1980). Cada eixo

Figura 5 – Modelo Circumplexo de Emoções.


Fonte: Adaptado de (POSNER; RUSSELL; PETERSON, 2005)

pode assumir qualquer valor real no intervalo de $[-1, 1]$. A dimensão de *Arousal* comunica se o estado emocional estudado é pouco intenso (como tédio ou calma) ou muito intenso (como surpresa ou medo). Já o eixo de *Valence* determina se o estado afetivo considerado é desprazeroso (como tristeza ou raiva) ou prazeroso (como alegria e contentamento). A figura 5 ilustra o posicionamento de algumas emoções nesse modelo.

## 1.4 Mátemática e Computação na Composição Musical

Computadores, processos algorítmicos e até mesmo estruturas matemáticas vêm sendo explorados na música há muito tempo. Já na Grécia Antiga, o filósofo Pitágoras propôs o conceito de *Musica Universalis*, no qual cada corpo celeste produzia um som específico de acordo com a forma de sua órbita, ideia essa inspirada na relação que encontrou entre o comprimento de uma corda e a frequência fundamental que emite ao vibrar.

Mais recentemente, no século XX, com o advento dos computadores, cientistas, programadores e músicos passaram a buscar maneiras de incorporar as máquinas na criação

e na performance musical. Já na década de 1950, o compositor Lejaren Hiller utilizou o computador Illiac, da Universidade de Illinois, para criar música experimental através de técnicas como Cadeias de Markov e Passeios Aleatórios (EDWARDS, 2011).

Outro exemplo interessante é o do compositor Iannis Xenakis, que utilizou seu conhecimento em engenharia e matemática para compor obras baseadas em processos estocásticos, onde cada elemento musical era "escolhido" pelo computador com base em distribuições de probabilidades fornecidas pelo programador. Cabia ao compositor, então, organizar o material produzido pelo modelo de forma a construir uma peça (Maurer IV, 1999).

Além da abordagem estocástica, existe também a determinística ou "baseada em regras", na qual os resultados da execução dos algoritmos são fixos. Esses sistemas costumam basear-se em regras composicionais, como é o caso do *Experiments in Musical Intelligence* (COPE, 1989), desenvolvido pelo compositor David Cope, no qual este buscou codificar em programas de computador os estilos de vários compositores, inclusive o seu próprio, de modo que os programas fossem capazes de construir obras completas automaticamente. Posteriormente, Cope desenvolveu um sistema composicional baseado na recombinação de trechos de peças existentes, resultando na criação de obras novas, porém seguindo o mesmo estilo daquelas nas quais se baseavam.

Por fim, a abordagem mais popular atualmente, e também a que é explorada ao longo deste trabalho, consiste no uso de modelos de Inteligência Artificial, e principalmente nas Redes Neurais pertencentes ao Aprendizado de Máquinas. Em 1989, Peter M. Todd usou uma RNN para produzir melodias monofônicas a partir de padrões identificados em uma ou mais melodias fornecidas à rede (TODD, 1989; FERNÁNDEZ; VICO, 2013). Exemplos posteriores incluem sistemas voltados para a harmonização de melodias, para improvisação em jazz e a geração de música polifônica, entre outros (FERNÁNDEZ; VICO, 2013). Porém, a popularização dos modelos de *Deep Learning* trouxe novos paradigmas para o campo da composição algorítmica, permitindo não apenas o desenvolvimento de modelos mais complexos, mas também de alguns capazes de produzir áudio diretamente, não necessitando de representações simbólicas da música.

## 1.5 Deep Learning na Música

Em geral, no contexto de *Deep Learning*, é possível separar os modelos que trabalham com música em dois grupos: aqueles que utilizam áudio digital no domínio *waveform* (por simplicidade, serão incluídos nesse grupo também os que trabalham em espectrogramas construídos a partir do áudio), e aqueles que operam em representações simbólicas que mantêm apenas as instruções para a performance musical, como a informação temporal sobre as notas e a intensidade com a qual devem ser tocada, e descartam todo

o áudio, assemelhando-se a partituras. Cada uma dessas abordagens traz suas próprias vantagens e desafios.

### 1.5.1 Modelos *waveform*

Em geral, informação sonora é armazenada digitalmente por meio de amostras tomadas em intervalos regulares que representam a amplitude da onda em cada instante considerado. Esse tipo de representação tende a ter uma alta densidade; para que 1 segundo de áudio seja armazenado em qualidade de CD, são necessárias 44100 amostras por canal. Assim, do ponto de vista computacional, a modelagem de áudio digital tende a ser bastante custosa, e só recentemente tornou-se possível graças a avanços teóricos e à disponibilidade de *hardware* dedicado, ainda que em certa medida os problemas mencionados persistam (DIELEMAN; OORD; SIMONYAN, 2018).

Por outro lado, contanto que haja os recursos necessários, ao se trabalhar no domínio *waveform*, garante-se que todo o material de interesse poderá ser modelado, incluindo não apenas elementos musicais básicos como notas e acordes, mais quaisquer detalhes relacionados a articulação, dinâmica, ornamentos e outros tipos de expressão musical que, apesar de muitas vezes serem sutis, acabam tendo grande influência sobre o produto final.

Naturalmente, modelos criados para trabalhar com áudio, como os já mencionados Wavenet e SampleRNN (OORD et al., 2016; MEHRI et al., 2016), podem ser utilizados para a geração musical no domínio do *waveform* sem muitas modificações. No entanto, é comum que certas adaptações sejam feitas nessas redes com o intuito de incorporar nos processos de treinamento e inferência conhecimentos específicos sobre estrutura musical.

O Wav2midi2Wav (HAWTHORNE et al., 2019) utiliza o modelo de transcrição Onsets and Frames (HAWTHORNE et al., 2018) para criar uma representação simbólica intermediária das peças de piano, então aplica um Transformer para gerar novo material diretamente no domínio dessa representação, e finalmente um *decoder* Wavenet (OORD et al., 2016) para transformar o conteúdo de volta para áudio.

A mesma ideia de converter o áudio para uma representação intermediária comprimida de forma a reduzir os custos envolvidos na modelagem generativa musical inspirou outros modelos. O Jukebox (DHARIWAL et al., 2020), que atinge resultados estado-da-arte na geração de música em *waveform*, utiliza essa estratégia. O modelo consiste em uma hierarquia de redes VQ-VAE (OORD; VINYALS; KAVUKCUOGLU, 2017; RAZAVI; OORD; VINYALS, 2019), juntamente a um Transformer (VASWANI et al., 2017) que atua no espaço latente. O método torna o processo de geração muito mais rápido, embora ainda demorado. Em (DIELEMAN; OORD; SIMONYAN, 2018), os autores exploram as dificuldades envolvidas na modelagem de áudio e propõem um

modelo semelhante ao anterior para realizar a tarefa.

Em se tratando especificamente de Redes Generativas Adversárias, o WaveGAN (DONAHUE; MCAULEY; PUCKETTE, 2018) é uma GAN capaz de gerar pequenos ($\sim 1s$) trechos de áudio. Por não ser autorregressivo, o processo de geração é rápido e não envolve grande custo computacional. Outro modelo capaz de realizar essa tarefa é o Parallel WaveGAN (YAMAMOTO; SONG; KIM, 2020), que combina a *loss* adversarial da GAN com outras obtidas através de espectrogramas em várias resoluções para aumentar a qualidade do áudio produzido.

Também existem modelos que usam representações intermediárias de áudio ou outros tipos de informações condicionais como entrada. A MelGAN (KUMAR et al., 2019) é uma GAN cujo objetivo é produzir inversão de alta qualidade de espectrogramas Mel[4] para áudio, e a GAN-TTS (BINKOWSKI et al., 2019) é capaz de alcançar síntese de fala condicionada por texto. Ambas empregam arquiteturas diferentes para as redes Geradora e Discriminatória, sendo que a primeira consiste em pilhas de blocos convolucionais residuais (HE et al., 2015) e a segunda é constituída por múltiplas redes (WANG et al., 2018) que operam em diferentes escalas de áudio. Essa estrutura permite que cada submodelo lide com uma faixa diferente do espectro de frequência. Os avanços obtidos por ambos os trabalhos indicam quais tipos de arquiteturas de rede podem ser mais adequadas para GANs que lidam com áudio no domínio de *waveform*.

Finalmente, o modelo GANSynth (ENGEL et al., 2019) gera representações de áudio semelhantes a espectrogramas semi-invertíveis que correspondem a tons únicos produzidos por vários instrumentos. Essas representações são construídas por meio de uma série de transformações que buscam captar os aspectos do áudio que são mais significativos do ponto de vista da percepção humana e que são mais facilmente compreendidos pelas redes.

### 1.5.2 Modelos simbólicos

Há um grande número de modelos generativos de música que usam representações simbólicas (BRIOT; HADJERES; PACHET, 2019). Anteriormente, quase todos eram baseados em RNNs. No entanto, de modo semelhante ao que ocorreu no NLP, diversas das redes estado-da-arte passaram a ser Transformers (VASWANI et al., 2017), ou ao menos começaram a depender significativamente do Mecanismo de Atenção.

#### 1.5.2.1 Representações

Quando lidando com modelos simbólicos, um dos aspectos mais importantes a se considerar é o tipo de representação utilizado durante o treinamento. Por um lado, é

[4] A escala Mel é uma escala perceptual onde os tons são organizados de forma a refletir a maneira como ouvintes humanos percebem distância entre esses tons

desejável eliminar redundâncias e simplificar o conteúdo musical de forma a manter apenas os elementos considerados mais essenciais - tais como informação temporal, melodia, harmonia e dinâmica - assim reduzindo a quantidade de recursos necessários para a modelagem generativa. No entanto, a depender do grau de simplificação aplicado no processo, é possível que sejam perdidos detalhes importantes do material original, de modo que os excertos produzidos pelo modelo treinado soem inexpressivos, repetitivos e desagradáveis. Portanto, existe um equilíbrio a ser buscado no processo de construção de uma representação simbólica, que precisa levar em conta a razão entre a quantidade de recursos disponíveis para o treinamento e a proporção do material original que se deseja manter.

Além disso, a representação criada deve depender do tipo de música a ser modelada, ou seja, a qual ou quais gêneros musicais pertencem essas músicas, em qual ou quais instrumentos elas foram performadas, qual é a duração média das obras, etc. Conclui-se, portanto, que a construção de uma representação simbólica necessita de considerável conhecimento prévio sobre estrutura musical por parte dos pesquisadores envolvidos, e pode variar significativamente de acordo com os fatores apresentados acima.

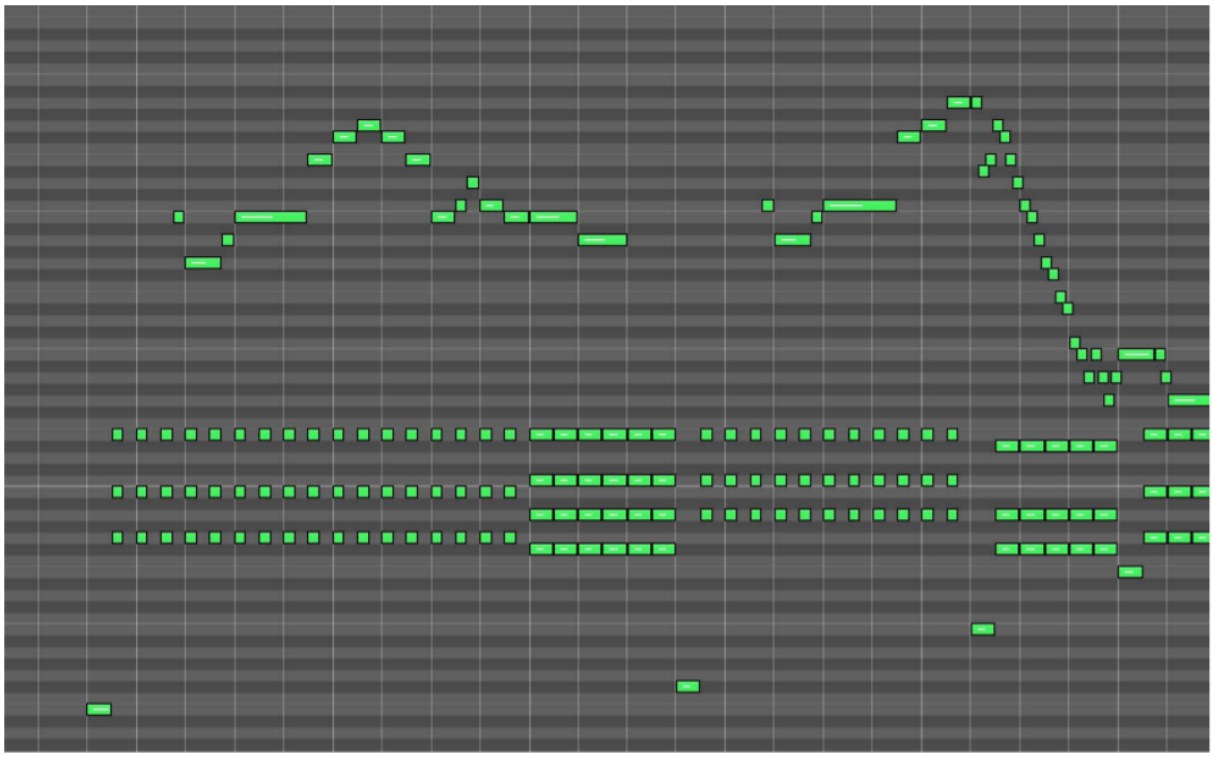
Uma forma comum de simbolizar conteúdo musical no contexto de modelagem generativa é por meio de *piano-rolls*, como ilustrado na Figura 6. Nesse caso, utilizam-se matrizes onde cada linha representa um tom e cada coluna uma unidade temporal. As limitações dessa abordagem incluem a possível esparsidade da matriz construída e a limitação a apenas um instrumento por matriz.

O padrão *Musical Instrument Digital Interface*, ou simplesmente MIDI, é um protocolo tecnológico que permite a comunicação entre instrumentos eletrônicos e computadores por meio de mensagens binárias transmitidas em série que definem as propriedades da articulação da sequência musical a ser tocada. A representação proposta em (OORE et al., 2020) herda características do MIDI, e converte a informação musical em uma sequência de eventos NOTE_ON e NOTE_OFF que indicam os tempos de início e término de cada nota, assim como o tom a ser reproduzido, eventos VELOCITY que comunicam a intensidade da nota[5], e eventos TIME_SHIFT que indicam a passagem do tempo. No trabalho, os autores utilizam essa representação para treinar um modelo que gera performances de piano dotadas de grande expressividade tanto dinâmica quanto temporal na forma de *rubato*.

A representação descrita acima, ainda que bastante poderosa, por permitir uma codificação precisa de cada evento musical representado no MIDI, é adequada apenas para gêneros musicais nos quais as obras são performadas com uma grande liberdade temporal e dinâmica, como é o caso das performances de música Clássica. Porém, no caso

[5] Intensidade é uma propriedade que basicamente indica a força e rapidez aplicadas ao se tocar uma tecla de piano, e está correlacionada ao volume do som a ser reproduzido em decorrência desse evento.

Figura 6 – *Piano Roll*



Fonte: (OORE et al., 2020)

de gêneros como Pop ou Hip-Hop, nos quais normalmente é seguida uma métrica rígida, há a necessidade de imbuir as sequências utilizadas com algum tipo de informação sobre a estrutura rítmica subjacente da música. A representação *revamped MIDI derived-events* (REMI) (HUANG; YANG, 2020) contém eventos NOTE_ON e NOTE_DURATION que indicam início e duração da nota, bem como o tom a ser reproduzido, eventos BEAT e POSITION que indicam o início de um compasso e em qual posição do compasso cada nota deve ser tocada, eventos TEMPO que permitem mudanças locais no andamento da música, e opcionalmente eventos CHORD encarregados de expressar conteúdo harmônico. Já a representação Compound Word (CP) cria uma codificação separada para cada tipo de elemento musical (tom, harmonia, andamento, etc.), permitindo um controle maior sobre a geração.

Em linhas gerais, as representações musicais simbólicas disponíveis na literatura tendem a seguir uma das abordagens apresentadas.

#### 1.5.2.2 Modelos

O Music Transformer (HUANG et al., 2019) se utiliza de uma versão modificada do Mecanismo de Atenção que considera a posição relativa, o tempo relativo e o intervalo entre os elementos da sequência musical. O MuseNet (PAYNE; OPENAI, 2019) usa *embeddings* temporais, melódicos e estruturais para fornecer contexto ao modelo, o que o

torna capaz de criar composições de vários minutos incluindo vários instrumentos e podendo pertencer a vários estilos amplamente diferentes, como *Country*, *Jazz*, ou mesmo em estilos de compositores clássicos específicos. O Pop Music Transformer (HUANG; YANG, 2020), como sugere o nome, é um modelo baseado em Transformers treinado em transcrições para o piano de músicas populares, e faz uso da representação REMI discutida acima, sendo capaz de gerar canções com estrutura rítmica melhor do que modelos anteriores. O Compund Word Transformer (HSIAO et al., 2021), que usa a representação CP também apresentada anteriormente, utiliza uma camada *feed-forward* para cada tipo específico de símbolo musical. A rede neural resultante pode ser vista como um modelo de aprendizado sobre hiper-grafos dinâmicos direcionados.

As GANs também já foram aplicadas à tarefa de geração musical simbólica. O Midinet (YANG; CHOU; YANG, 2017) é constituído por Redes Neurais Convolucionais (CNNs) que formam uma GAN, juntamente a um modelo Condicionador que permite a geração de melodias compasso a compasso considerando o que foi sintetizado previamente. O MuseGAN (DONG et al., 2018) produz sequências musicais *multitrack* polifônicas onde um modelo especializado fica responsável por cada instrumento presente. A DMB-GAN utiliza uma estrutura multi-modelo semelhante à anterior, e conta com o Mecanismo de Atenção para aumentar a consistência dos excertos sintetizados (Guan; Yu; Yang, 2019). Em (ZHANG, 2020), técnicas de *Reinforcement Learning* são utilizadas para possibilitar o treinamento de um algoritmo de geração musical no domínio simbólico baseado em Transformer-GANs, e em (MUHAMED et al., 2021) um resultado semelhante é obtido através da técnica de Gumbel-Softmax (JANG; GU; POOLE, 2017).

Finalmente, no que diz respeito à geração musical condicionada por rótulos emocionais, é limitado o número de modelos capazes de executar essa tarefa, principalmente por conta das dificuldades envolvidas na criação de *datasets* que contenham um número satisfatório de amostras juntamente a valores de *valence* e *arousal* correspondentes. Em (FERREIRA; WHITEHEAD, 2019), os autores treinam uma LSTM para gerar música não condicionalmente, e utilizam-se do fato de que a rede é capaz de aprender a criar uma representação do conteúdo afetivo das músicas apenas através desse aprendizado não-supervisionado para então guiar o processo de geração por meio de um algoritmo genético que atua nas células específicas que ficam responsáveis por construir tal representação. Em (MAKRIS; AGRES; HERREMANS, 2021), rótulos emocionais pré-definidos são associados a cada acorde para que o condicionamento seja feito passo-a-passo. Em (HUNG et al., 2021), um *dataset* novo é criado para possibilitar que um modelo Transformer treinado com a representação CP (HSIAO et al., 2021) crie excertos musicais condicionados por *arousal* e *valence*.

## 1.6 Catalogação musical e Métricas para avaliação de modelos

Há vários *datasets* que reúnem trechos musicais e a esses associam características de interesse para pesquisadores na área de MIR.

O *dataset* MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) (HAWTHORNE et al., 2019) reúne mais de 200 horas de performances virtuosísticas de piano capturadas durante 10 anos da *International Piano-e-Competition*[6]. Ele conta com amostras tanto acústicas quanto simbólicas. A restrição a apenas um instrumento e a grande quantidade de dados presentes no *dataset* configuram-no como uma ótima ferramenta para a tarefa de geração musical.

O Medley-Solos-Db (LOSTANLEN; CELLA, 2016) é uma base de dados contendo aproximadamente 20000 amostras de áudio, cada uma com cerca de $3s$, separadas entre oito classes de instrumentos (clarinete, guitarra elétrica distorcida, voz feminina, flauta, piano, saxofone tenor, trompete e violino). Apesar da duração relativamente curta dos trechos sonoros no *dataset*, dada a complexidade da tarefa de geração musical no domínio de áudio digital, essa restrição pode ser bem vinda.

O *Dataset* AILabs 17k (HSIAO et al., 2021) contém mais de 100 horas de performances de músicas Pop executadas no piano e transcritas para MIDI pelo modelo Onsets and Frames (HAWTHORNE et al., 2018).

A base de dados EMOPIA apresenta passagens musicais separadas de acordo com estados emocionais. Os criadores do projeto usam quatro combinações diferentes de *arousal* e *valence* - ambos podem ser classificados como alto ou baixo - para construírem grupos de tamanho semelhante que podem então ser aplicados na geração condicionada ou classificação emocional. Assim como no caso do *dataset* anterior, os excertos que compõem a base são transcrições para MIDI de canções performadas no piano.

Quando se treina um modelo generativo musical, é necessário considerar quais métodos avaliativos serão utilizados para a comprovação da eficácia do modelo. No caso de redes que trabalham com áudio, é possível aplicar as métricas IS (SALIMANS et al., 2016) e FID (HEUSEL et al., 2017), já canônicas na avaliação de GANs. Para modelos que operam em representações simbólicas, no entanto, dado o fato de que é possível extrair facilmente cada elemento do conteúdo sintetizado, existem também métricas que avaliam propriedades musicalmente relevantes. Algumas propostas de padronizar as métricas para avaliação de modelos simbólicos surgiram para evitar que autores utilizem critérios próprios incompatíveis com outros modelos, ou simplesmente não informativos. Em (YANG; LERCH, 2020), os autores analisam os tipos de *features* mais utilizadas na literatura e propõem um conjunto de métricas que consideram ser interessantes e universalmente aplicáveis. O Muspy (DONG et al., 2020) é uma biblioteca *open source* criada para fornecer ferramentas

[6] http://piano-e-competition.com

essenciais para o desenvolvimento de um sistema de geração musical, incluindo também métricas para a avaliação final desses sistemas, sendo estas retiradas de trabalhos já bem estabelecidos na área. Algumas das características comumente analisadas na literatura são o número total de tons, o intervalo entre o tom mais grave e o mais agudo, a proporção de compassos vazios e o nível de polifonia; para essas, comparam-se as propriedades do conjunto de dados real com um conjunto de músicas geradas. Quanto menor for a diferença entre as estatísticas calculadas dos dois grupos, melhor é o modelo estudado.

# 2 Self Attention GAN aplicada à geração musical condicionada por instrumento

Parte considerável das GANs são baseadas em Redes Neurais Convolucionais (CNNs). Convoluções contam com um forte viés indutivo que prioriza as interações locais, fato este que as torna especialmente adequadas para lidar com imagens naturais. No entanto, em muito casos, é desejável que a rede seja capaz de capturar a estrutura global do objeto considerado. O mecanismo de Atenção (VASWANI et al., 2017) é capaz de modelar as dependências de longo alcance presentes nos dados que recebe, e foi demonstrado (CORDONNIER; LOUKAS; JAGGI, 2019) que as camadas de Atenção não só podem realizar convoluções, mas que na prática, ao lidar com imagens naturais, tendem a fazê-lo. Portanto, o mecanismo atende a padrões de pixels de forma semelhante às camadas convolucionais, fato este que ilustra sua adaptabilidade.

A Self-Attention GAN (ZHANG et al., 2019) combina camadas convolucionais nas redes Geradora e Discriminatória com módulos de Atenção, possibilitando a geração de amostras com estruturas globais muito mais realistas do que aquelas produzidas pelos modelos sem o mecanismo.

Este capítulo da dissertação relata os resultados da aplicação de SAGANs à tarefa de geração de trechos musicais condicionados por instrumento. Utilizando métricas automáticas para a avaliação de GANs disponíveis na literatura, foi comprovada a eficácia do modelo proposto e sua superioridade em relação a um *baseline*. O modelo pode ser aplicado como uma ferramenta na área de Composição Musical Assistida por Máquina. De fato, tecnologias computacionais vêm sendo utilizadas por compositores há décadas. A música eletroacústica, por exemplo, originada na metade do século XX com o movimento francês *musique concrète*, e também o movimento alemão *Elektronische musik*, abrange composições construídas a partir de manipulações eletrônicas feitas em áudio gravado, assim como outras inteiramente constituídas por sons gerados eletronicamente, sendo que algumas dessas obras podem ser tocadas sem a necessidade de músicos humanos. Em se tratando do sistema aqui apresentado, compositores em busca de novas maneiras de usar a tecnologia em suas obras podem utilizar passagens musicais curtas geradas pela rede, combinando-as e organizando-as de forma a produzir uma obra completa.

O modelo é capaz de geração totalmente paralela, o que o torna rápido quando comparado a outros no ramo do *Deep Learning*, uma vez que não é auto-regressivo, portanto não sendo necessário esperar longos períodos ou ter várias GPUs potentes para usá-lo.

## 2.1 Métodos

### 2.1.1 Arquitetura

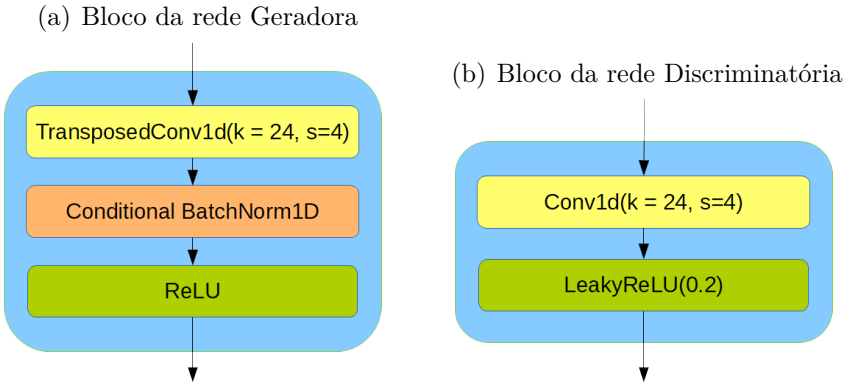
O modelo é constituído por duas redes neurais convolucionais que operam em janelas de 65536 amostras de áudio. O áudio utilizado conta com uma taxa de amostragem de $22050Hz$, ou seja, as redes trabalham com passagens de aproximadamente 3 segundos.

A rede Geradora é composta principalmente por Convoluções Transpostas, que realizam o *upsampling* de um vetor 100-dimensional com entradas amostradas de uma distribuição normal $\mathcal{N}(0, I)$ concatenado a um vetor de condicionamento de comprimento 8 (o número de classes no *dataset*) para uma saída de tamanho 65536 com um canal. Entre certas camadas Convolucionais Transpostas, foi inserido um módulo de Auto-atenção para permitir que a rede capturasse informações não locais. Optou-se por colocar este módulo próximo a uma das camadas de *feature* de alto nível. Este é o posicionamento ideal para o módulo, pois permite que a rede entenda as dependências globais presentes em mapas de *features* maiores (ZHANG et al., 2019). Como feito em (KUMAR et al., 2019), o tamanho do *kernel* das camadas convolucionais transpostas foi escolhido como um múltiplo do tamanho de salto (*stride*) da convolução. Isso é feito para evitar o surgimento de padrões característicos que produzem ruídos de alta frequência no áudio gerado. O condicionamento da rede foi feito através de camadas de normalização *BatchNorm* (IOFFE; SZEGEDY, 2015) especializadas, nas quais cada classe possui um par de parâmetros de viés e escala diferentes. A arquitetura do modelo é apresentada na Tabela 1.

A rede Discriminatória espelha a Geradora em sua estrutura e na posição do bloco de Atenção, mas as Convoluções Transpostas são substituídas por Convoluções com salto que realizam o *downsampling* da entrada. Para alimentar a rede Discriminatória com a informação condicional, foi aplicada uma estratégia simples que consiste na concatenação entre o tensor de entrada e uma projeção do vetor condicional para um mapa de *features*. Este método permitiu ao modelo identificar as características predominantes de cada classe. Também foram feitos experimentos com o método da rede Discriminatória baseada em projeção (MIYATO; KOYAMA, 2018), no entanto, o método resultou em colapso de modos e baixa qualidade nas amostras geradas. Este fato não é uma indicação de que o método de condicionamento é inerentemente inferior a outros, apenas que mais ajustes precisam ser feitos para adaptá-lo à tarefa em questão.

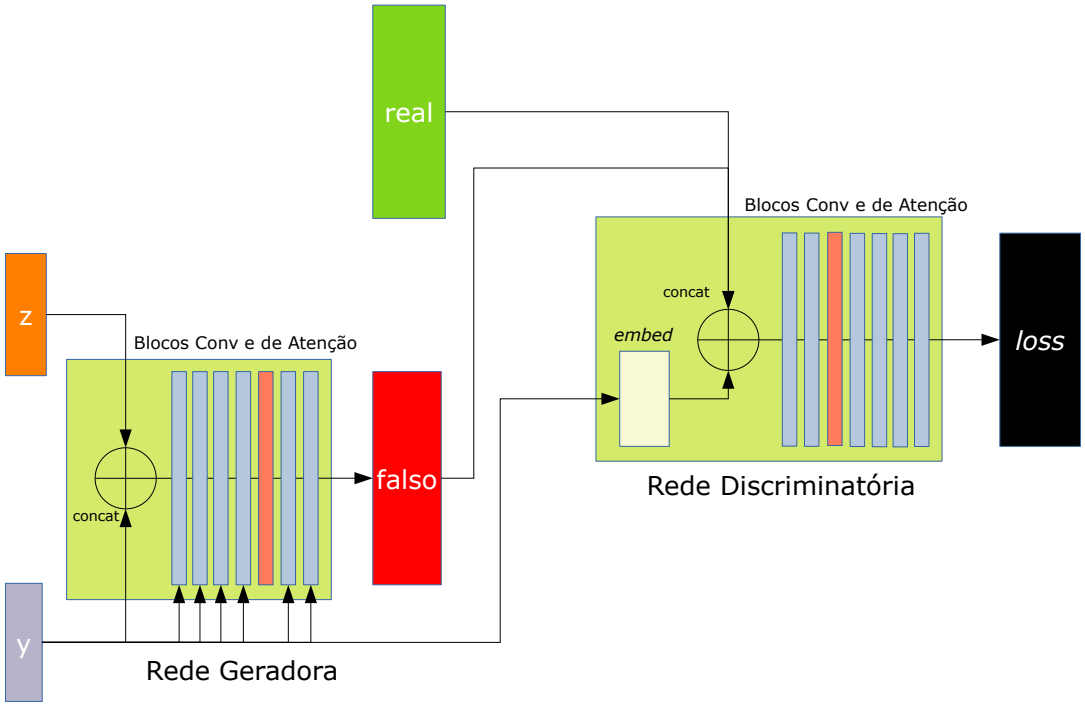
Os detalhes sobre a construção da rede encontram-se na Tabela 2. Além disso, os diagramas dos blocos básicos a partir dos quais os modelos foram criados estão ilustrados na Figura 7, e o esquema geral de uma iteração de treinamento está detalhado na Figura 8.

Figura 7 – Blocos básicos a partir dos quais as redes geradora e discriminatória foram criadas. Os tamanhos de *kernel* e de salto são dados por k e s, respectivamente.




Fonte: (NEVES, 2021)

Figura 8 – Representação em alto nível do esquema de treinamento. Os blocos de Auto-atenção estão destacados em vermelho. $z$ é o vetor de variáveis aleatórias, e $y$ é o vetor de variáveis condicionais.




Fonte: (NEVES, 2021)

Tabela 1 – Arquitetura da rede Geradora. Aqui, os termos entre parênteses após os nomes das tabelas indicam o número de dimensões de entrada e de saída daquela camada. O número ch indica um hipermarâmetro que define a quantidade de canais na menor camada convolucional (para este modelo, 16).

| Layer | Input→Output |
|---|---|
| Linear(100 + 8, 16ch) | $128 \rightarrow 16\text{ch} \cdot 16$ |
| Reshape(16ch, 16) | $16\text{ch} \cdot 16 \rightarrow 16\text{ch} \times 16$ |
| Gen Block(16ch, 16ch) | $16\text{ch} \times 16 \rightarrow 16\text{ch} \times 64$ |
| Gen Block(16ch, 8ch) | $16\text{ch} \times 64 \rightarrow 8\text{ch} \times 56$ |
| Gen Block(8ch, 4ch) | $8\text{ch} \times 256 \rightarrow 4\text{ch} \times 1024$ |
| Gen Block(4ch, 2ch) | $4\text{ch} \times 1024 \rightarrow 2\text{ch} \times 4096$ |
| Attention Block | $2\text{ch} \times 4096 \rightarrow 2\text{ch} \times 4096$ |
| Gen Block(2ch, ch) | $2\text{ch} \times 4096 \rightarrow \text{ch} \times 16384$ |
| TransposedConv(ch, 1) | $\text{ch} \times 16384 \rightarrow 1 \times 65536$ |
| Tanh | $1 \times 65536 \rightarrow 1 \times 65536$ |

Fonte: (NEVES, 2021)

Tabela 2 – Arquitetura da rede Discriminatória. A notação para as camadas e o hiperparâmetro ch possuem o mesmo significado que na rede Geradora.

| Layer | Input→Output |
|---|---|
| Disc Block(1, ch)(x+ Linear(Embed(y))) | $2 \times 65536 \rightarrow \text{ch} \times 16384$ |
| Disc Block(ch, 2ch) | $\text{ch} \times 16384 \rightarrow 2\text{ch} \times 4096$ |
| Attention Block | $2\text{ch} \times 4096 \rightarrow 2\text{ch} \times 4096$ |
| Disc Block(2ch, 4ch) | $2\text{ch} \times 4096 \rightarrow 4\text{ch} \times 1024$ |
| Disc Block(4ch, 8ch) | $4\text{ch} \times 1024 \rightarrow 8\text{ch} \times 256$ |
| Disc Block(8ch, 16ch) | $8\text{ch} \times 256 \rightarrow 16\text{ch} \times 64$ |
| Disc Block(16ch, 16ch) | $16\text{ch} \times 64 \rightarrow 16\text{ch} \times 16$ |
| Flatten | $16\text{ch} \times 16 \rightarrow 16\text{ch} \cdot 16$ |
| Linear(16ch · 16 , 1) | $16\text{ch} \cdot 16 \rightarrow 1$ |

Fonte: (NEVES, 2021)

### 2.1.2 Treinamento

Para o objetivo de treinamento, foi escolhida a função de perda *Hinge Loss* (LIM; YE, 2017), descrita pelas equações 2.1 e 2.2. A normalização espectral (MIYATO et al., 2018) foi aplicada a todas as convoluções e convoluções transpostas em ambas as redes. Utilizou-se o otimizador Adam (KINGMA; BA, 2015) com $\beta_1 = 0.5$ e $\beta_2 = 0.9$. As taxas de aprendizagem foram $1 \cdot 10^{-4}$ para a rede Geradora e $3 \cdot 10^{-4}$, para a Discriminatória, culminando em uma estratégia do tipo *two time-scale update rule* (HEUSEL et al., 2017). A rede Discriminatória foi atualizada uma vez por atualização da Geradora. As escolhas de normalização e função *loss* foram informadas pelo seu uso frequente na literatura. As taxas de aprendizado de $D$ e $G$ foram determinadas por meio de um teste com várias

configurações diferentes. Todos os modelos foram treinados por 100 épocas com um tamanho de *minibatch* de 16 (cerca de 135.000 iterações). O Algoritmo 2 apresenta os passos necessários para a otimização da GAN.

$$\mathcal{L}_{Hinge,D} = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[\max(0, 1 - D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\max(0, 1 + D(G(\boldsymbol{z})))\right] \tag{2.1}$$

$$\mathcal{L}_{Hinge,G} = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[-D(G(\boldsymbol{z}))] \tag{2.2}$$

**Algoritmo 2** – Self-Attention GAN. A rede Discriminatória $D$ tem parâmetros $w$, enquanto a Geradora $G$ tem parâmetros $\theta$.

**Input:** $\alpha_g$, taxa de aprendizado da rede Geradora, $\alpha_d$, taxa de aprendizado da rede Discriminatória. $m$, o tamanho de *minibatch*. $n_{disc}$, o número de iterações de $D$ por iteração de $G$. Opt e ó otimizador (Adam (KINGMA; BA, 2015).

**enquanto** $\theta$ *não convergir* **faça**

  **para** $j$ *em* $0, ..., n_{disc}$ **faça**

    Amostre $\{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$ do conjunto de dados real.

    Amostre $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m}$ de $\mathcal{N}(0, I)$

    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m}\max(0, 1 - D_w(\boldsymbol{x}^{(i)})) + \frac{1}{m}\sum_{i=1}^{m}\max(0, 1 + D_w(G_\theta(\boldsymbol{z}^{(i)})))\right]$

    $w \leftarrow w - \alpha_d \cdot \mathrm{Opt}_D(w, g_w)$

  **fim**

  Amostre $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m}$ de $\mathcal{N}(0, I)$.

  $g_\theta \leftarrow \nabla_\theta \left[-\frac{1}{m}\sum_{i=1}^{m} D_w(G_\theta(\boldsymbol{z}^{(i)}))\right]$

  $\theta \leftarrow \theta - \alpha_g \cdot \mathrm{Opt}_G(\theta, g_\theta)$

**fim**

#### 2.1.2.1 *Data Augmentation*

Trabalhos recentes sugerem que o uso de *data augmentation* no contexto de treinamento de GANs pode melhorar substancialmente o desempenho dos modelos, particularmente quando se trabalha com *datasets* pequenos ($\sim$ 10000 amostras ou menos). Especificamente, em (ZHAO et al., 2020), propõe-se uma estratégia que consiste em aplicar transformações diferenciáveis em todas as amostras recebidas pela rede Discriminatória, o que evita o *overfitting* da rede Discriminatória nos dados de treinamento. Com isso em mente, realizou-se uma série de transformações diferenciáveis para todas as amostras reais e falsas fornecidas à rede.

Como o conjunto de dados utilizado não é muito grande (21572 amostras divididas em 8 classes), esse processo foi essencial para maximizar a qualidade das saídas

Tabela 3 – Procedimentos de *Data Augmentation* Diferenciáveis testados durante o treinamento.

| Procedimento | Descrição |
|---|---|
| NoiseInjection(std) | Adiciona à entrada valores retirados de uma distribuição Gaussiana com média zero e desvio padrão std. |
| TimeShift(s, f) | Desloca a entrada por uma porcentagem aleatória de seu comprimento contida no intervalo [s, f]. |
| Gain(min_db, max_db) | Modifica o volume total da entrada por uma quantia aleatória contida no intervalo [min_db, max_db], onde os valores são dados em Db. |

Fonte: (NEVES, 2021)

da rede Geradora. Testaram-se várias combinações dos procedimentos de *data augmentation*, que estão descritos na Tabela 3. No final, decidiu-se por *Gain* seguido por *Time Shifting*, pois estes forneceram uma boa performance em relação ao custo computacional demandado.

## 2.2 Experimentos e Discussão

#### 2.2.0.1 Dataset

Neste trabalho, foi utilizado o *dataset* Medley-solos-Db (LOSTANLEN; CELLA, 2016). A base consiste em 21572 clipes de 3 segundos, sendo que cada clipe contém um único instrumento de um total de 8, que são clarinete, guitarra elétrica distorcida, voz feminina, flauta, piano, saxofone tenor, trompete e violino. O *dataset* também é dividido em conjuntos de treinamento, validação e teste, mas, para os propósitos deste trabalho, foi melhor tratá-lo como um único conjunto de treinamento. Os dados consistiam originalmente em arquivos em formato WAV amostrados a $44100Hz$ com um único canal a uma profundidade de 32 bits e uma duração fixa de 2972 milissegundos. No entanto, por conta das limitações nos recursos computacionais disponíveis, mudamos a taxa de amostragem de todos os clipes para $22050Hz$, de forma que cada clipe usado durante o treinamento teve 65536 amostras no total.

Embora o conjunto de dados tenha sido projetado principalmente para auxiliar o desenvolvimento de algoritmos de reconhecimento de instrumentos em gravações solo, a separação de segmentos de áudio em classes com base no instrumento presente fornece a estrutura ideal para a tarefa de geração condicional buscada. Além disso, embora a duração dos trechos do conjunto de dados seja curta ($\sim$ 3s), gerar novos dados que se enquadrem nessa distribuição ainda é uma tarefa difícil, pois o número de amostras que compõem um

único desses arquivos é muito grande, e cada instrumento tem seu próprio conjunto de características exclusivas, incluindo timbre, faixa de frequência e faixa dinâmica.

#### 2.2.0.2 Experimentos

O *Inception Score* (IS) (SALIMANS et al., 2016) e a *Frechét Inception Distance* (FID) (HEUSEL et al., 2017), apresentados na seção 1.1.1, foram usados como medidas quantitativas do desempenho dos modelos. Para GANs treinadas em imagens, IS e FID são calculados em um conjunto de vetores de *features* obtidos da rede neural classificadora de imagens *Inception-V3* (SZEGEDY et al., 2015) pré-treinada no *dataset* ImageNet (DENG et al., 2009). Para o IS, a variedade e fidelidade das amostras sintéticas é avaliada, e pontuações mais altas são melhores. Para a FID, as estatísticas dos vetores de *features* correspondentes às amostras reais e geradas são comparadas, e pontuações mais baixas indicam melhor qualidade e diversidade entre as imagens.

Como a rede aqui discutida trabalha com áudio e não com imagens, não foi possível usar o modelo *Inception*. Em vez disso, foi construída uma rede convolucional classificatória simples que trabalha diretamente com áudio. Ela foi treinada para realizar uma tarefa de classificação de instrumentos nas 8 classes do mesmo *dataset* que usamos para treinar a GAN. Sua estrutura é semelhante à da rede Discriminatória. No entanto, as convoluções com salto são substituídas por camadas de *Average Pooling*, e a normalização por *batch* é usada no lugar da normalização espectral. Ele obtém uma precisão de 69% no conjunto de teste.

Deve-se notar que todas as métricas automáticas usadas na avaliação de GANs têm certos problemas inerentes (BARRATT; SHARMA, 2018; BORJI, 2018), que se tornam ainda mais pungentes quando elas são aplicadas em bases de dados pequenas, como a usada aqui. No entanto, elas ainda são a melhor alternativa quando se deseja comparar entre diferentes técnicas, arquiteturas e estratégias de treinamento utilizadas no contexto de uma pesquisa envolvendo GANs.

Realizou-se um estudo de ablação para comparar o modelo *baseline*, que não contém módulos de Auto-atenção, com a SAGAN. Estes modelos também foram comparados com suas versões treinadas sem o *data augmentation*. Os resultados dos experimentos estão dispostos na Tabela 5, e a Tabela 4 apresenta os hiperparâmetros usados durante o treinamento.

Os resultados sugerem que, assim como no caso das imagens, como era de se esperar, as GANs que trabalham com áudio se beneficiam da inserção do mecanismo de Auto-Atenção, pois este auxilia as redes a captar informações globais que seriam perdidas caso apenas convoluções fossem usadas. Os procedimentos de *data augmentation* empregados durante o treinamento também melhoraram o desempenho do modelo por uma margem significativa. A Figura 9 mostra o progresso de treino da *baseline* e do modelo

Tabela 4 – Hiperparâmetros utilizados durante o treinamento.

| Nome | Valor |
|---|---|
| Tamanho de *batch* | 16 |
| Loss | Hinge Loss (LIM; YE, 2017) |
| Atualizações de **D** por atualizações de **G** | 1 |
| Otimizador | **G** - Adam(lr=$1e-4$, $\beta_1 = 0.5$, $\beta_2 = 0.9$)<br>**D** - Adam (lr=$3e-4$, $\beta_1 = 0.5$, $\beta_2 = 0.9$) |
| Épocas | 100 |

Fonte: (NEVES, 2021)

Tabela 5 – *Inception Score* e *Frechét Inception Distance* para todas as versões da rede. As pontuações abaixo são as melhores obtidas para cada modelo durante o treinamento. Os valores em negrito indicam as melhores performances.

| Modelo | IS | FID |
|---|---|---|
| GAN (s/ *augmentation*) | 2.24 | 557.55 |
| SA-GAN (s/ *augmentation*) | 2.30 | 597.84 |
| GAN (c/ *augmentation*) | **3.83** | 86.67 |
| SA-GAN (c/ *augmentation*) | 3.76 | **45.85** |
| Dados Reais | 4.78 | 0.00 |

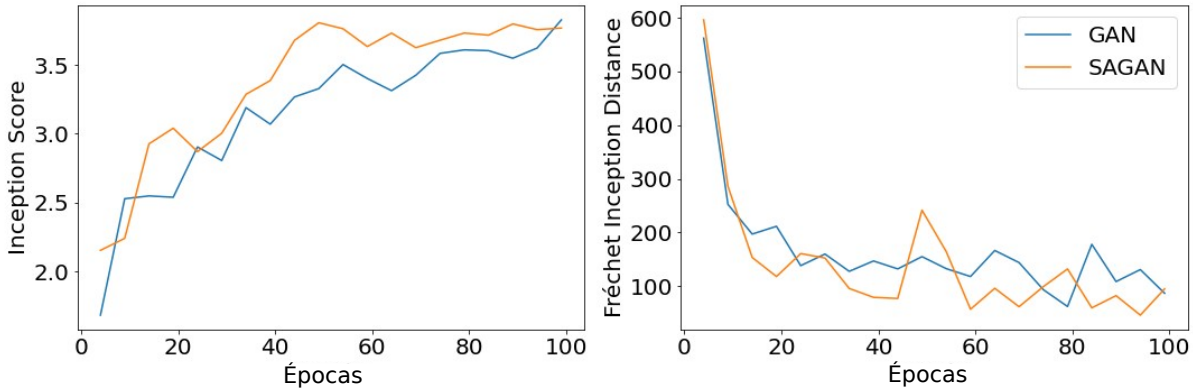
Fonte: (NEVES, 2021)

proposto, ambos treinados com *data augmentation*, em termos de IS e FID. Percebe-se que embora a Tabela 5 indique uma performance ligeiramente melhor para a GAN com respeito ao IS, a Figura 9 deixa claro que a performance da SAGAN proposta permanece consistentemente melhor do que a da *baseline* durante todo o processo de otimização. Além do mais, a FID alcançada pelo modelo proposto é quase metade daquele que foi obtido pela *baseline*, e já foi demonstrado que a FID é consistente com o julgamento humano e mais robusto que o IS (HEUSEL et al., 2017; BORJI, 2018).

O tempo de treinamento total da GAN foi de 7h30m, enquanto o da SAGAN proposta foi de 10h30m. As redes foram treinadas em uma máquina com uma GPU NVIDIA V100, CPU , Intel(R) Xeon(R) CPU @ 2.30GHz, 25GB de RAM com Sistema Operacional Linux Ubuntu 18.04, e o código foi escrito no *framework* Pytorch 1.9.0 (PASZKE et al., 2019). O tempo adicional de treinamento pode ser explicado pelo custo computacional envolvido no cálculo das matrizes de Atenção, e apesar de ser significativo, justifica-se pelo ganho em performance.

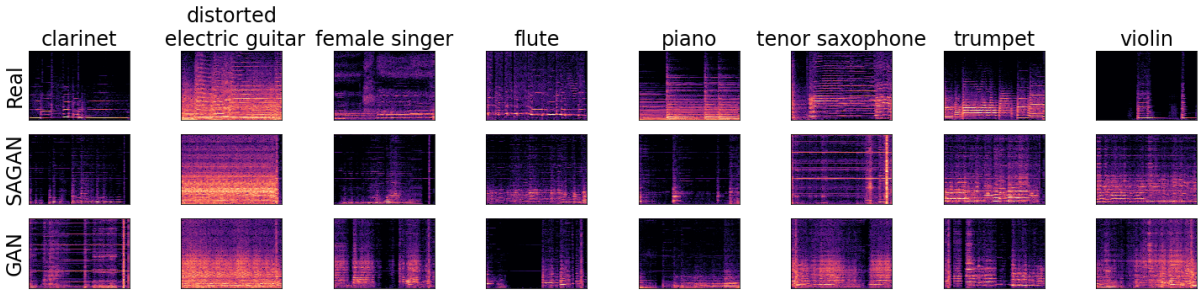
Para uma visualização dos resultados das redes, a Figura 10 apresenta espectrogramas extraídos de amostras produzidas pela GAN e pela SAGAN, assim como de amostras reais, para todas as classes de instrumentos do *dataset*.

Figura 9 – Comparação entre os progressos de treinamento da *baseline* e do modelo proposto com respeito a IS e FID.




Fonte: (NEVES, 2021)

Figura 10 – Espectrogramas correspondentes a amostras do conjunto de dados real e de áudio gerado pelas redes.




Fonte: (NEVES, 2021)

Mesmo que o realismo musical das amostras produzidas pelo modelo seja ainda limitado, sua superioridade em comparação com a *baseline* que representa o estado-da-arte aponta para caminhos promissores em termos de pesquisas futuras, e sugere aplicações interessantes, como a geração automática de música e a composição assistida por computador.

# 3 VQGAN aplicada à codificação de amostras musicais

Dada a densidade extrema e dados do áudio digital, gerar música que seja coerente em longas escalas temporais (da ordem de minutos ou mais) usando modelos de *Deep Learning* ainda é um desafio considerável, sendo necessária uma grande quantidade de poder computacional até mesmo para a modelagem de poucos segundos de áudio. Redes Neurais que trabalham com representações simbólicas, como *piano-rolls*, lidam parcialmente com esse problema por meio da construção de códigos que carregam um conjunto desejado de características importantes da música (nota, tempo, velocidade, duração, etc), descartando a informação acústica em si. No entanto, algumas das escolhas feitas na criação dessas representações podem fazer com que elas negligenciem aspectos sutis, porém importantes, que contribuem para a qualidade final de uma música.

Nesta seção do projeto, um híbrido de *Vector Quantized Variational Autoencoder* (VQ-VAE) (OORD; VINYALS; KAVUKCUOGLU, 2017) com Rede Generativa Adversária (GAN) (GOODFELLOW et al., 2014) foi treinado para construir uma representação altamente comprimida mas consideravelmente expressiva de trechos musicais curtos originados de performances de peças de piano. A representação criada pelo modelo é análoga a uma representação musical simbólica criada manualmente. Porém, com a ajuda de um sinal de aprendizagem que combina conhecimento prévio sobre a maneira como seres humanos percebem o som com a expressividade de um sinal adversarial, a alta capacidade representativa dos modelos de *Deep Learning* foi aplicada na automatização deste processo.

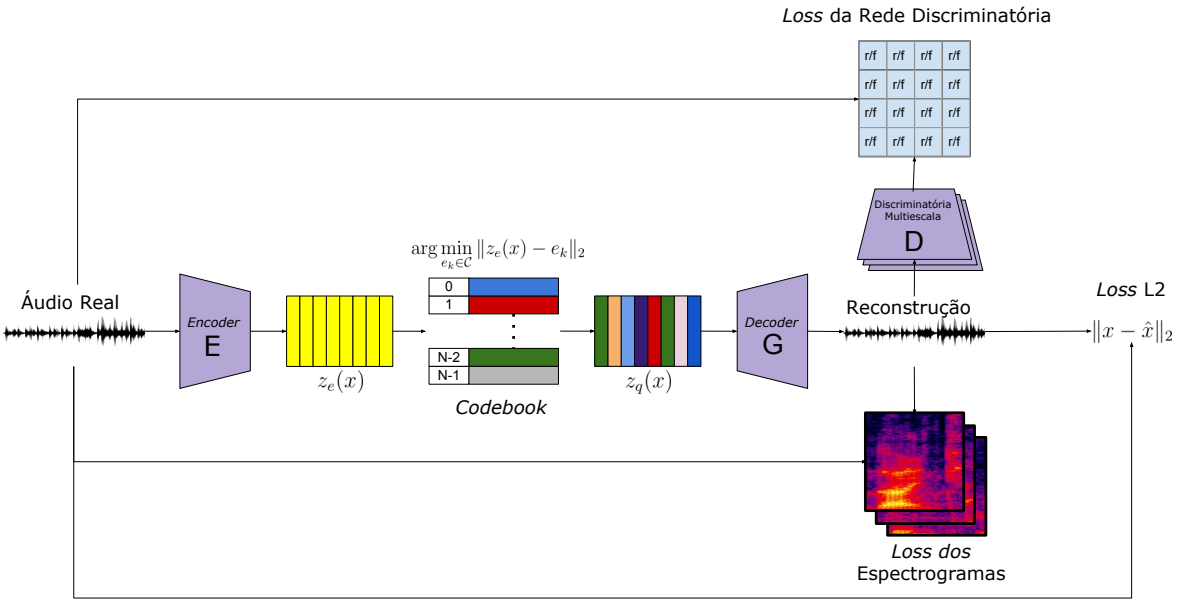
A rede *Autoencoder* treinada pode ser combinada com um modelo sequencial capaz de realizar modelagem generativa diretamente nos códigos gerados pelo modelo, assim reduzindo significativamente os custos envolvidos na geração musical automatizada em formato de áudio digital.

## 3.1 Métodos

### 3.1.1 Modelos

O VQ-VAE é composto por um *encoder* e um *decoder* acoplados a um módulo *Vector Quantizer* (OORD; VINYALS; KAVUKCUOGLU, 2017). Os dois primeiros são redes neurais convolucionais com blocos de convoluções estacadas no estilo ResNet (HE et al., 2015), mas com parâmetros de salto diferentes que, juntamente a módulos de

Figura 11 – Estrutura de uma iteração de Treinamento do modelo. Os termos $z_e(x)$ e $z_q(x)$ representam o *output* do Encoder e o *embedding* do Vector Quantizer criados a partir dos elementos do *codebook*. Os modelos estão nomeados, e os termos de perda explicados ao longo do texto. Mais detalhes sobre as GANs e sobre o VQVAE estão dados nas seções 1.1 e 1.2




Fonte: (NEVES, 2021)

Auto-Atenção (VASWANI et al., 2017; ZHANG et al., 2019), capturam as dependências dos dados ao longo de múltiplas camadas.

A rede Discriminatória constitui-se em uma hierarquia de redes convolucionais, cada uma das quais trabalhando em uma escala diferente de áudio. Ela se aproxima da rede proposta em (KUMAR et al., 2019), consistindo de três sub-redes ($D1, D2, D3$). A primeira opera no áudio original, enquanto a segunda e terceira trabalham no áudio com um número de amostras reduzido em fatores de 2 e 4, respectivamente, através de camadas *Average Pooling*. Alguns trabalhos da área de geração de áudio através de redes adversárias indicam que redes que operam em áudio digital podem se beneficiar dessa estrutura multi-escala, pois ela permite que cada submodelo lide com uma faixa de frequências especifica e que capture detalhes diferentes do *intput* (KUMAR et al., 2019; BINKOWSKI et al., 2019).

### 3.1.2 Treinamento

Um dos pontos negativos da abordagem proposta em (DHARIWAL et al., 2020) é a necessidade de uma hierarquia de códigos que deixam a geração mais devagar, além de aumentar os custos computacionais associados (de acordo com os autores, em uma placa de vídeo NVIDIA V100 com 16GBs de RAM, é necessária uma hora para a geração de 1 minuto de áudio). A abordagem aqui proposta, por outro lado, envolve apenas um conjunto

de códigos latentes, e um nível de compressão maior do que o modelo mencionado.

O objetivo de treino da rede é:

$$\mathcal{L}_{VQGAN} = \mathcal{L}_{VQ} + \alpha\mathcal{L}_{GAN}, \tag{3.1}$$

onde $\mathcal{L}_{VQ} = \mathcal{L}_{VAE} + \gamma\mathcal{L}_{multispec}$. Nesta última equação, o primeiro termo é a função *loss* padrão do modelo VQVAE, dado pela equação 3.2:

$$\mathcal{L}_{VAE} = \|x - \hat{x}\|_2^2 + \|sg[z_e(x)] - z_q(x)\|_2^2 + \beta\|z_e(x) - sg[z_q(x)]\|_2^2, \tag{3.2}$$

onde o primeiro termo é uma *loss* $L_2$ simples entre o material real e sua reconstrução, o segundo e o terceiro são responsáveis por penalizar as distâncias entre a representação construída por $E$ e os elementos do *codebook* (OORD; VINYALS; KAVUKCUOGLU, 2017), e $sg[\cdot]$ denota o operador *stop-gradient*, que impede a propagação dos gradientes pela rede.

Para a construção de $\mathcal{L}_{multispec}$, a função *loss* calculada a partir de espectrogramas proposta em (DHARIWAL et al., 2020) foi substituída por uma que está melhor alinhada com a maneira como seres humanos percebem mudanças tanto nos espectros de amplitude quanto de frequência. Esta função *loss* diferencia ente espectrogramas de Mel em escala-log construídos a partir do áudio original e de sua reconstrução. Formalmente, dado um conjunto de hiperparâmetros $(N, S, W, M)$ representando NFFT (Número de Transformadas de Fourier), Tamanho de Salto (*Hop size*), Tamanho de Janela (*Window size*) e número de Faixas de Mel, respectivamente, cada fator na *loss* pode ser escrito como:

$$\mathcal{L}_{spec} = \frac{1}{N}\|\log(\mathrm{MF}(M)\cdot|\mathrm{STFT}(x;N,S,W)|) - \log(\mathrm{MF}(M)\cdot|\mathrm{STFT}(\hat{x};N,S,W)|)\|_F, \tag{3.3}$$

onde MF é a matriz de filtros de Mel que projeta intervalos FFT para intervalos de frequência de Mel. Finalmente, a função *loss* multiescala de espectrogramas Mel é dada pela Equação 3.4:

$$\mathcal{L}_{multispec} = \frac{1}{M}\sum_{m=1}^{M}\mathcal{L}_{spec}(N_m, S_m, W_m, M_m), \tag{3.4}$$

onde cada fator $\mathcal{L}_{multispec}$ é computado com um conjunto de hiperparâmetros único. Uma descrição completa dos hiperparâmetros está dada na Tabela 6.

Finalmente, para o objetivo da GAN, $\mathcal{L}_{GAN}$, foi escolhida *Hinge Loss* (LIM; YE, 2017), que está dada nas Equações 3.5 e 3.6 para D e G, respectivamente:

$$\mathcal{L}_{Hinge,D} = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\left[\max(0, 1 - D(\boldsymbol{x})\right] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[\max(0, 1 + D(G(\boldsymbol{z})))\right] \tag{3.5}$$

Tabela 6 – Hiperparâmetros utilizados durante o treinamento.

| Nome | Valor |
|---|---|
| Tamanho de *batch* | 2 |
| Loss | Hinge Loss (LIM; YE, 2017),<br>VQVAE Loss<br>(OORD; VINYALS; KAVUKCUOGLU, 2017) |
| Atualizações de **D**<br>por atualizações de **G** | 1 |
| Otimizador | **G** - Adam(lr=$5e-5$, $\beta_1 = 0.5$, $\beta_2 = 0.9$)<br>**D** - Adam (lr=$2e-4$, $\beta_1 = 0.5$, $\beta_2 = 0.9$) |
| Iterações de treino | $\sim$ 300.000 |
| NFFT | 2048, 1024, 512 |
| Tamanho de Salto | 240, 120, 50 |
| Tamanho de Janela | 1200, 600, 240 |
| Número de Bandas de Mel | 512, 256, 128 |
| $\alpha$ (*loss*) | Calculada iterativamente conforme a razão entre<br>os gradientes das *losses* do VQVAE da rede Geradora,<br>tomados na última camada do Decoder. Esse processo<br>balanceia a magnitude dos dois termos,<br>como visto em (ESSER; ROMBACH; OMMER, 2021). |
| $\beta$ (VQVAE) | 0.25 |
| $\gamma$ (*Spec loss*) | 0.01 |

Fonte: (NEVES, 2021)

$$\mathcal{L}_{Hinge,G} = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[-D(G(\boldsymbol{z}))] \tag{3.6}$$

Note que G, sendo ao mesmo tempo um VQ-VAE e uma Rede Geradora, deve otimizar o objetivo combinado 3.1, enquanto D é simplesmente uma rede Discriminatória, otimizando apenas 2.1. A Figura 11 fornece uma visão geral do modelo, e o Algoritmo 3

detalha o treinamento da rede.

**Algoritmo 3** – VQ-GAN. A rede Discriminatória tem parâmetros $w$, enquanto a Geradora, $G$, que é composta por um *Encoder* $E$, um *Decoder* $H$ e um *Vector Quantizer* $VQ$, tem parâmetros $\theta$.

**Input:** $\alpha_g$, taxa de aprendizado da rede Geradora, $\alpha_d$, taxa de aprendizado da rede Discriminatória. m, o tamanho de *minibatch*. $n_{disc}$, o número de iterações de $D$ por iteração de $G$. Opt é o Otimizador.

**enquanto** $\theta$ *não convergir* **faça**

  **para** $j$ *em* $0, ..., n_{disc}$ **faça**

    Amostre $\{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$ do conjunto de dados real.

    $z_e^{(i)}(\boldsymbol{x}) \leftarrow E_\theta(\boldsymbol{x}^{(i)})$

    $z_q^{(i)}(\boldsymbol{x}) \leftarrow \{\boldsymbol{e}_j \quad \text{t.q} \quad j = \arg\min_{\boldsymbol{e}_k \in \mathcal{C}} \|z_e^{(i)}(\boldsymbol{x}) - \boldsymbol{e}_k\|_2\}$

    $\hat{\boldsymbol{x}}^{(i)} \leftarrow H_\theta(\boldsymbol{z}_q^{(i)})$

    $$g_w \leftarrow \nabla_w \left[ \frac{1}{m} \sum_{i=1}^{m} \min(0, D_w(\boldsymbol{x}^{(i)}) - 1) + \frac{1}{m} \sum_{i=1}^{m} \min(0, -D_w(\hat{\boldsymbol{x}}^{(i)}) - 1) \right]$$

    $w \leftarrow w - \alpha_d \cdot \text{Opt}_D(w, g_w)$

  **fim**

  Amostre $\{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$ do conjunto de dados real.

  $z_e^{(i)}(\boldsymbol{x}) \leftarrow E_\theta(\boldsymbol{x}^{(i)})$

  $z_q^{(i)}(\boldsymbol{x}) \leftarrow \{\boldsymbol{e}_j \quad \text{t.q} \quad j = \arg\min_{\boldsymbol{e}_k \in \mathcal{C}} \|z_e^{(i)}(\boldsymbol{x}) - \boldsymbol{e}_k\|_2\}$

  $\hat{\boldsymbol{x}}^{(i)} \leftarrow H_\theta(\boldsymbol{z}_q^{(i)})$

  $$\begin{aligned} g_\theta \leftarrow \nabla_\theta \Bigg[ \frac{1}{m} \sum_{i=1}^{m} \Big\{ & -\alpha D_w(\hat{\boldsymbol{x}}^{(i)}) + \|\boldsymbol{x}^{(i)} - \hat{\boldsymbol{x}}^{(i)}\|_2^2 \\ & + \|sg[z_e^{(i)}(\boldsymbol{x})] - z_q^{(i)}(\boldsymbol{x})\|_2^2 + \beta \|z_e^{(i)}(\boldsymbol{x}) - sg[z_q^{(i)}(\boldsymbol{x})]\|_2^2 \\ & + \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \| \log(\text{MF}(M_m) \cdot |\text{STFT}(\boldsymbol{x}^{(i)}; N_m, S_m, W_m)|) \\ & - \log(\text{MF}(M_m) \cdot |\text{STFT}(\hat{\boldsymbol{x}}^{(i)}; N_m, S_m, W_m)|) \|_F \Big\} \Bigg] \end{aligned}$$

  $\theta \leftarrow \theta - \alpha_g \cdot \text{Opt}_G(\theta, g_\theta)$

**fim**

O objetivo escolhido foi a *Hinge loss* (LIM; YE, 2017), como está escrita nas equações 2.1 e 2.2. Foi aplicada a *Weight Normalization* (SALIMANS; KINGMA, 2016) em todas as convoluções da rede Discriminatória. O Otimizador Adam (KINGMA; BA, 2015) foi escolhido, com hiperparâmetros $\beta_1 = 0.5$ e $\beta_2 = 0.9$. A rede Geradora foi otimizada com uma taxa de aprendizado de $5 \cdot 10^{-5}$, e a Discriminatória com uma taxa de $2 \cdot 10^{-4}$. O número de iterações foi o mesmo para as duas redes. O modelo foi treinado por cerca de 300.000 iterações com *minibatches* de tamanho 2, e foram utilizados trechos sonoras de aproximadamente 3 segundos de duração (131072 amostras de áudio). Os procedimentos de *Data Augmentation* detalhados na seção anterior também foram aplicados no treinamento da rede.

## 3.2 Experimentos

### 3.2.1 Dataset

Os experimentos foram conduzidos do *dataset* MAESTRO (HAWTHORNE et al., 2019), uma coleção contendo mais de 200 horas de obras dos períodos barroco, clássico e romântico performadas durante 9 anos da *International Piano e-Competition*[1]. Levando-se em consideração que esta é uma das maiores bases de dados musicais disponíveis, e que as músicas nela contidas são restritas a apenas um instrumento e a uma quantidade limitada de estilos, é fácil ver que a base funciona como uma campo de testes interessante para a tarefa em questão. Para treinamento, foi utilizado 80% do *dataset*, e a avaliação foi realizada com um subconjunto correspondente a 10% do *dataset*. O *dataset* já foi utilizado em diversos outros trabalhos na área de *Musical Information Retrieval*, especialmente na área de Geração Musical, como em (HAWTHORNE et al., 2019), onde um modelo constituído por um *Encoder* converte áudio para MIDI, um modelo Sequencial gera dados nessa representação, e finalmente um *Decoder* converte o conteúdo produzido de volta ao áudio. Este modelo tem algumas similaridades com o aqui proposto. No entanto, a representação intermediária apresentada aqui é construída automaticamente durante o treinamento, não escolhida previamente, e o modelo proposto precisa apenas de áudio para ser treinado, tendo o potencial de ser utilizado para a geração simultânea de qualquer combinação de instrumentos, sem a necessidade de submodelos diferentes para cada um desses. O modelo também assemelha-se aos propostos em (DHARIWAL et al., 2020) e (DIELEMAN; OORD; SIMONYAN, 2018). No entanto, a rede aqui apresentada possui o diferencial de ter sido treinada adversarialmente.

### 3.2.2 Avaliação

A rede VQGAN proposta foi comparada com o VQVAE utilizado no modelo Jukebox (DHARIWAL et al., 2020). De acordo com os autores do trabalho, esta rede possui 2 milhões de parâmetros, enquanto a proposta aqui possui 150 milhões. No entanto, o Jukebox necessita, além do modelo que trabalha autorregressivamente nos códigos correspondentes ao nível mais alto de compressão, de redes *Upsamplers* que traduzem entre os vários níveis de códigos usados pelo modelo e, segundo os autores, estas redes possuem um total de 1 bilhão de parâmetros. Dado o fato de que o modelo apresentado gera apenas um conjunto de códigos que poderiam ser modelados por uma única rede autorregressiva, não sendo necessário o uso de *Upsamplers*, e que as sequências produzidas possuem uma extensão 128 vezes menor do que o áudio original - enquanto os níveis usados pelo Jukebox são 4, 32 e 128 vezes menores - o número adicional de parâmetros pode ser justificado.

[1] https://piano-e-competition.com/

Um outro adendo que deve ser feito diz respeito aos conjuntos de treinamento utilizados na otimização de cada modelo. Como já discutido, a rede proposta foi treinada apenas em músicas clássicas performadas no piano. Enquanto isso, para o treinamento do Jukebox, utilizou-se uma enorme quantidade de amostras musicais de diversos gêneros, incluindo música clássica.

Dadas as diferenças entre os algoritmos, a comparação realizada aqui não tem o intuito de demonstrar a superioridade de nenhum dos modelos, mas sim de ilustrar que a rede apresentada possui uma performance competitiva quando comparada a uma semelhante que é considerada o estado-da-arte na tarefa, tendo a vantagem de produzir apenas um conjunto de códigos latentes em um nível de compressão maior do que o Jukebox, sendo necessária assim uma quantidade menor de recursos para a modelagem desses códigos. Para que uma comparação mais profunda pudesse ser feita, seria necessário que fosse treinado também o modelo gerador de códigos. No entanto, os recursos computacionais disponíveis durante a realização do projeto não foram suficientes para isso.

Os testes foram feitos através da Fréchet Audio Distance (FAD) (KILGOUR et al., 2019), uma adaptação para o domínio do áudio da métrica Fréchet Inception Distance (FID) (HEUSEL et al., 2017) que é comumente utilizada na avaliação de modelos generativos e que já foi apresentada na seção 1.1.1. Neste caso, as *features* aplicadas no cálculo da métrica vêm do modelo de classificação de áudio VGGish (HERSHEY et al., 2017), treinado em um *dataset* contendo 70 milhões de amostras de áudio retiradas de vídeos disponíveis *online*.

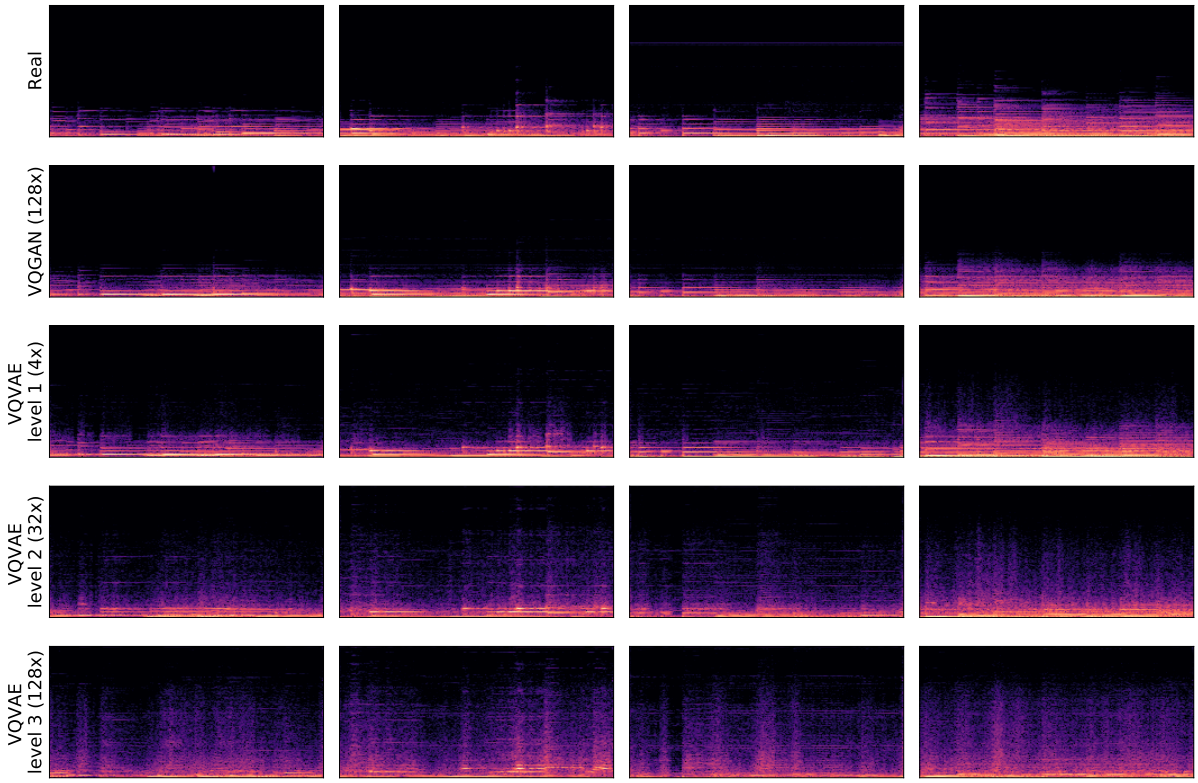
A Tabela 7 apresenta os resultados do experimento. Como é possível ver, o modelo proposto obtém um resultado melhor do que os dois maiores Níveis de compressão do Jukebox, ficando atrás apenas daquele que é $32x$ menos comprimido, porém estando consideravelmente à frente do nível $4x$ mais comprimido e do equiparavelmente comprimido. Esses números demonstram o potencial da VQGAN, e deixam claro que resultados positivos podem vir da combinação de GANs com *Autoencoders*. Por fim, assim como foi feito no último capítulo, espectrogramas das reconstruções feitas por cada modelo estão ilustrados na Figura 12.

Tabela 7 – *Frechét Inception Distance* para o modelo VQGAN e para o VQVAE. O valor em negrito indica a melhor performance.

| Modelo | FAD |
|---|---|
| VQGAN (128x) | 0.0510 |
| VQVAE Nível 1 (4x) | **0.0025** |
| VQVAE Nível 2 (32x) | 0.3713 |
| VQVAE Nível 3 (128x) | 1.5073 |

Fonte: (NEVES, 2021)

Figura 12 – Espectrogramas correspondentes a amostras do conjunto de dados real e de áudio reconstruído pelas redes.




Fonte: (NEVES, 2021)

# 4 Geração musical condicionada por sentimentos com Transformer-GAN

Um das características mais interessantes da música é sua capacidade de evocar estados afetivos específicos no ser humano. Há uma área de pesquisa exclusivamente dedicada ao desenvolvimento de tecnologias para reconhecimento de emoções em passagens musicais (KIM et al., 2010), e serviços de *streaming* utilizam-se dessas tecnologias para melhorar seus algoritmos de recomendação de conteúdo. Nesse contexto, as emoções humanas são quantificadas de acordo com as dimensões de *Valence* e *Arousal* pertencentes ao modelo circumplexo de Russel (RUSSELL, 1980).

Uma tarefa relacionada ao reconhecimento de emoções na música, e que essencialmente funciona como uma inversão dessa, é a de geração musical condicionada por emoções. A ideia é treinar um modelo para associar padrões melódicos, rítmicos e harmônicos a estados afetivos específicos e depois gerar novas passagens que imprimam esses mesmos estados em pessoas que venham a escutá-las. Para isso, é necessário que existam bases de dados contendo excertos musicais e os valores de *Arousal* e *Valence* aos quais correspondem, de acordo com a percepção de ouvintes humanos.

O treinamento de redes generativas de música simbólica geralmente é feito através do método de *teacher forcing*, no qual a rede tenta prever o próximo elemento de uma sequência vinda do conjunto de treino com base nos elementos anteriores. Porém, essa abordagem apresenta um problema significativo: as distribuições de dados vistas pela rede durante o treinamento e durante a geração são diferentes - no segundo caso, a cada passo, a rede completa as sequências que ela própria gerou. Uma das estratégias possíveis para lidar com tal discrepância é adicionar uma rede Discriminatória no processo, ficando esta responsável por diferenciar entre as sequências do conjunto de treinamento e as produzidas integralmente pela rede Geradora. Dessa forma, o algoritmo transforma-se em uma GAN.

Esta seção relata os resultados de um experimento no qual treinou-se uma GAN (GOODFELLOW et al., 2014) baseada em Transformers (VASWANI et al., 2017) para gerar excertos musicais condicionados por *Arousal* e *Valence*. O modelo opera no domínio simbólico, e a técnica de Gumbel-Softmax (JANG; GU; POOLE, 2017) foi utilizada para possibilitar a otimização do modelo mesmo no contexto discreto.

## 4.1 Métodos

### 4.1.1 Arquitetura

Tanto a rede Generativa quanto a Discriminatória são Transformers (VASWANI et al., 2017). Para possibilitar que os modelos trabalhassem com excertos mais longos dados os recursos computacionais disponíveis, foi aplicada uma versão do mecanismo de Atenção cujos custos em memória e computação crescem linearmente com o comprimento das sequências que recebe (KATHAROPOULOS et al., 2020). Cada modelo é composto por 6 blocos de Atenção.

A rede Geradora cumpre dois papéis: Em primeiro lugar, ela fica encarregada de prever cada item de cada sequência do *dataset* real com base nos elementos anteriores. Para tal, aplica-se uma máscara que impossibilita que o modelo acesse os elementos futuros da sequência, uma técnica comumente utilizada no treinamento de modelos de linguagem (BROWN et al., 2020). O segundo objetivo da rede é o de gerar sequências que se assemelhem àquelas vindas do conjunto real, de forma a enganar a rede Discriminatória. Essas sequências são geradas passo a passo de maneira autorregressiva, e para que tal processo fosse possível, foi utilizada a reformulação em termos de RNNs do mecanismo de Atenção causal, proposta em (KATHAROPOULOS et al., 2020). Este processo foi detalhado na seção 1.2.

Para realizar o condicionamento da rede, as camadas de normalização Layer-Norm (BA; KIROS; HINTON, 2016) presentes nos modelos Transformers foram substituídas por uma versão condicional, onde cada classe do *dataset* fica associada a um conjunto de parâmetros de escala e viés específicos. Nesse caso, se $i$, $k$ e $c$ representam, respectivamente, a posição de um elemento na sequência, a posição ao longo do vetor de *features*, e a classe de condicionamento, obtém-se:

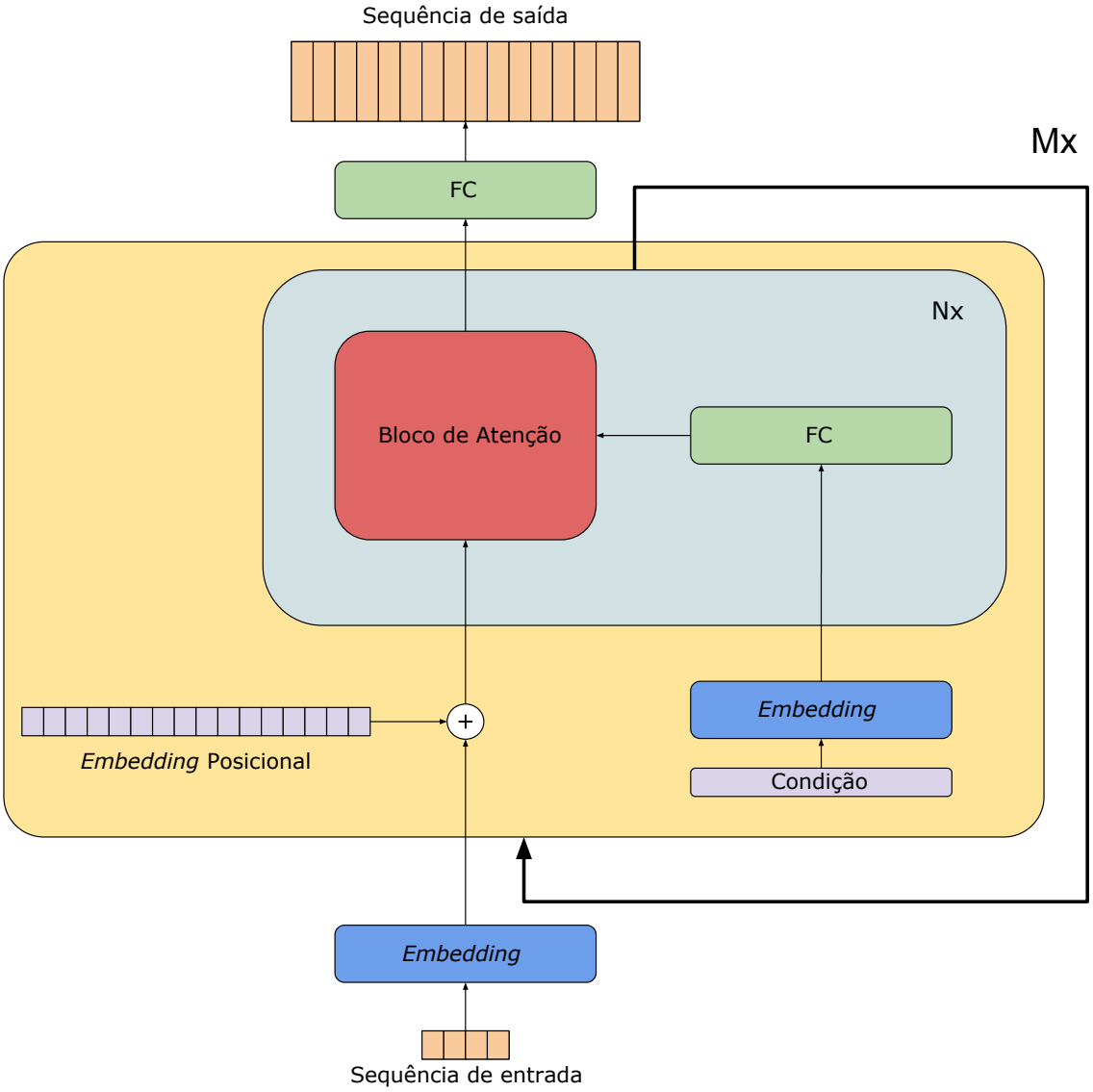
$$s'_{i,k} = \gamma^c_k \cdot s_{i,k} + \beta^c_k, \tag{4.1}$$

onde $s$ e $s'$ são as sequências de *input* e *output* da camada, e $\gamma$ e $\beta$ são os parâmetros de escala e viés. A estrutura de rede Generativa está dada na Figura 13.

A rede Discriminatória, por sua vez, observa cada sequência por inteiro e tenta determinar se ela é real ou falsa. Para reduzir custos computacionais e aumentar a eficiência do algoritmo, obteve-se inspiração da rede Vision Transformer (DOSOVITSKIY et al., 2021), que alcança resultados competitivos na classificação de imagens. Cada sequência recebida pela rede é separada em subsequências de tamanho pré-estabelecido, que em seguida são comprimidas para vetores de *feature* únicos que devem capturar as informações essenciais sobre o material original.

A rede Discriminatória produz dois *outputs*. Em primeiro lugar, antes da

Figura 13 – Rede Generativa Transformer.




Fonte: (NEVES, 2021)

passagem da sequência pela rede, um vetor [CLS] é concatenado a seu início. Esse vetor é comumente utilizado por modelos Transformers como um token de classificação (DEVLIN et al., 2019), e aqui ele cumpre o papel de armazenar informação global sobre o conteúdo visto, além de incorporar a informação condicional. O condicionamento é feito através do método de projeção (MIYATO; KOYAMA, 2018), no qual é realizado o produto interno entre um *embedding* da informação condicional e o resultado da passagem do token [CLS] pelo modelo.

O outro *output* do modelo é um mapa construído a partir da sequência de vetores de *feature*, onde cada elemento informa se a subsequência da qual foi extraída é real ou falsa. Uma técnica semelhante é geralmente aplicada no treinamento de GANs que operam em imagens (ISOLA et al., 2017) com o intuito de fazer com que o modelo priorize estrutura local. A Figura 14 ilustra o modelo Discriminatório.

O uso desses dois *outputs* é informado por conhecimento prévio sobre a estrutura musical. Várias escalas temporais devem ser consideradas para que se obtenha o entendimento completo de uma obra. Desde motivo e frase até a totalidade da peça, é necessário que a interação entre os vários elementos seja modelada. O mapa de *features* locais serve para garantir que ao menos no contexto mais basal, os padrões exibidos são coerentes. Certamente, mais estruturas além da local e global poderiam ser consideradas. No entanto, dados os recursos computacionais disponíveis e por razões de simplicidade, escolheu-se trabalhar apenas com essas duas.

### 4.1.2 Datasets

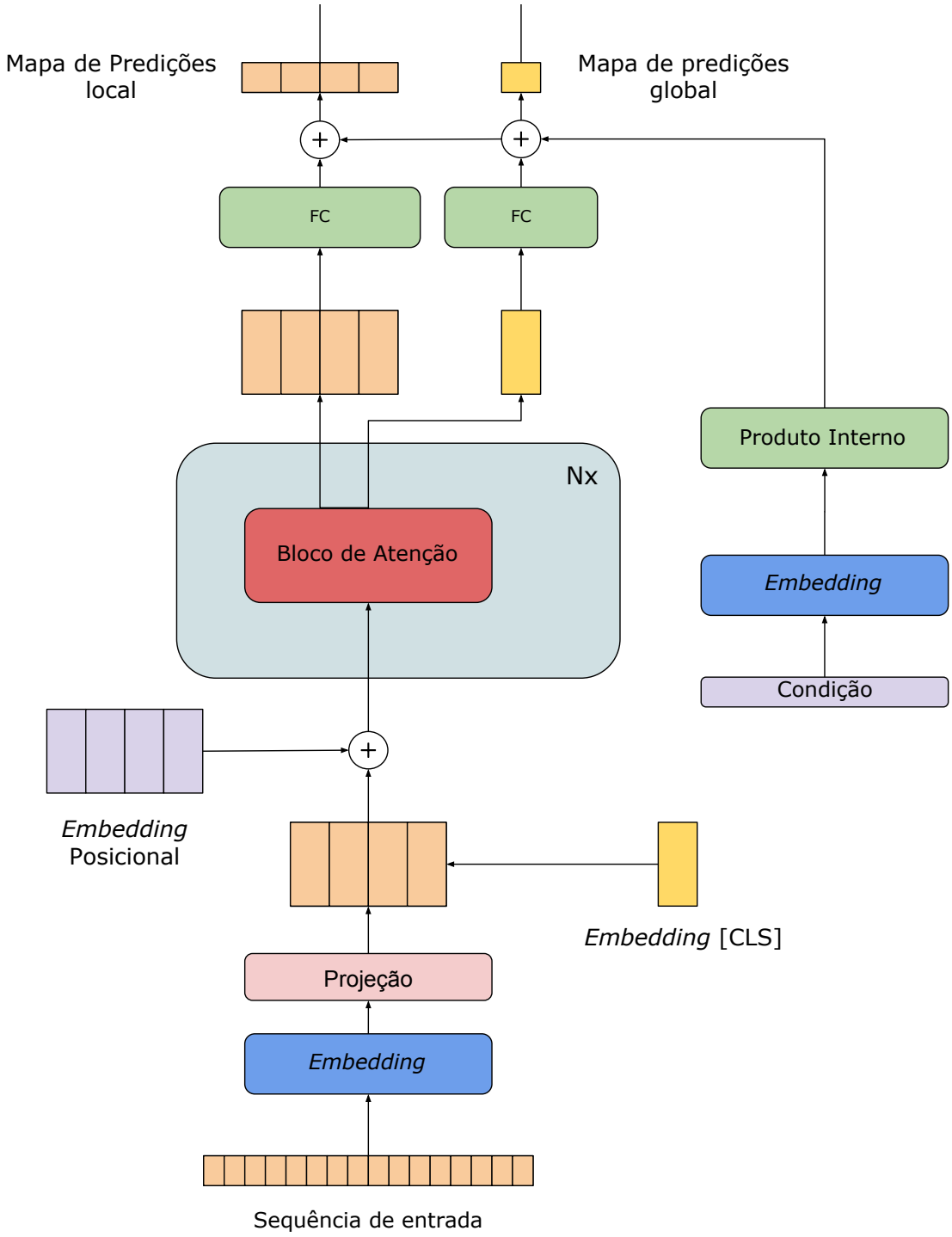
Para o treinamento do modelo, foram utilizadas duas bases de dados. O *dataset* AILABS17k (HSIAO et al., 2021) é constituído por adaptações de músicas populares para o piano, e então transformadas em MIDI pelo modelo de transcrição *Onsets and Frames* (HAWTHORNE et al., 2018). O *dataset* EMOPIA foi construído de forma semelhante. Porém, os excertos musicais presentes na base são separados em classes de acordo com valores de *Arousal* e *Valence*. Ambas as características podem ser classificadas como alta ou baixa, formando, assim, 4 classes ao todo.

O uso de dois *datasets* se faz necessário pois a base contendo valores anotados é relativamente pequena ($\sim$ 11h). Sendo assim, o modelo constrói uma representação geral da música em um conjunto de dados maior ($\sim$ 108h) e não condicionado, e aprende a associar o conteúdo musical às classes de estados afetivos graças ao conjunto menor.

### 4.1.3 Treinamento

A representação utilizada é a *revamped MIDI events* (REMI) (HUANG; YANG, 2020). Antes da introdução da rede Discriminatória, a rede Geradora foi treinada até a

Figura 14 – Rede Discriminatória Transformer.




Fonte: (NEVES, 2021)

otimalidade através do método de *teacher forcing*, que geralmente é aplicado no treinamento de modelos de linguagem. Esse pré-treino garantiu a estabilidade do estágio adversarial seguinte (ZHANG, 2020; MUHAMED et al., 2021). Neste estágio, os modelos são treinados em ambas as bases de dados simultaneamente, e alternou-se entre passos de otimização nas duas bases de forma a equilibrar o efeito de cada uma, levando em consideração a diferença de tamanho. A rede trabalha com sequências de 2048 elementos, correspondendo, em média, a 1 minuto de conteúdo.

Para o treinamento adversarial, foram utilizadas sequências de tamanho 128. Para diminuir a variância de gradiente inerente ao processo de amostragem, sequências vindas do conjunto real foram fornecidas para a rede Geradora de forma que esta tivesse um ponto de partida para a geração de novas amostras (ZHANG, 2020). A rede Discriminatória trabalhou com subsequências de tamanho 16, e o treinamento foi feito apenas na base de dados EMOPIA (HUNG et al., 2021).

As redes foram otimizadas através de uma combinação entre o objetivo MLE e o objetivo RSGAN (JOLICOEUR-MARTINEAU, 2019), e a penalidade de gradiente centrada em 1 (GULRAJANI et al., 2017) foi aplicada para garantir a estabilidade do treino. Uma taxa de aprendizado de $1 \cdot 10^{-4}$ foi usada para ambas as redes. A função *loss* final da rede Geradora é:

$$\mathcal{L}_G = \mathcal{L}_{\text{MLE}} + \alpha\mathcal{L}_{\text{RSGAN}_G\text{-global}} + \beta\mathcal{L}_{\text{RSGAN}_G\text{-local}}, \tag{4.2}$$

onde $\alpha$ e $\beta$ são hiperparâmetros que controlam a intensidade relativa de cada fator da função *loss*, $\mathcal{L}_{\text{MLE}}$ é o objetivo de *teacher forcing*, $\mathcal{L}_{\text{RSGAN}_G\text{-global}}$ é o objetivo adversarial global, e $\mathcal{L}_{\text{RSGAN}_G\text{-local}}$ é o objetivo adversarial local.

Já para a rede Discriminatória, o objetivo final é:

$$\mathcal{L}_D = \mathcal{L}_{\text{RSGAN}_D\text{-global}} + \alpha\mathcal{L}_{\text{RSGAN}_D\text{-local}} + \lambda(\mathcal{L}_{\text{GP-global}} + \mathcal{L}_{\text{GP-local}}), \tag{4.3}$$

onde $\mathcal{L}_{\text{RSGAN}_D\text{-global}}$ é o objetivo adversarial global, $\mathcal{L}_{\text{RSGAN}_D\text{-local}}$ é o objetivo adversarial local, e $\mathcal{L}_{\text{GP-global}}$ e $\mathcal{L}_{\text{GP-local}}$ são as penalidades de gradiente global e local, respectivamente. O Algoritmo 4 detalha o processo de otimização. A tabela 8 detala os hiperparâmetros

usados para o treinamento das redes.

**Algoritmo 4** – Transformer-GAN. A rede Discriminatória, $D$, tem parâmetros $w$, enquanto a Geradora, $G$, tem parâmetros $\theta$.

**Input:** $\alpha_g$, a taxa de aprendizado da rede Geradora, $\alpha_d$, a taxa de aprendizado da rede Discriminatória. $m$, o tamanho de *minibatch*. $n_{disc}$, o número de iterações da rede Discriminatória por iteração da Geradora. $n_{\text{pt}}$ o número de iterações de pré-treino. $n_{\text{total}}$, o número total de passos. Opt é o otimizador.

**enquanto** $\theta$ *não convergir* **faça**

Amostre $\{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$ do *dataset* real.

$$g_\theta \leftarrow \nabla_\theta \left[ -\frac{1}{m} \sum_{i=1}^{m} \log G_\theta(\boldsymbol{x}^{(i)}) \right]$$

$\theta \leftarrow \theta - \alpha_g \cdot \text{Opt}_G(\theta, g_\theta)$

**fim**

**enquanto** $\theta$ *não convergir* **faça**

**para** $j$ *in* $0, ..., n_{disc}$ **faça**

Amostre $\{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$ do *dataset* real.

Amostre $\{\epsilon\}_{i=1}^{m}$ de U$[0, 1]$

$\tilde{\boldsymbol{x}}^{(i)} \leftarrow G_\theta(\boldsymbol{x}^{(i)})$

$\hat{\boldsymbol{x}}^{(i)} \leftarrow \epsilon^{(i)}\boldsymbol{x}^{(i)} + (1 - \epsilon^{(i)})\tilde{\boldsymbol{x}}^{(i)}$

$$\begin{aligned} g_w \leftarrow \nabla_w \Bigg[ & -\frac{1}{m} \sum_{i=1}^{m} \log(\text{sigmoid}(D_w^{\text{glo}}(\boldsymbol{x}^{(i)}) - D_w^{\text{glo}}(G_\theta(\boldsymbol{x}^{(i)})))) \\ & -\frac{\alpha}{m} \sum_{i=1}^{m} \log(\text{sigmoid}(D_w^{\text{loc}}(\boldsymbol{x}^{(i)}) - D_w^{\text{loc}}(G_\theta(\boldsymbol{x}^{(i)})))) \\ & +\frac{\lambda}{m} \sum_{i=1}^{m} ((\|\nabla_{\hat{\boldsymbol{x}}} D_w^{\text{glo}}(\hat{\boldsymbol{x}}^{(i)})\| - 1)^2 + (\|\nabla_{\hat{\boldsymbol{x}}} D_w^{\text{loc}}(\hat{\boldsymbol{x}}^{(i)})\| - 1)^2) \Bigg] \end{aligned}$$

$w \leftarrow w - \alpha_d \cdot \text{Opt}_D(w, g_w)$

**fim**

Amostre $\{\boldsymbol{x}^{(i)}\}_{i=1}^{m}$ do *dataset* real.

$$\begin{aligned} g_\theta \leftarrow \nabla_\theta \Bigg[ & -\frac{1}{m} \sum_{i=1}^{m} \log G_\theta(\boldsymbol{x}^{(i)}) \\ & -\frac{\alpha}{m} \sum_{i=1}^{m} \log(\text{sigmoid}(D_w^{\text{glo}}(G_\theta(\boldsymbol{x}^{(i)}) - D_w^{\text{glo}}(\boldsymbol{x}^{(i)})))) \\ & -\frac{\beta}{m} \sum_{i=1}^{m} \log(\text{sigmoid}(D_w^{\text{loc}}(G_\theta(\boldsymbol{x}^{(i)}) - D_w^{\text{loc}}(\boldsymbol{x}^{(i)}))))\Bigg] \end{aligned}$$

$\theta \leftarrow \theta - \alpha_g \cdot \text{Opt}_G(\theta, g_\theta)$

**fim**

Tabela 8 – Hiperparâmetros utilizados durante o treinamento.

| Nome | Valor |
|---|---|
| Tamanho de *batch* | 8 (pré-treino), 64 (treino) |
| Tamanho das sequências | 2048 (pré-treino), 128 (treino) |
| *Loss* | RSGAN Loss (LIM; YE, 2017) |
| Atualizações de **D** por atualizações de **G** | 1 |
| Otimizador | **G** - Adam(lr=$1e-4$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) |
| Iterações | 26000 (pré-treino), 26000 (treino) |
| $\lambda$ (WGAN-GP) (GULRAJANI et al., 2017) | 10 |
| $\alpha$ e $\beta$ (hiperparâmetros de escala entre *losses* local e global) | 1 |

Fonte: (NEVES, 2021)

## 4.2 Experimentos

Primeiramente, os modelos foram avaliados através de métricas automáticas que quantificam atributos musicalmente relevantes das amostras geradas e sua capacidade de comunicar os estados afetivos desejados. O modelo proposto foi comparado com um modelo equivalente treinado sem o sinal adversário, e com o modelo estado-da-arte baseado em Transformers (HUNG et al., 2021). Especificamente, foram utilizadas as métricas de Distância Tonal (distância entre a nota mais grave e mais aguda), Número de Classes Tonais, e Polifonia (número médio de notas tocadas simultaneamente). As métricas foram calculadas por meio da biblioteca Muspy (DONG et al., 2020). Para cada modelo avaliado, geraram-se 400 amostras (100 para cada classe), e essas foram avaliadas de acordo com as características mencionadas acima. Então, tomou-se a média dos resultados de cada modelo. Para as métricas automáticas utilizadas, quanto mais próximos os valores das distribuições sintéticas estão daqueles correspondentes às distribuições reais, melhor é a performance do modelo, pois na tarefa de geração automática, deseja-se aproximar sinteticamente a distribuição de dados reais. Os resultados estão dispostos na Tabela 9.

Tabela 9 – Comparação, em termos das métricas automáticas, entre as amostras geradas. DT é Distância Tonal, NCT abrevia Número de Classes Tonais e POLI significa Polifonia. Os valores em negrito indicam as melhores performances.

| | DT | NCT | POLI |
|---|---|---|---|
| Dados Reais (EMOPIA)(HUNG et al., 2021) | 50.94 | 8.50 | 5.60 |
| Baseline (HUNG et al., 2021) | 49.76 | **8.52** | 4.36 |
| Transformer | 48.79 | 8.65 | 4.37 |
| Transformer GAN | **50.73** | 9.45 | **4.43** |

Fonte: (NEVES, 2021)

Como é possível observar, o modelo proposto obtém uma performance melhor do que o *baseline* e do que o modelo pré-treinado em duas das três métricas automáticas utilizadas. Pode-se ressaltar ainda o fato de que o Gerador é cerca de 2 vezes menor do que o *baseline*, tendo apenas 6 camadas de Atenção, enquanto o segundo possui 12. Por último, cabe mencionar que a representação utilizada no treinamento do *baseline* é comprovadamente superior àquela usada pelo modelo proposto (HSIAO et al., 2021), sendo também consideravelmente mais complexa. Ainda assim, o modelo proposto obtém uma performance superior. Tendo isso em mente, a adaptação desta representação para o contexto adversarial poderia ocasionar em uma performance ainda melhor.

Finalmente, foi realizada uma pesquisa onde participantes avaliaram características musicalmente relevantes do material sintetizado, bem como seu conteúdo afetivo. A pesquisa foi realizada virtualmente, contando com um total de 18 participantes recrutados por meio das redes sociais, e portanto advindos do público geral. Cada um desses participantes teve que classificar as amostras ouvidas em uma escala Likert de 5 pontos, indo de muito baixo a muito alto, de acordo com as características abaixo:

- Humanidade - o quanto a amostra parece ter sido composta por um humano, ou o quanto ela demonstra características de peças musicais humanas.
- Originalidade - se a amostra exibe padrões musicais interessantes e inovadores.
- Estrutura - se a música apresenta uma estrutura bem definida, com padrões recorrentes ou desenvolvimento de ideias.
- Qualidade Geral - A impressão geral do participante sobre a música.
- *Arousal* - Se a amostra comunica um estado emocional pouco intenso (como tédio ou calma) ou muito intenso (como surpresa ou medo).
- *Valence* - Se a amostra comunica um estado emocional desprazeroso (como tristeza ou raiva) ou prazeroso (como alegria e satisfação).

Cada participante teve que ouvir a doze excertos musicais no total - quatro para cada modelo e, dentre esses quatro, um para cada combinação de *Valence* e *Arousal*. Antes do teste, um texto explicando os conceitos fundamentais para a realização da pesquisa foi apresentado. A página *web* correspondente ao questionário está disponível na seção de Apêndices.

As pontuações médias para a GAN, o Transformer convencional e o estado-da-arte para as características de Humanidade, Originalidade, Estrutura, Qualidade Geral, assim como a Acurácia de Classificação, estão dadas na Tabela 10. Neste caso, quanto mais alto o valor alcançado, melhor é a performance do modelo.

Tabela 10 – Resultado da pesquisa onde participantes classificaram amostras produzidas por diversos modelos. As colunas são, respectivamente, Humanidade, Originalidade, Estrutura, Qualidade Geral.

| | H | O | E | QG |
|---|---|---|---|---|
| Baseline (HUNG et al., 2021) | 3.32 ± 1.29 | 2.93 ± 1.13 | 3.18 ± 1.30 | 3.49 ± 1.04 |
| Transformer | 3.75 ± 1.24 | 3.22 ± 1.19 | 3.76 ± 1.14 | **3**.89 ± 1.14 |
| Transformer-GAN | 3.56 ± 1.34 | 3.06 ± 1.21 | 3.38 ± 1.09 | 3.44 ± 1.15 |

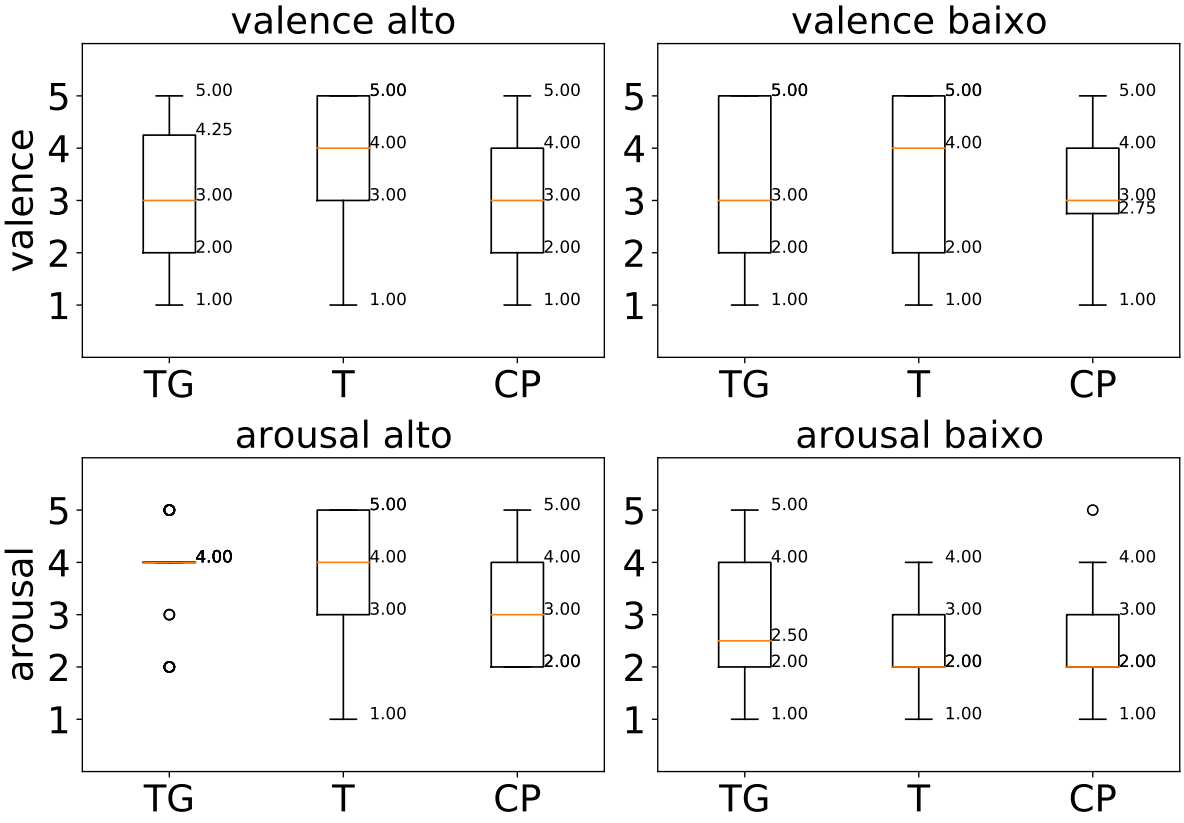
Fonte: (NEVES, 2021)

Como observa-se pelas pontuações e pelos erros, tanto o Transformer quanto a Transformer-GAN obtêm performances competitivas com o *baseline* nas métricas. Ainda que em um primeiro momento esses resultados possam sugerir que a GAN é equivalente ao Transformer padrão, deve-se ressaltar que a GAN obtém um desempenho melhor do que o *baseline* proposto em (HUNG et al., 2021) em termos das métricas automáticas, enquanto o Transformer fica atrás do *baseline* nas métricas automáticas.

As classificações de *Valence* e *Arousal* dadas pelos participantes foram comparadas com os rótulos reais usados durante a geração. A Figura 15 ilustra os resultados desse experimento.

Como observa-se pelos gráficos, o Transformer-GAN obtém um desempenho competitivo com o estado da arte e também com o Transformer básico. De fato, as amostras produzidas pelo modelo apresentam medianas que se aproximam mais das regiões das quais as amostras de treino são retiradas, principalmente no quesito *arousal*. Assim, fica evidenciado o potencial da rede Discriminatória no condicionamento das amostras produzidas pela rede Geradora.

Figura 15 – Resultado da pesquisa na qual participantes classificaram as amostras em termos de suas percepções sobre *valence* e *arousal*. As siglas TG, T e CP correspondem, respectivamente, aos modelos Transformer GAN, Transformer Tradicional e Baseline.




Fonte: (NEVES, 2021)

# 5 Considerações Finais

Ao longo deste projeto, foram propostos e estudados três modelos de aprendizado de máquinas baseados em Redes Generativas Adversárias e voltados para a geração musical. O objetivo principal do projeto era o desenvolvimento de uma rede neural capaz de sintetizar passagens musicais condicionadas por valores de *valence* e *arousal*, que expressam quantitativamente as emoções humanas. No entanto, ao longo do caminho, surgiram perspectivas que viabilizaram a conclusão de outros objetivos, que acabaram por impactar positivamente o desenvolvimento do modelo final.

O primeiro dos sistemas, que conta com uma camada de auto-atenção responsável pelo cálculo de relações globais entre os elementos dos tensores que recebe, foi construído para a geração de trechos musicais curtos condicionados por instrumento musical. Especificamente, é possível fornecer um código representando um entre 8 instrumentos musicais disponíveis. Através das métricas Fréchet Inception Distange (FID) (HEUSEL et al., 2017) e Inception Score (IS) (SALIMANS et al., 2016), já muito utilizadas para a avaliação de GANs, mostrou-se que ele obtém um desempenho superior àquele que não faz o uso do mecanismo de auto-atenção.

O segundo modelo, que assim como o último atua diretamente em áudio digital, realiza a tarefa de reconstruir passagens muscais a partir de representações comprimidas aprendidas dinamicamente. Baseado em uma combinação do *Vector Quantized Variational Autoencoder* (VQVAE) com uma GAN, e treinado em uma base de dados constituída por peças clássicas de piano performadas por virtuosos, ele diminui a extensão das passagens sonoras em 128x, representando-as na forma de códigos discretos que compõem um vocabulário de tamanho fixo. Os elementos desse vocabulário podem então ser usados no treinamento de uma rede que produz novas amostras de forma autorregressiva, reduzindo os custos computacionais envolvidos na geração musical. O modelo foi comparado com um semelhante que representa o estado-na-arte no campo, denominado Jukebox (DHARIWAL et al., 2020). A comparação foi feita por meio da métrica automática Fréchet Audio Distance (FAD) (KILGOUR et al., 2019), que é especialmente voltada para a avaliação de redes que lidam com áudio. O VQGAN obteve uma performance competitiva com o estado-da-arte, tendo também as vantagens de necessitar de um único conjunto de códigos e de permitir um alto nível de compressão.

O terceiro sistema, que é o principal modelo desenvolvido durante a pesquisa, trabalha com representações musicais simbólicas e sintetiza passagens inteiras condicionadas por estados emocionais. Também sendo uma GAN, ele escala a arquitetura Transformer (VASWANI et al., 2017) tanto no papel de rede Geradora quanto de Discriminatória. Dessa

vez, além de métricas automáticas que computam características musicalmente relevantes, o modelo foi avaliado também por ouvintes humanos. Para ambos os tipos de avaliação, obteve-se uma performance competitiva com o estado-da-arte (HUNG et al., 2021), que faz o uso de uma representação comprovadamente melhor e que vale-se de uma rede geradora com mais parâmetros do que a proposta. Além disso, os mesmos resultados apontam que o modelo é também competitivo na tarefa de geração condicionada por classes emocionais, isto é, no objetivo primário do projeto.

À frente das considerações acima, algumas conclusões podem ser tomadas. Em primeiro lugar, fica claro que a utilização de Redes Generativas Adversárias no contexto de geração musical automatizada fomenta um grande ganho de performance, especialmente no contexto de geração condicionada. Ainda que tenha sido dispensada uma quantidade menor de esforços de pesquisa para a aplicação das GANs nos domínios de áudio e de dados discretos - e especificamente música em formato simbólico - e mesmo perante as dificuldades envolvidas na adaptação do treinamento adversário para estes domínios, os resultados discutidos ao longo deste trabalho evidenciam o grande potencial dessa estrutura.

No entanto, o projeto também deixa claras as limitações atuais das GANs e as dificuldades envolvidas na geração musical tanto em forma de áudio quanto no formato simbólico. Como explicitado ao longo do texto, a grande densidade de amostras que compõem o áudio digital fazem com que este seja difícil de modelar. Essa realidade, quando combinada com a complexidade inerente ao esquema adversarial que caracteriza as GANs, e também com os recursos computacionais demandados pelo algoritmo de treino, torna o processo extremamente instável, custoso e demorado. De fato, estes desafios tornaram impossível a conclusão dos experimentos relacionados ao segundo modelo apresentado. O estágio final de treino, que deveria consistir no desenvolvimento de uma rede capaz de modelar os códigos gerados pelo *Autoencoder*, requisitava de poder computacional que não estava disponível ao longo do trabalho.

Em termos de música representada de maneira discreta, os desafios são outros. Aqui, a natureza dos dados, não apropriada para o treinamento das Redes Adversárias por dificultar o fluxo do sinal de treino através das redes, faz com que seja necessária a busca por ferramentas matemáticas que viabilizem a execução dos algoritmos. As soluções disponíveis atualmente, ainda que em muitos casos funcionem, adicionam uma complexidade ainda maior ao processo, incluindo hiperparâmetros adicionais ou a realização de mais cálculos ao longo das iterações, o que muda a dinâmica de treino e incorre em custos significativamente maiores. Além disso, devem-se mencionar também as dificuldades intrínsecas ao processo de transformação do conteúdo musical em material simbólico. Muitas nuances da música (como articulação, dinâmica, e rubato) podem ser perdidas neste processo, e ele potencialmente diminui a quantidade de dados dos quais pode-se fazer o uso, já que a maior parte das canções existentes só está disponível em formato de áudio.

Mesmo que existam modelos cuja função é transcrever áudio para MIDI, atualmente ainda é necessário um de tais modelos de transcrição para cada instrumento com o qual se deseja trabalhar, e a transcrição de faixas polifônicas é um projeto que aparenta estar distante de ser concluído satisfatoriamente.

O trabalho também deixa algumas questões em aberto. Primeiramente, como os sinais condicionais na forma de rótulos emocionais foram usados apenas no treinamento de modelos que operam em representações musicais simbólicas - para que fosse possível utilizá-los no domínio do áudio, seria necessária uma quantidade consideravelmente maior de poder computacional - ainda é necessário estudar como poderiam ser desenvolvidos sistemas análogos que operam em áudio, e quais resultados estão ao alcance do estado-da-arte atual.

Um outro objetivo que pode ser buscado é o de transformar os modelos apresentados em ferramentas ou produtos aplicáveis em situações reais. Algoritmos capazes de gerar músicas que projetam estados emocionais específicos têm o potencial de ser usados em contextos terapêuticos, como no tratamento de vítimas de derrames ou para o incentivo da comunicação entre crianças com autismo. assim como na criação automática de trilhas sonoras para jogos eletrônicos procedurais ou para formatos artísticos baseados em narrativa. Além disso, as múcicas sintetizadas por esses sistemas, a princípio, podem ser usadas livremente, por não possuírem direitos autorais.

Em se tratando dos sistemas que trabalham com áudio, acredita-se que a trajetória de pesquisa com o maior potencial de gerar mais frutos positivos para o campo é a da criação de modelos mais eficientes e leves que tenham a habilidade de desempenhar as tarefas examinadas aqui, até mesmo porque caso esses sistemas venham a ser utilizados pelo público geral, uma das questões mais importantes a se estudar é a de como fazer com que eles sejam suportados pelo maior número de máquinas possível.

Para todas as redes, dado o sucesso alcançado na implementação de sinais condicionais responsáveis por guiar o processo de geração, abre-se a discussão sobre quais outros tipos de dados poderiam ser usados no desenvolvimento das GANs, ou de quais outras formas esses dados poderiam ser aplicados. Algumas possibilidades incluem produzir canções que pertençam a estilos musicais pré-definidos ou que incluam um conjunto específico de instrumentos, permitir um controle maior sobre as emoções representadas por meio do uso de dados contínuos de *valence* e *arousal*, assim como investigar a aplicação dessas informações condicionais, especialmente as que representam estados emocionais, em outros domínios de áudio, como paisagens sonoras ou voz. Em suma, o trabalho nos deixa com a seguintes perspectivas: Redes Generativas Adversárias podem ser aplicadas bem-sucedidamente nas tarefas de geração musical condicionada tanto no domínio do áudio quanto no domínio simbólico. De fato, o objetivo principal da pesquisa, isto é, a geração musical condicionada por estados emocionais, foi concluído com êxito, assim como os outros objetivos intermediários estudados no decorrer do projeto. Cada um dos domínios

estudados - áudio e simbólico - apresenta desafios específicos e muito distintos daqueles que caracterizam o outro, tendo sido necessária a aplicação de um arsenal de técnicas e ferramentas matemáticas especializadas para cada um. Por fim, ficou claro que ainda há muito o que ser feito em termos da otimização das GANs no contexto de geração musical, e que há bastante espaço para pesquisas que busquem novas formas de utilizar esse algoritmo tanto para a melhora dos resultados até então alcançados como para a exploração de outros caminhos na área de MIR.

# Referências

ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. Wasserstein generative adversarial networks. In: PMLR. *International conference on machine learning*. [S.l.], 2017. p. 214–223. Citado 2 vezes nas páginas 23 e 24.

ASSAYAG, G.; RUEDA, C.; LAURSON, M.; AGON, C.; DELERUE, O. Computer-assisted composition at ircam: From patchwork to openmusic. *Computer Music Journal*, v. 23, n. 3, p. 59–72, 1999. Disponível em: <https://doi.org/10.1162/014892699559896>. Citado 2 vezes nas páginas 16 e 18.

BA, J. L.; KIROS, J. R.; HINTON, G. E. *Layer Normalization*. arXiv, 2016. Disponível em: <https://arxiv.org/abs/1607.06450>. Citado na página 62.

BARRATT, S.; SHARMA, R. *A Note on the Inception Score*. arXiv, 2018. Disponível em: <https://arxiv.org/abs/1801.01973>. Citado 3 vezes nas páginas 26, 27 e 50.

BINKOWSKI, M.; DONAHUE, J.; DIELEMAN, S.; CLARK, A.; ELSEN, E.; CASAGRANDE, N.; COBO, L. C.; SIMONYAN, K. High fidelity speech synthesis with adversarial networks. *CoRR*, abs/1909.11646, 2019. Disponível em: <http://arxiv.org/abs/1909.11646>. Citado 2 vezes nas páginas 38 e 54.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. New York: Springer, 2006. Citado na página 16.

BORJI, A. Pros and cons of GAN evaluation measures. *CoRR*, abs/1802.03446, 2018. Disponível em: <http://arxiv.org/abs/1802.03446>. Citado 2 vezes nas páginas 50 e 51.

BRIOT, J.-P.; HADJERES, G.; PACHET, F.-D. *Deep Learning Techniques for Music Generation – A Survey*. [s.n.], 2019. Disponível em: <https://hal.sorbonne-universite.fr/hal-01660772>. Citado 2 vezes nas páginas 17 e 38.

BROCK, A.; DONAHUE, J.; SIMONYAN, K. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018. Disponível em: <http://arxiv.org/abs/1809.11096>. Citado 2 vezes nas páginas 17 e 26.

BROWN, T. B.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A. et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. Citado na página 62.

CHILD, R.; GRAY, S.; RADFORD, A.; SUTSKEVER, I. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. Citado na página 31.

CHOROMANSKI, K. M.; LIKHOSHERSTOV, V.; DOHAN, D.; SONG, X.; GANE, A.; SARLOS, T.; HAWKINS, P.; DAVIS, J. Q.; MOHIUDDIN, A.; KAISER, L.; BELANGER, D. B.; COLWELL, L. J.; WELLER, A. Rethinking attention with performers. In: *International Conference on Learning Representations*. [s.n.], 2021. Disponível em: <https://openreview.net/forum?id=Ua6zuk0WRH>. Citado na página 31.

COPE, D. Experiments in musical intelligence (emi): Non-linear linguistic-based composition. *Journal of New Music Research*, Taylor & Francis, v. 18, n. 1-2, p. 117–139, 1989. Citado na página 36.

CORDONNIER, J.; LOUKAS, A.; JAGGI, M. On the relationship between self-attention and convolutional layers. *CoRR*, abs/1911.03584, 2019. Disponível em: <http://arxiv.org/abs/1911.03584>. Citado na página 44.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*. [S.l.: s.n.], 2009. Citado na página 50.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: *NAACL*. [S.l.: s.n.], 2019. Citado na página 64.

DHARIWAL, P.; JUN, H.; PAYNE, C.; KIM, J. W.; RADFORD, A.; SUTSKEVER, I. *Jukebox: A Generative Model for Music*. arXiv, 2020. Disponível em: <https://arxiv.org/abs/2005.00341>. Citado 6 vezes nas páginas 34, 37, 54, 55, 58 e 72.

DIELEMAN, S.; OORD, A. van den; SIMONYAN, K. The challenge of realistic music generation: modelling raw audio at scale. *CoRR*, abs/1806.10474, 2018. Disponível em: <http://arxiv.org/abs/1806.10474>. Citado 4 vezes nas páginas 17, 34, 37 e 58.

DONAHUE, C.; MCAULEY, J. J.; PUCKETTE, M. S. Synthesizing audio with generative adversarial networks. *CoRR*, abs/1802.04208, 2018. Disponível em: <http://arxiv.org/abs/1802.04208>. Citado na página 38.

DONG, H.-W.; CHEN, K.; MCAULEY, J.; BERG-KIRKPATRICK, T. Muspy: A toolkit for symbolic music generation. In: *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*. [S.l.: s.n.], 2020. Citado 2 vezes nas páginas 42 e 68.

DONG, H.-W.; HSIAO, W.-Y.; YANG, L.-C.; YANG, Y.-H. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. [S.l.: s.n.], 2018. Citado na página 41.

DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEHGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S.; USZKOREIT, J.; HOULSBY, N. An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations*. [s.n.], 2021. Disponível em: <https://openreview.net/forum?id=YicbFdNTTy>. Citado na página 62.

EDWARDS, M. Algorithmic composition: computational thinking in music. *Communications of the ACM*, ACM New York, NY, USA, v. 54, n. 7, p. 58–67, 2011. Citado na página 36.

ENGEL, J.; AGRAWAL, K. K.; CHEN, S.; GULRAJANI, I.; DONAHUE, C.; ROBERTS, A. GANSynth: Adversarial neural audio synthesis. In: *International Conference on Learning Representations*. [s.n.], 2019. Disponível em: <https://openreview.net/forum?id=H1xQVn09FX>. Citado na página 38.

ENGEL, J.; RESNICK, C.; ROBERTS, A.; DIELEMAN, S.; NOROUZI, M.; ECK, D.; SIMONYAN, K. Neural audio synthesis of musical notes with wavenet autoencoders. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2017. p. 1068–1077. Citado na página 34.

ESSER, P.; ROMBACH, R.; OMMER, B. Taming transformers for high-resolution image synthesis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2021. p. 12873–12883. Citado na página 56.

FERNÁNDEZ, J. D.; VICO, F. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, v. 48, p. 513–582, 2013. Citado na página 36.

FERREIRA, L. N.; WHITEHEAD, J. Learning to generate music with sentiment. *Proceedings of the Conference of the International Society for Music Information Retrieval*, 2019. Citado na página 41.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 25.

GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial networks. *Advances in Neural Information Processing Systems*, v. 3, 06 2014. Citado 5 vezes nas páginas 16, 21, 22, 53 e 61.

Guan, F.; Yu, C.; Yang, S. A gan model with self-attention mechanism to generate multi-instruments symbolic music. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2019. p. 1–6. Citado na página 41.

GULRAJANI, I.; AHMED, F.; ARJOVSKY, M.; DUMOULIN, V.; COURVILLE, A. C. Improved training of wasserstein gans. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf>. Citado 3 vezes nas páginas 24, 66 e 68.

HAWTHORNE, C.; ELSEN, E.; SONG, J.; ROBERTS, A.; SIMON, I.; RAFFEL, C.; ENGEL, J.; OORE, S.; ECK, D. Onsets and frames: Dual-objective piano transcription. In: *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, 2018*. [s.n.], 2018. Disponível em: <https://arxiv.org/abs/1710.11153>. Citado 3 vezes nas páginas 37, 42 e 64.

HAWTHORNE, C.; STASYUK, A.; ROBERTS, A.; SIMON, I.; HUANG, C.-Z. A.; DIELEMAN, S.; ELSEN, E.; ENGEL, J.; ECK, D. Enabling factorized piano music modeling and generation with the MAESTRO dataset. In: *International Conference on Learning Representations*. [s.n.], 2019. Disponível em: <https://openreview.net/forum?id=r1lYRjC9F7>. Citado 3 vezes nas páginas 37, 42 e 58.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. Disponível em: <http://arxiv.org/abs/1512.03385>. Citado 3 vezes nas páginas 16, 38 e 53.

HERSHEY, S.; CHAUDHURI, S.; ELLIS, D. P. W.; GEMMEKE, J. F.; JANSEN, A.; MOORE, C.; PLAKAL, M.; PLATT, D.; SAUROUS, R. A.; SEYBOLD, B.; SLANEY, M.; WEISS, R.; WILSON, K. Cnn architectures for large-scale audio classification. In: *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [s.n.], 2017. Disponível em: <https://arxiv.org/abs/1609.09430>. Citado na página 59.

HEUSEL, M.; RAMSAUER, H.; UNTERTHINER, T.; NESSLER, B.; KLAMBAUER, G.; HOCHREITER, S. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017. Disponível em: <http://arxiv.org/abs/1706.08500>. Citado 8 vezes nas páginas 26, 27, 42, 47, 50, 51, 59 e 72.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado 2 vezes nas páginas 17 e 29.

HSIAO, W.; LIU, J.; YEH, Y.; YANG, Y. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. *CoRR*, abs/2101.02402, 2021. Disponível em: <https://arxiv.org/abs/2101.02402>. Citado 4 vezes nas páginas 41, 42, 64 e 69.

HUANG, C.-Z. A.; VASWANI, A.; USZKOREIT, J.; SIMON, I.; HAWTHORNE, C.; SHAZEER, N.; DAI, A. M.; HOFFMAN, M. D.; DINCULESCU, M.; ECK, D. Music transformer. In: *International Conference on Learning Representations*. [s.n.], 2019. Disponível em: <https://openreview.net/forum?id=rJe4ShAcF7>. Citado na página 40.

HUANG, Y.-S.; YANG, Y.-H. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In: *Proceedings of the 28th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2020. (MM '20), p. 1180–1188. ISBN 9781450379885. Disponível em: <https://doi.org/10.1145/3394171.3413671>. Citado 3 vezes nas páginas 40, 41 e 64.

HUNG, H.-T.; CHING, J.; DOH, S.; KIM, N.; NAM, J.; YANG, Y.-H. EMOPIA: A multi-modal pop piano dataset for emotion recognition and emotion-based music generation. In: *Proc. Int. Society for Music Information Retrieval Conf.* [S.l.: s.n.], 2021. Citado 5 vezes nas páginas 41, 66, 68, 70 e 73.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 448–456. Citado na página 45.

ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1125–1134. Citado na página 64.

JANG, E.; GU, S.; POOLE, B. Categorical reparameterization with gumbel-softmax. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. Disponível em: <https://openreview.net/forum?id=rkE3y85ee>. Citado 4 vezes nas páginas 17, 28, 41 e 61.

JOLICOEUR-MARTINEAU, A. The relativistic discriminator: a key element missing from standard GAN. In: *International Conference on Learning Representations*. [s.n.], 2019. Disponível em: <https://openreview.net/forum?id=S1erHoR5t7>. Citado 2 vezes nas páginas 25 e 66.

KARRAS, T.; LAINE, S.; AITTALA, M.; HELLSTEN, J.; LEHTINEN, J.; AILA, T. Analyzing and improving the image quality of stylegan. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 8110–8119. Citado 2 vezes nas páginas 17 e 26.

KATHAROPOULOS, A.; VYAS, A.; PAPPAS, N.; FLEURET, F. Transformers are rnns: Fast autoregressive transformers with linear attention. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2020. p. 5156–5165. Citado 3 vezes nas páginas 31, 32 e 62.

KILGOUR, K.; ZULUAGA, M.; ROBLEK, D.; SHARIFI, M. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In: *INTERSPEECH*. [S.l.: s.n.], 2019. p. 2350–2354. Citado 2 vezes nas páginas 59 e 72.

KIM, Y.; SCHMIDT, E.; MIGNECO, R.; MORTON, B.; RICHARDSON, P.; SCOTT, J.; SPECK, J.; TURNBULL, D. Music emotion recognition: A state of the art review. *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, 01 2010. Citado na página 61.

KINGMA, D.; WELLING, M. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, v. 12, p. 307–392, 01 2019. Citado na página 33.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: *ICLR (Poster)*. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1412.6980>. Citado 3 vezes nas páginas 47, 48 e 57.

KINGMA, D. P.; WELLING, M. Auto-Encoding Variational Bayes. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. [S.l.: s.n.], 2014. Citado na página 32.

KITAEV, N.; KAISER, L.; LEVSKAYA, A. Reformer: The efficient transformer. In: *International Conference on Learning Representations*. [s.n.], 2020. Disponível em: <https://openreview.net/forum?id=rkgNKkHtvB>. Citado na página 31.

KUMAR, K.; KUMAR, R.; BOISSIERE, T. de; GESTIN, L.; TEOH, W. Z.; SOTELO, J.; BREBISSON, A. de; BENGIO, Y.; COURVILLE, A. *MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis*. 2019. Citado 3 vezes nas páginas 38, 45 e 54.

KUSNER, M. J.; HERNáNDEZ-LOBATO, J. M. *GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution*. arXiv, 2016. Disponível em: <https://arxiv.org/abs/1611.04051>. Citado na página 28.

LIM, J. H.; YE, J. C. Geometric GAN. *arXiv preprint arXiv:1705.02894*, 2017. Citado 7 vezes nas páginas 25, 47, 51, 55, 56, 57 e 68.

LIU*, P. J.; SALEH*, M.; POT, E.; GOODRICH, B.; SEPASSI, R.; KAISER, L.; SHAZEER, N. Generating wikipedia by summarizing long sequences. In: *International Conference on Learning Representations*. [s.n.], 2018. Disponível em: <https://openreview.net/forum?id=Hyg0vbWC->. Citado na página 31.

LOSTANLEN, V.; CELLA, C.-E. Deep convolutional networks on the pitch spiral for music instrument recognition. In: *ISMIR*. [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 42 e 49.

MAKRIS, D.; AGRES, K. R.; HERREMANS, D. Generating lead sheets with affect: A novel conditional seq2seq framework. *arXiv preprint arXiv:2104.13056*, 2021. Citado na página 41.

Maurer IV, J. A. *A Brief History of Algorithmic Composition*. 1999. <https://ccrma.stanford.edu/~blackrse/algorithm.html#:~:text=Algorithmic%20composition%2C%20sometimes%20also%20referred,%22%20(Alpern%2C%201995).&text=Computers%20have%20given%20composers%20new%20opportunities%20to%20automate%20the%20compositional%20process.> Accessado em: 06-02-2022. Citado na página 36.

MEHRI, S.; KUMAR, K.; GULRAJANI, I.; KUMAR, R.; JAIN, S.; SOTELO, J.; COURVILLE, A.; BENGIO, Y. *SampleRNN: An Unconditional End-to-End Neural Audio Generation Model*. arXiv, 2016. Disponível em: <https://arxiv.org/abs/1612.07837>. Citado 2 vezes nas páginas 32 e 37.

MIRZA, M.; OSINDERO, S. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. Disponível em: <http://arxiv.org/abs/1411.1784>. Citado na página 22.

MIYATO, T.; KATAOKA, T.; KOYAMA, M.; YOSHIDA, Y. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018. Disponível em: <http://arxiv.org/abs/1802.05957>. Citado 2 vezes nas páginas 25 e 47.

MIYATO, T.; KOYAMA, M. cGANs with projection discriminator. In: *International Conference on Learning Representations*. [s.n.], 2018. Disponível em: <https://openreview.net/forum?id=ByS1VpgRZ>. Citado 2 vezes nas páginas 45 e 64.

MUHAMED, A.; LI, L.; SHI, X.; YADDANAPUDI, S.; CHI, W.; JACKSON, D.; SURESH, R.; LIPTON, Z. C.; SMOLA, A. J. Symbolic music generation with transformer-gans. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2021. v. 35, p. 408–417. Citado 2 vezes nas páginas 41 e 66.

NIERHAUS, G. *Algorithmic composition: paradigms of automated music generation*. [S.l.]: Springer Science & Business Media, 2009. Citado na página 16.

OORD, A. van den; DIELEMAN, S.; ZEN, H.; SIMONYAN, K.; VINYALS, O.; GRAVES, A.; KALCHBRENNER, N.; SENIOR, A. W.; KAVUKCUOGLU, K. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016. Disponível em: <http://arxiv.org/abs/1609.03499>. Citado 4 vezes nas páginas 17, 32, 33 e 37.

OORD, A. van den; LI, Y.; BABUSCHKIN, I.; SIMONYAN, K.; VINYALS, O.; KAVUKCUOGLU, K.; DRIESSCHE, G. van den; LOCKHART, E.; COBO, L. C.; STIMBERG, F.; CASAGRANDE, N.; GREWE, D.; NOURY, S.; DIELEMAN, S.; ELSEN, E.; KALCHBRENNER, N.; ZEN, H.; GRAVES, A.; KING, H.; WALTERS, T.; BELOV, D.; HASSABIS, D. Parallel wavenet: Fast high-fidelity speech synthesis. *CoRR*, abs/1711.10433, 2017. Disponível em: <http://arxiv.org/abs/1711.10433>. Citado na página 32.

OORD, A. van den; VINYALS, O.; KAVUKCUOGLU, K. Neural discrete representation learning. *CoRR*, abs/1711.00937, 2017. Disponível em: <http://arxiv.org/abs/1711.00937>. Citado 6 vezes nas páginas 33, 34, 37, 53, 55 e 56.

OORE, S.; SIMON, I.; DIELEMAN, S.; ECK, D.; SIMONYAN, K. This time with feeling: Learning expressive musical performance. *Neural Computing and Applications*, Springer, v. 32, n. 4, p. 955–967, 2020. Citado 2 vezes nas páginas 39 e 40.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L. et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, v. 32, p. 8026–8037, 2019. Citado na página 51.

PAYNE, C.; OPENAI. *"MuseNet"*. [S.l.], 2019. Acesso em: 16-04-2020. Disponível em: <openai.com/blog/musenet>. Citado na página 40.

POSNER, J.; RUSSELL, J. A.; PETERSON, B. S. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, v. 17, p. 715 – 734, 2005. Citado na página 35.

RAVANELLI, M.; ZHONG, J.; PASCUAL, S.; SWIETOJANSKI, P.; MONTEIRO, J.; TRMAL, J.; BENGIO, Y. Multi-task self-supervised learning for robust speech recognition. In: IEEE. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2020. p. 6989–6993. Citado na página 16.

RAZAVI, A.; OORD, A. van den; VINYALS, O. Generating diverse high-fidelity images with VQ-VAE-2. *CoRR*, abs/1906.00446, 2019. Disponível em: <http://arxiv.org/abs/1906.00446>. Citado 2 vezes nas páginas 34 e 37.

RUSSELL, J. A. A circumplex model of affect. *Journal of personality and social psychology*, American Psychological Association, v. 39, n. 6, p. 1161, 1980. Citado 3 vezes nas páginas 17, 34 e 61.

SALIMANS, T.; GOODFELLOW, I. J.; ZAREMBA, W.; CHEUNG, V.; RADFORD, A.; CHEN, X. Improved techniques for training GANs. *CoRR*, abs/1606.03498, 2016. Disponível em: <http://arxiv.org/abs/1606.03498>. Citado 4 vezes nas páginas 26, 42, 50 e 72.

SALIMANS, T.; KINGMA, D. P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in neural information processing systems*, v. 29, p. 901–909, 2016. Citado na página 57.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. Disponível em: <http://arxiv.org/abs/1512.00567>. Citado 2 vezes nas páginas 26 e 50.

TAJBAKHSH, N.; JEYASEELAN, L.; LI, Q.; CHIANG, J. N.; WU, Z.; DING, X. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, Elsevier, v. 63, p. 101693, 2020. Citado na página 16.

TODD, P. M. A connectionist approach to algorithmic composition. *Computer Music Journal*, JSTOR, v. 13, n. 4, p. 27–43, 1989. Citado na página 36.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L. u.; POLOSUKHIN, I. Attention is all you need. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN,

S.; GARNETT, R. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>. Citado 9 vezes nas páginas 17, 29, 37, 38, 44, 54, 61, 62 e 72.

VILLANI, C. *Optimal transport: old and new*. [S.l.]: Springer, 2009. v. 338. Citado na página 23.

WANG, T.-C.; LIU, M.-Y.; ZHU, J.-Y.; TAO, A.; KAUTZ, J.; CATANZARO, B. High-resolution image synthesis and semantic manipulation with conditional gans. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 8798–8807. Citado na página 38.

WILLIAMS, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, Springer, v. 8, n. 3, p. 229–256, 1992. Citado na página 28.

YAMAMOTO, R.; SONG, E.; KIM, J.-M. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In: IEEE. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2020. p. 6199–6203. Citado na página 38.

YANG, L.-C.; CHOU, S.-Y.; YANG, Y.-H. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. In: *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR'18)*. [S.l.: s.n.], 2017. Citado na página 41.

YANG, L.-C.; LERCH, A. On the evaluation of generative models in music. *Neural Computing and Applications*, Springer, v. 32, n. 9, p. 4773–4784, 2020. Citado na página 42.

YU, L.; ZHANG, W.; WANG, J.; YU, Y. Seqgan: Sequence generative adversarial nets with policy gradient. In: *Proceedings of the AAAI conference on artificial intelligence*. [S.l.: s.n.], 2017. v. 31, n. 1. Citado 2 vezes nas páginas 17 e 28.

ZHANG, H.; GOODFELLOW, I.; METAXAS, D.; ODENA, A. *Self-Attention Generative Adversarial Networks*. 2019. Citado 3 vezes nas páginas 44, 45 e 54.

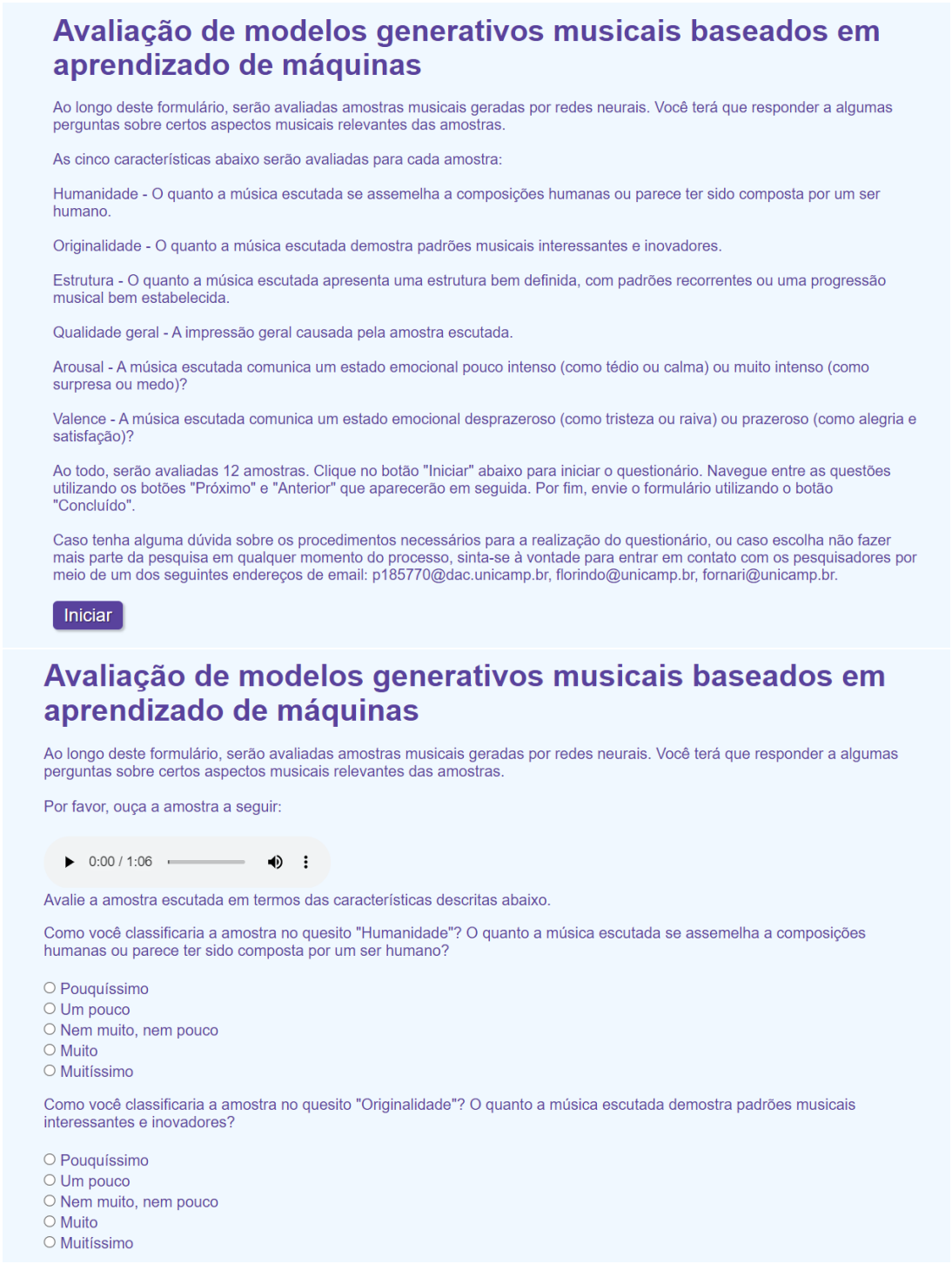
ZHANG, N. Learning adversarial transformer for symbolic music generation. *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–10, 2020. Citado 2 vezes nas páginas 41 e 66.

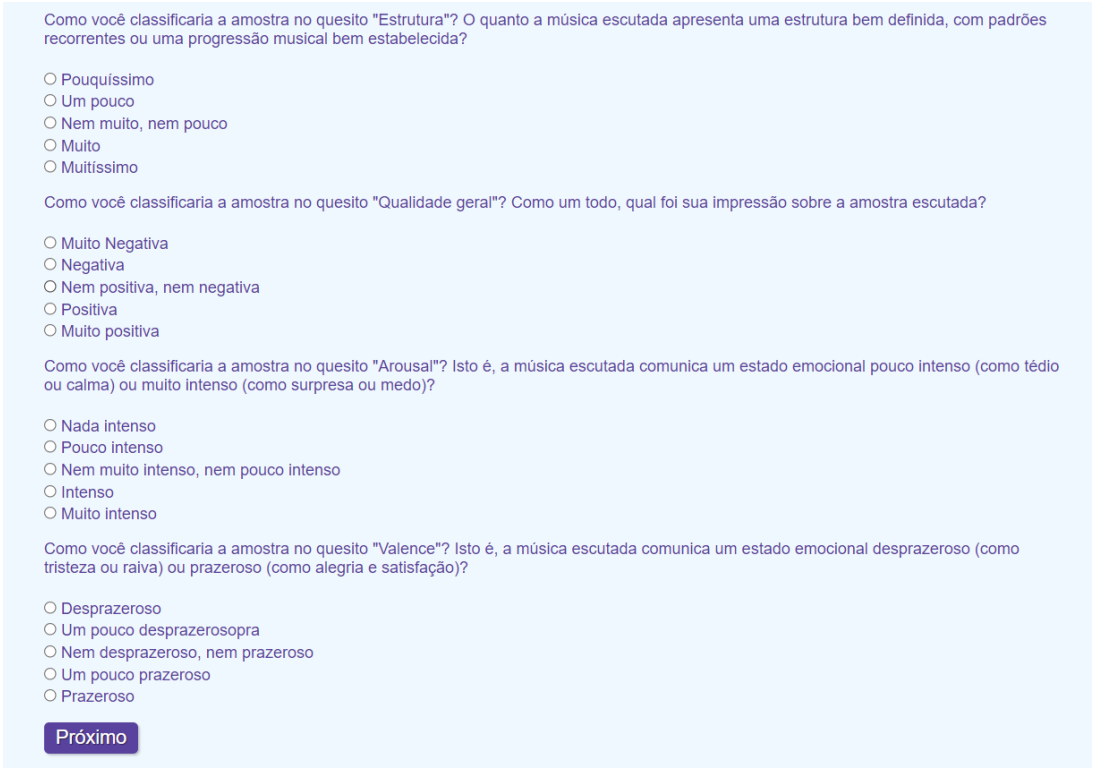
ZHANG, R.; ISOLA, P.; EFROS, A. A.; SHECHTMAN, E.; WANG, O. The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 586–595. Citado na página 28.

ZHAO, S.; LIU, Z.; LIN, J.; ZHU, J.-Y.; HAN, S. Differentiable augmentation for data-efficient gan training. In: *Conference on Neural Information Processing Systems (NeurIPS)*. [S.l.: s.n.], 2020. Citado na página 48.

# APÊNDICE A - Formulário de avaliação dos modelos Transformer

Figura 16 – Formulário de avaliação dos modelos de geração Musical Simbólica.






Fonte: (NEVES, 2021)