Seminário de Deep Learning

ViTGAN: Training GANs with Vision Transformers

Juan Belieni
juan.araujo@fgv.edu.br

Juliana Souza
juliana.souza.1@fgv.edu.br

FGV/EMAp

Novembro de 2023

Introdução

ViTGAN

ViTGAN é um modelo de geração de imagens que utiliza Vision Transformes (ViTs) para o treinamento de uma Generative Adversarial Network (GAN), possuindo resultados comparáveis a GANs baseadas em CNNs.

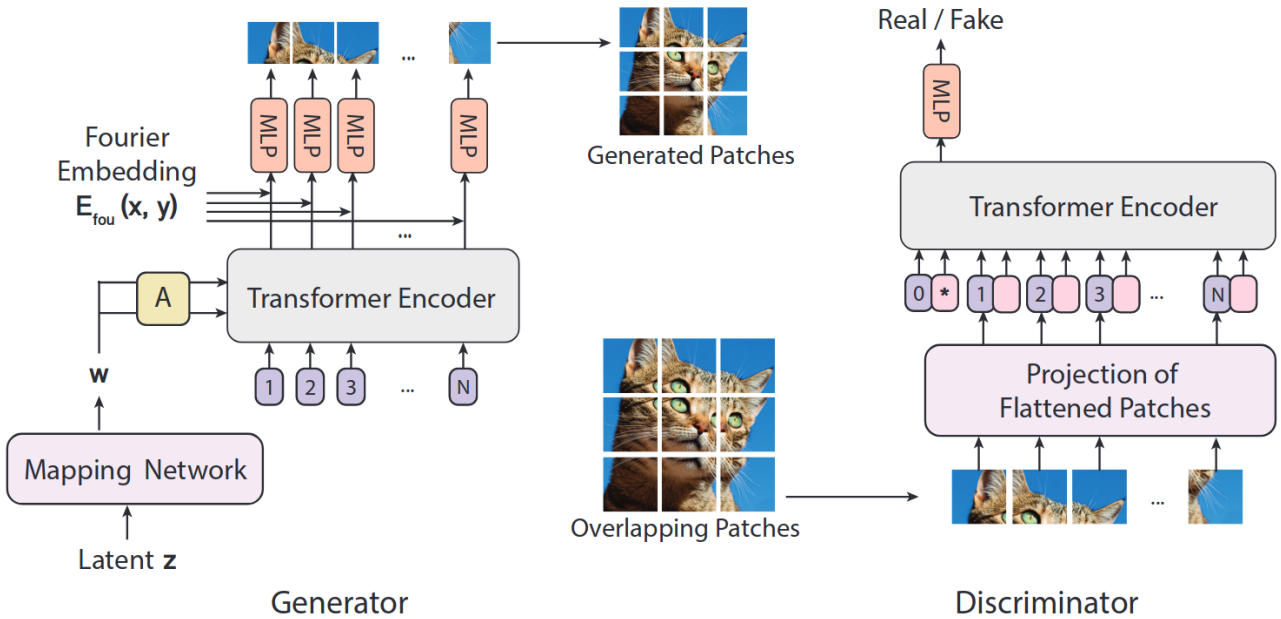


Figura: arquitetura do ViTGAN (Lee et al. [1]).

Vision Transformer (ViT)

ViT é uma arquitetura que se baseia exclusivamente no conceito de tranformers para classificação de imagens. Suas operações consistem em:

- Dividir a imagem em uma sequência de patches $\mathbf{x}_p \in \mathbb{R}^{N \cdot (P^2 \cdot C)}$.
- Projetar os patches em um espaço latente $\mathbb{R}^D$ por meio de uma transformação linear.
- Adicionar um embedding posicional unidimensional em cada patch.
- Utilizar o resultado dessas operações como entrada em um transformer encoder.

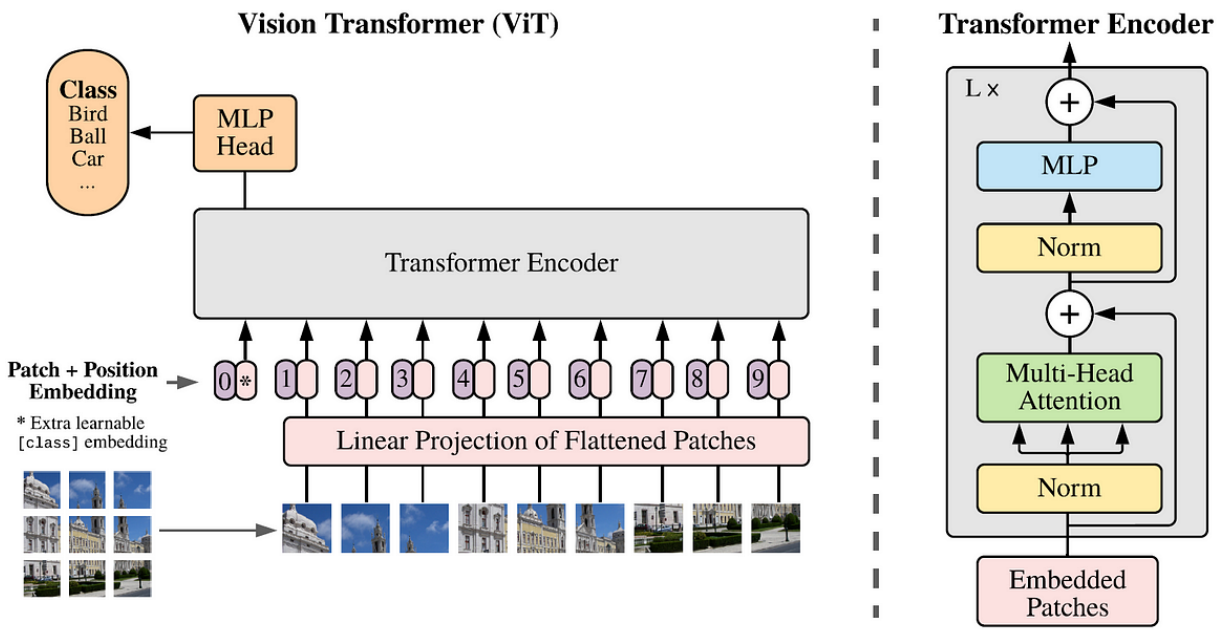


Figura: arquitetura do ViT (Dosovitskiy et al. [2]).

Vision Transformer (ViT)

ViT é uma arquitetura que se baseia exclusivamente no conceito de tranformers para classificação de imagens. Suas operações consistem em:

- Dividir a imagem em uma sequência de patches $\mathbf{x}_p \in \mathbb{R}^{N \cdot (P^2 \cdot C)}$.
- Projetar os patches em um espaço latente $\mathbb{R}^D$ por meio de uma transformação linear.
- Adicionar um embedding posicional unidimensional em cada patch.
- Utilizar o resultado dessas operações como entrada em um transformer encoder.

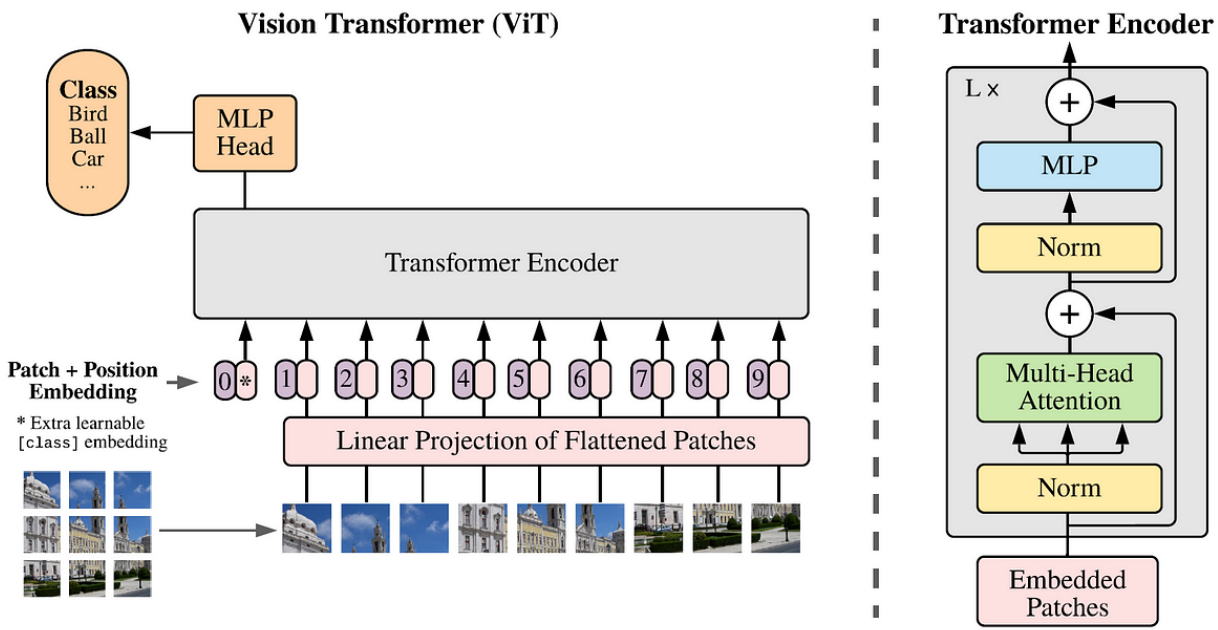


Figura: arquitetura do ViT (Dosovitskiy et al. [2]).

Vision Transformer (ViT)

ViT é uma arquitetura que se baseia exclusivamente no conceito de tranformers para classificação de imagens. Suas operações consistem em:

- Dividir a imagem em uma sequência de patches $\mathbf{x}_p \in \mathbb{R}^{N \cdot (P^2 \cdot C)}$.
- Projetar os patches em um espaço latente $\mathbb{R}^D$ por meio de uma transformação linear.
- Adicionar um embedding posicional unidimensional em cada patch.
- Utilizar o resultado dessas operações como entrada em um transformer encoder.

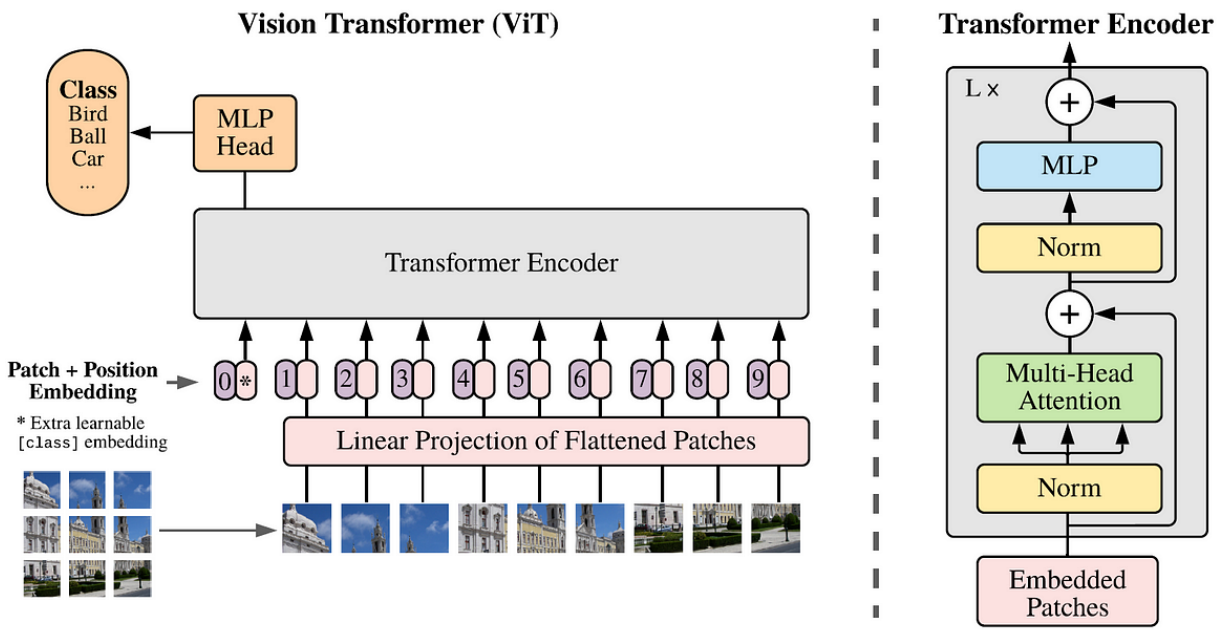


Figura: arquitetura do ViT (Dosovitskiy et al. [2]).

Vision Transformer (ViT)

ViT é uma arquitetura que se baseia exclusivamente no conceito de tranformers para classificação de imagens. Suas operações consistem em:

- Dividir a imagem em uma sequência de patches $\mathbf{x}_p \in \mathbb{R}^{N \cdot (P^2 \cdot C)}$.
- Projetar os patches em um espaço latente $\mathbb{R}^D$ por meio de uma transformação linear.
- Adicionar um embedding posicional unidimensional em cada patch.
- Utilizar o resultado dessas operações como entrada em um transformer encoder.

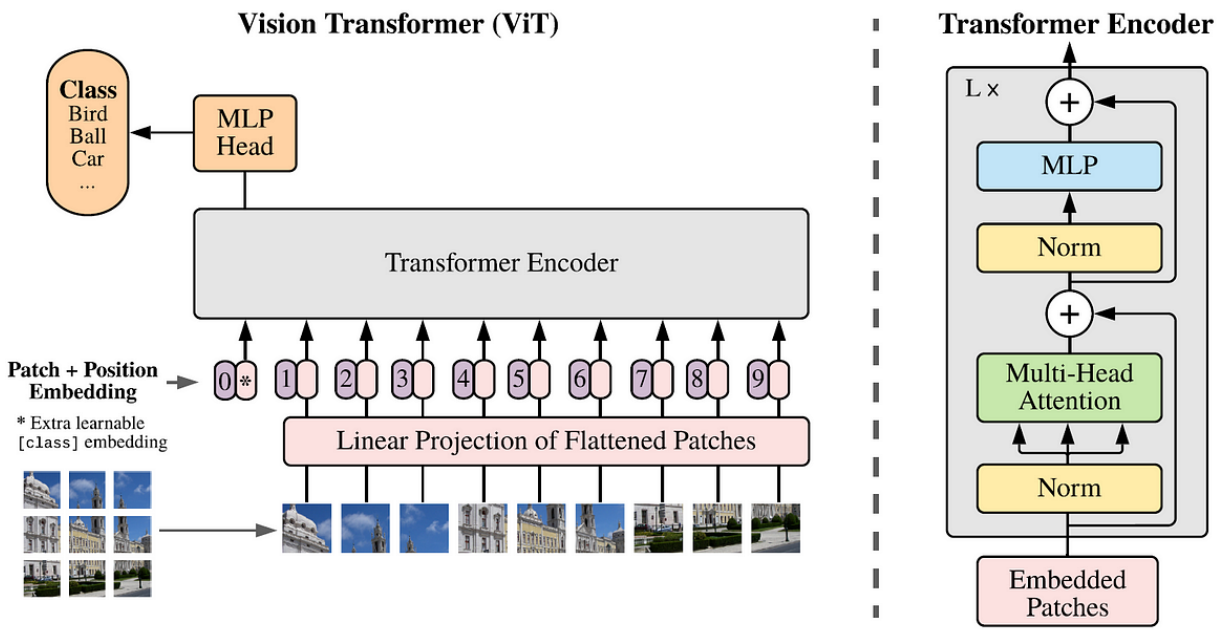


Figura: arquitetura do ViT (Dosovitskiy et al. [2]).

Vision Transformer (ViT)

ViT é uma arquitetura que se baseia exclusivamente no conceito de tranformers para classificação de imagens. Suas operações consistem em:

- Dividir a imagem em uma sequência de patches $\mathbf{x}_p \in \mathbb{R}^{N \cdot (P^2 \cdot C)}$.
- Projetar os patches em um espaço latente $\mathbb{R}^D$ por meio de uma transformação linear.
- Adicionar um embedding posicional unidimensional em cada patch.
- Utilizar o resultado dessas operações como entrada em um transformer encoder.

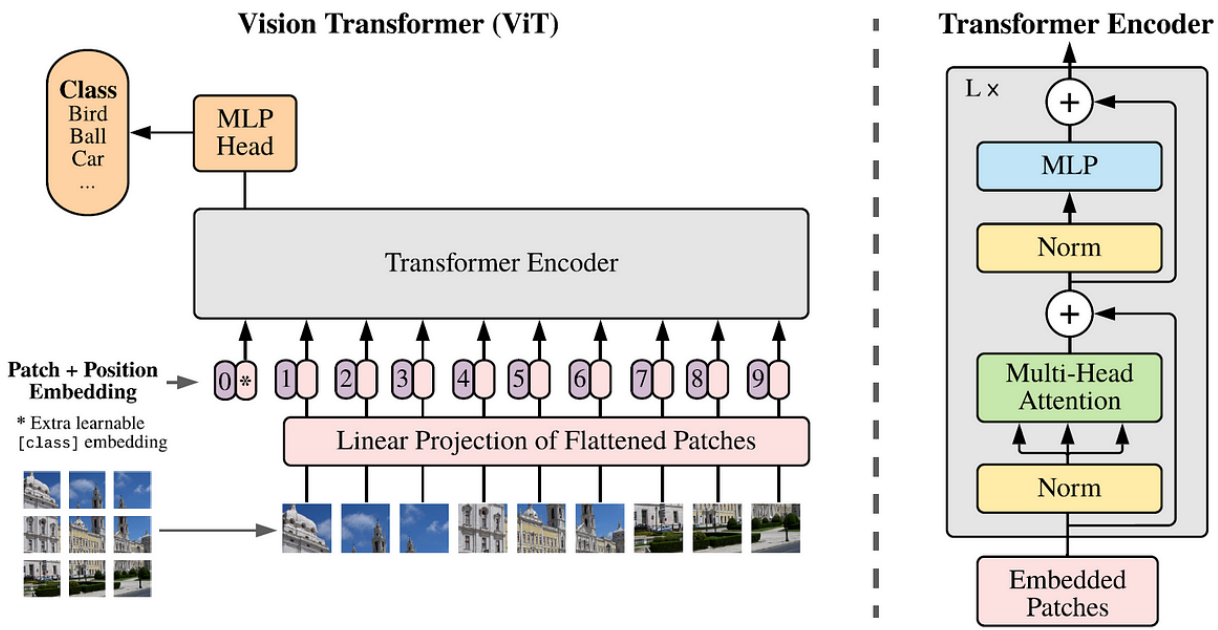


Figura: arquitetura do ViT (Dosovitskiy et al. [2]).

Houston, we have a problem...

Problemas com o ViT

Utilizar diretamente o ViT nas redes geradora e discriminadora pode tornar o processo de treinamento instável e volátil. Para resolver esses problemas, duas ideias foram consideradas:

1. Regularizar a rede discriminadora baseada em ViT.
   - Restringindo o discriminador para ser Lipschitz contínuo.
   - Normalização espectral aprimorada.
   - Sobreposição dos patches das imagens.
2. Construir uma nova arquitetura para a rede geradora.
   - *LayerNorm automodulada*.
   - *Representação neural implícita para geração de patches*.

Problemas com o ViT

Utilizar diretamente o ViT nas redes geradora e discriminadora pode tornar o processo de treinamento instável e volátil. Para resolver esses problemas, duas ideias foram consideradas:

1. Regularizar a rede discriminadora baseada em ViT.
   - Restringindo o discriminador para ser Lipschitz contínuo.
   - Normalização espectral aprimorada.
   - Sobreposição dos patches das imagens.
2. Construir uma nova arquitetura para a rede geradora.
   - *LayerNorm automodulada.*
   - *Representação neural implícita para geração de patches.*

Problemas com o ViT

Utilizar diretamente o ViT nas redes geradora e discriminadora pode tornar o processo de treinamento instável e volátil. Para resolver esses problemas, duas ideias foram consideradas:

1. Regularizar a rede discriminadora baseada em ViT.
   - Restringindo o discriminador para ser Lipschitz contínuo.
   - Normalização espectral aprimorada.
   - Sobreposição dos patches das imagens.
2. Construir uma nova arquitetura para a rede geradora.
   - *LayerNorm automodulada*.
   - *Representação neural implícita para geração de patches*.

Regularizar a rede discriminadora baseada em ViT

Restringindo o discriminador para ser Lipschitz contínuo

A continuidade de Lipschitz é uma forma mais forte de continuidade para funções.

Definition 2.1. Given two metric spaces $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$, a function $f : \mathcal{X} \to \mathcal{Y}$ is called *Lipschitz continuous* (*or K-Lipschitz*) if there exists a constant $K \geq 0$ such that

$$d_{\mathcal{Y}}(f(x), f(x')) \leq K \cdot d_{\mathcal{X}}(x, x') \quad \forall x, x' \in \mathcal{X}.$$

The smallest such K is the *Lipschitz constant* of f, denoted $\text{Lip}(f)$ (Kim, Papamakarios e Mnih [3]).

Restringindo o discriminador para ser Lipschitz contínuo

A continuidade de Lipschitz é uma forma mais forte de continuidade para funções.

Definition 2.1. Given two metric spaces $(\mathcal{X}, d_{\mathcal{X}})$ and $(\mathcal{Y}, d_{\mathcal{Y}})$, a function $f : \mathcal{X} \to \mathcal{Y}$ is called *Lipschitz continuous* (*or K-Lipschitz*) if there exists a constant $K \geq 0$ such that

$$d_{\mathcal{Y}}(f(x), f(x')) \leq K \cdot d_{\mathcal{X}}(x, x') \quad \forall x, x' \in \mathcal{X}.$$

The smallest such K is the *Lipschitz constant* of f, denoted $\text{Lip}(f)$ (Kim, Papamakarios e Mnih [3]).

Restringindo o discriminador para ser Lipschitz contínuo

Foi demonstrado que um discriminador Lipschitz garante a existência de uma função discriminadora ótima e a existência de um único equilíbro de Nash. No entanto, também foi mostrado que a constante de Lipschitz de um mecanismo padrão de atenção pode ser ilimitado.

Dessa forma, para garantir a continuidade de Lipschitz, foi adotada uma atenção L2, onde a similaridade por produto escalar foi substituída por uma distância euclidiana e os pesos W_q e W_k foram igualados:

$$\text{Attention}_h(\mathbf{X}) = \text{softmax}\left(\frac{d(\mathbf{X}\mathbf{W}_q, \mathbf{X}\mathbf{W}_k)}{\sqrt{d_h}}\right)\mathbf{X}\mathbf{W}_v, \quad \text{onde } \mathbf{W}_q = \mathbf{W}_k.$$

Essa alteração melhora a estabilidade do transformer quando utilizada em redes discriminadores de GANs.

Restringindo o discriminador para ser Lipschitz contínuo

Foi demonstrado que um discriminador Lipschitz garante a existência de uma função discriminadora ótima e a existência de um único equilíbro de Nash. No entanto, também foi mostrado que a constante de Lipschitz de um mecanismo padrão de atenção pode ser ilimitado.

Dessa forma, para garantir a continuidade de Lipschitz, foi adotada uma atenção L2, onde a similaridade por produto escalar foi substituída por uma distância euclidiana e os pesos W_q e W_k foram igualados:

$$\text{Attention}_h(\mathbf{X}) = \text{softmax}\left(\frac{d(\mathbf{X}\mathbf{W}_q, \mathbf{X}\mathbf{W}_k)}{\sqrt{d_h}}\right)\mathbf{X}\mathbf{W}_v, \quad \text{onde } \mathbf{W}_q = \mathbf{W}_k.$$

Essa alteração melhora a estabilidade do transformer quando utilizada em redes discriminadores de GANs.

Restringindo o discriminador para ser Lipschitz contínuo

Foi demonstrado que um discriminador Lipschitz garante a existência de uma função discriminadora ótima e a existência de um único equilíbro de Nash. No entanto, também foi mostrado que a constante de Lipschitz de um mecanismo padrão de atenção pode ser ilimitado.

Dessa forma, para garantir a continuidade de Lipschitz, foi adotada uma atenção L2, onde a similaridade por produto escalar foi substituída por uma distância euclidiana e os pesos W_q e W_k foram igualados:

$$\text{Attention}_h(\mathbf{X}) = \text{softmax}\left(\frac{d(\mathbf{X}\mathbf{W}_q, \mathbf{X}\mathbf{W}_k)}{\sqrt{d_h}}\right)\mathbf{X}\mathbf{W}_v, \quad \text{onde} \;\; \mathbf{W}_q = \mathbf{W}_k.$$

Essa alteração melhora a estabilidade do transformer quando utilizada em redes discriminadores de GANs.

Restringindo o discriminador para ser Lipschitz contínuo

Foi demonstrado que um discriminador Lipschitz garante a existência de uma função discriminadora ótima e a existência de um único equilíbro de Nash. No entanto, também foi mostrado que a constante de Lipschitz de um mecanismo padrão de atenção pode ser ilimitado.

Dessa forma, para garantir a continuidade de Lipschitz, foi adotada uma atenção L2, onde a similaridade por produto escalar foi substituída por uma distância euclidiana e os pesos W_q e W_k foram igualados:

$$\text{Attention}_h(\mathbf{X}) = \text{softmax}\left(\frac{d(\mathbf{X}\mathbf{W}_q, \mathbf{X}\mathbf{W}_k)}{\sqrt{d_h}}\right)\mathbf{X}\mathbf{W}_v, \quad \text{onde} \;\; \mathbf{W}_q = \mathbf{W}_k.$$

Essa alteração melhora a estabilidade do transformer quando utilizada em redes discriminadores de GANs.

Normalização espectral aprimorada

A normalização espectral, nesse trabalho, é utilizada para fortalecer a continuidade de Lipschitz no treinamento da rede discriminadora. Essa normalização consiste em estimar a norma espectral da matriz de projeção em cada camada da rede neural, utilizando essa estimativa para dividir a matriz de pesos de tal maneira que a constante de Lipschitz da matriz de projeção seja igual a 1.

Porém, percebeu-se que blocos de transformer são sensíveis à escala da constante de Lipschitz e utilizar apenas a normalização espectral torna a etapa de treinamento extremamente lenta. Para resolver isso, foi proposto multiplicar a matriz de pesos de cada camada pela norma espectral dos pesos iniciais:

$$\bar{W}_{\text{ISN}}(\mathbf{W}) := \sigma_{\text{esp.}}(\mathbf{W}_{\text{inicial}}) \cdot \frac{\mathbf{W}}{\sigma_{\text{esp.}}(\mathbf{W})}.$$

Normalização espectral aprimorada

A normalização espectral, nesse trabalho, é utilizada para fortalecer a continuidade de Lipschitz no treinamento da rede discriminadora. Essa normalização consiste em estimar a norma espectral da matriz de projeção em cada camada da rede neural, utilizando essa estimativa para dividir a matriz de pesos de tal maneira que a constante de Lipschitz da matriz de projeção seja igual a 1.

Porém, percebeu-se que blocos de transformer são sensíveis à escala da constante de Lipschitz e utilizar apenas a normalização espectral torna a etapa de treinamento extremamente lenta. Para resolver isso, foi proposto multiplicar a matriz de pesos de cada camada pela norma espectral dos pesos iniciais:

$$\bar{W}_{\text{ISN}}(\mathbf{W}) := \sigma_{\text{esp.}}(\mathbf{W}_{\text{inicial}}) \cdot \frac{\mathbf{W}}{\sigma_{\text{esp.}}(\mathbf{W})}.$$

Normalização espectral aprimorada

A normalização espectral, nesse trabalho, é utilizada para fortalecer a continuidade de Lipschitz no treinamento da rede discriminadora. Essa normalização consiste em estimar a norma espectral da matriz de projeção em cada camada da rede neural, utilizando essa estimativa para dividir a matriz de pesos de tal maneira que a constante de Lipschitz da matriz de projeção seja igual a 1.

Porém, percebeu-se que blocos de transformer são sensíveis à escala da constante de Lipschitz e utilizar apenas a normalização espectral torna a etapa de treinamento extremamente lenta. Para resolver isso, foi proposto multiplicar a matriz de pesos de cada camada pela norma espectral dos pesos iniciais:

$$\bar{W}_{\text{ISN}}(\mathbf{W}) := \sigma_{\text{esp.}}(\mathbf{W}_{\text{inicial}}) \cdot \frac{\mathbf{W}}{\sigma_{\text{esp.}}(\mathbf{W})}.$$

Sobreposição dos patches das imagens

Os discriminadores baseados em ViT são bem propensos a overfitting devido sua alta capacidade de aprendizagem. Por causa disso, utilizar patches $P \times P$ sem sobreposição em um grid predefinido pode incentivar a rede discriminadora a aprender características locais e não oferecer uma loss significativa para a rede geradora.

Para mitigar esse problema, foi utilizada uma técnica simples de sobreposição entre os patches ao estender a imagem em o pixeis, tornando o tamanho do patch $(P + 2o) \times (P + 2o)$.

Isso resulta em uma sequência com o mesmo comprimento, mas com menos sensibilidade às grades predefinidas. Também pode oferecer ao transformer uma noção melhor de quais são os patches vizinhos do patch atual, dando assim uma melhor noção de localidade.

Sobreposição dos patches das imagens

Os discriminadores baseados em ViT são bem propensos a overfitting devido sua alta capacidade de aprendizagem. Por causa disso, utilizar patches $P \times P$ sem sobreposição em um grid predefinido pode incentivar a rede discriminadora a aprender características locais e não oferecer uma loss significativa para a rede geradora.

Para mitigar esse problema, foi utilizada uma técnica simples de sobreposição entre os patches ao estender a imagem em o pixeis, tornando o tamanho do patch $(P + 2o) \times (P + 2o)$.

Isso resulta em uma sequência com o mesmo comprimento, mas com menos sensibilidade às grades predefinidas. Também pode oferecer ao transformer uma noção melhor de quais são os patches vizinhos do patch atual, dando assim uma melhor noção de localidade.

Sobreposição dos patches das imagens

Os discriminadores baseados em ViT são bem propensos a overfitting devido sua alta capacidade de aprendizagem. Por causa disso, utilizar patches $P \times P$ sem sobreposição em um grid predefinido pode incentivar a rede discriminadora a aprender características locais e não oferecer uma loss significativa para a rede geradora.

Para mitigar esse problema, foi utilizada uma técnica simples de sobreposição entre os patches ao estender a imagem em o pixeis, tornando o tamanho do patch $(P + 2o) \times (P + 2o)$.

Isso resulta em uma sequência com o mesmo comprimento, mas com menos sensibilidade às grades predefinidas. Também pode oferecer ao transformer uma noção melhor de quais são os patches vizinhos do patch atual, dando assim uma melhor noção de localidade.

Construir uma nova arquitetura para a rede geradora

Desafios para projetar um gerador baseado na arquitetura do ViT

- Adaptar o ViT, tradicionalmente preditivo, em um gerador de imagens não é uma tarefa trivial.
- Para entender como isso será feito, os autores apresentam dois modelos baselines para comparar com o modelo generativo proposto.

Modelos baseline

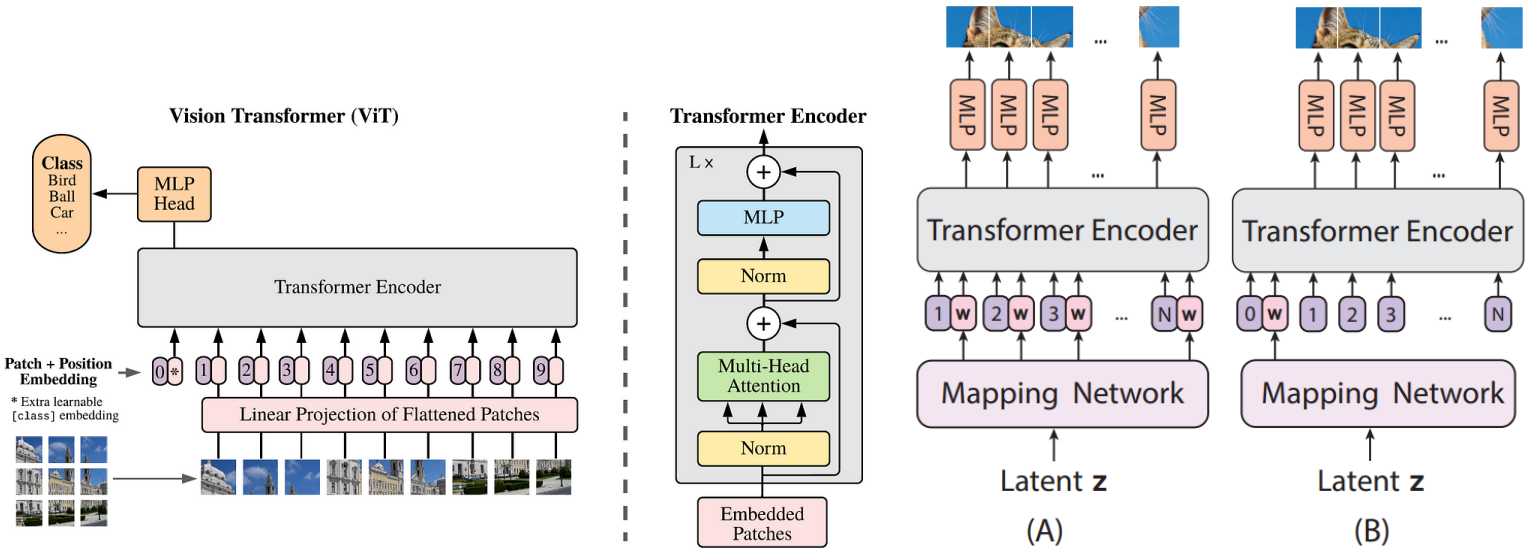


Figura: ViT (esquerda) vs modelos generativos baselines **(A)** e **(B)** (direita) (Lee et al. [1]).

A rede de mapeamento (mapping network) na figura 3 consiste em substituir a saída preditiva do ViT por uma saída que gera pixels a partir dos embeddings. Especificamente, do vetor latente w podemos escrever: $w = MLP(z)$, onde z é um ruído Gaussiano.

Observe que a diferença entre as baselines **(A)** e **(B)** é que **(A)** recebe o embedding intermediário w para cada embedding posicional, e **(B)** recebe o embedding e o acrescenta na sequência.

Modelos baseline

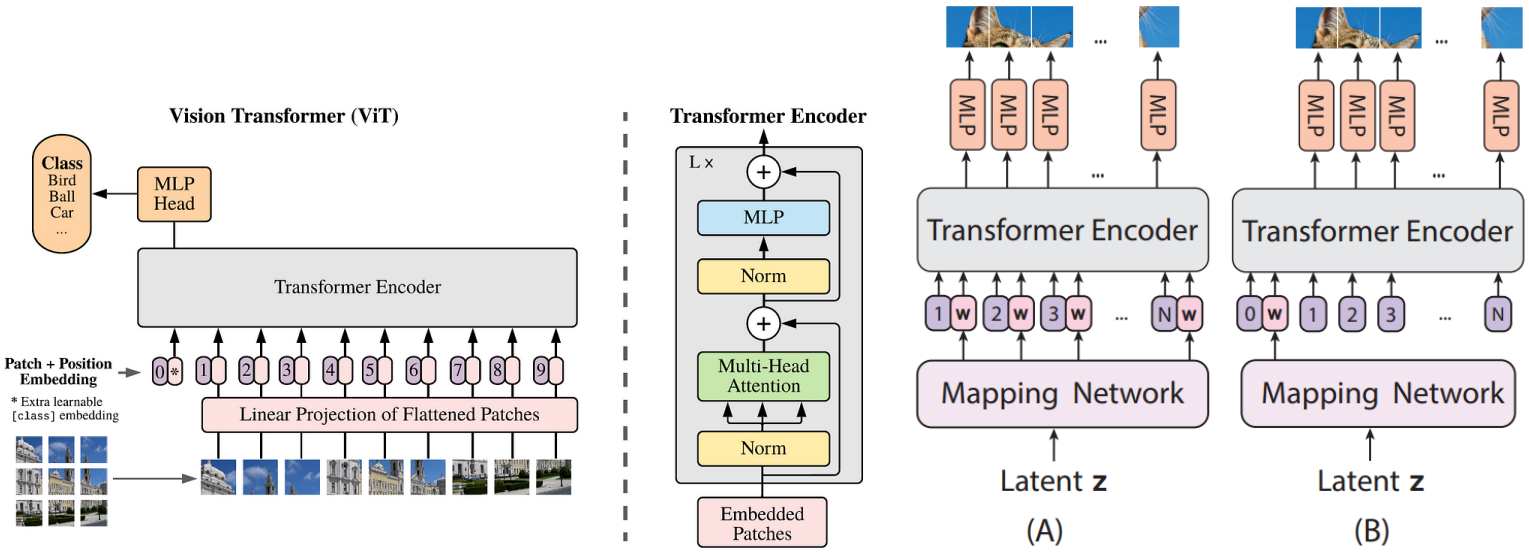


Figura: ViT (esquerda) vs modelos generativos baselines **(A)** e **(B)** (direita) (Lee et al. [1]).

A rede de mapeamento (mapping network) na figura 3 consiste em substituir a saída preditiva do ViT por uma saída que gera pixels a partir dos embeddings. Especificamente, do vetor latente w podemos escrever: $w = MLP(z)$, onde z é um ruído Gaussiano.

Observe que a diferença entre as baselines **(A)** e **(B)** é que **(A)** recebe o embedding intermediário w para cada embedding posicional, e **(B)** recebe o embedding e o acrescenta na sequência.

Geração de pixels

Para gerar os valores dos pixels:

- Para mapear a saída de um embedding de dimensão D para um patch de tamanho $P \times P \times C$, uma projeção linear $E \in \mathbb{R}^{D \times (P^2 \cdot C)}$ é apreendida em ambos os modelos.
- A sequência com um total de L patches $[x_p^i]_{i=1}^L$, onde $L = \frac{H \times W}{P^2}$ é reformulada para formar a imagem x de tamanho $H \times W$.

Arquitetura da rede geradora aprimorada

Os dois modelos baselines infelizmente não desempenham muito bem comparados a um gerador baseado em CNNs. Assim, é necessário propor um novo gerador aprimorado, que consiste em dois componenetes principais: um bloco com um transformer e uma camada de mapeamento de saída.

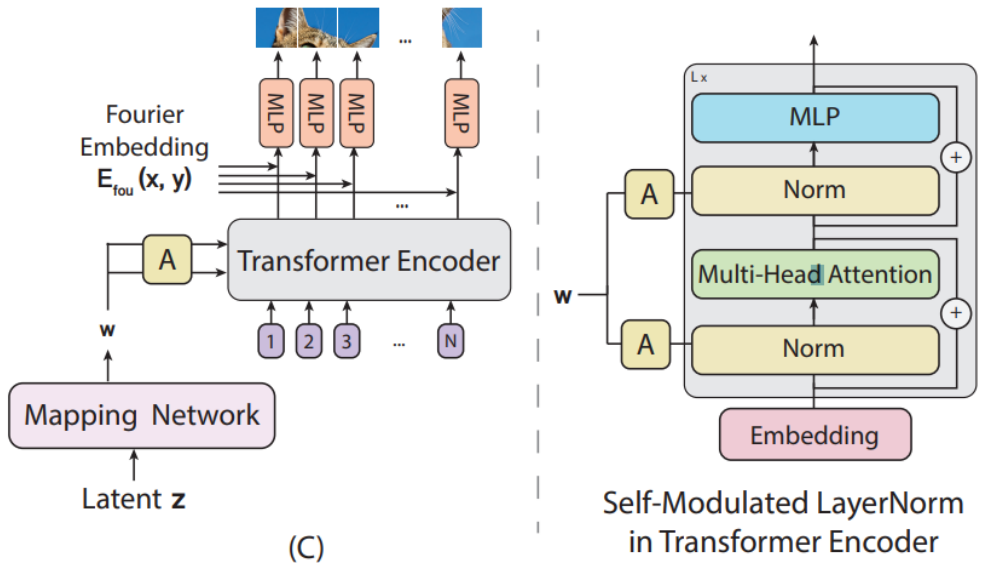


Figura: Gerador proposto. Aqui A indica a MLP utilizada para produzir os parâmetros γ e β a serem utilizados na layernorm automodulada. Assim, substitui-se a normalização pela layernorm automodulada (SLN) calculada pela transformação afim aprendida (denotada como A na figura) de w (Lee et al. [1]).

Equações do gerador

No ViT é utilizada a classificação, com x_{class} e E_{pos} formulando o patch embedding h_0; no ViTGAN $w = MLP(z)$ é utilizada para substituir o embedding de classificação h_L^0:

Equações do ViT:

$$\mathbf{h}_0 = \left[\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^L\mathbf{E}\right] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{\left(P^2 \cdot C\right) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(L+1) \times D} \tag{1}$$

$$\mathbf{h}'_\ell = \text{MSA}\left(\text{LN}\left(\mathbf{h}_{\ell-1}\right)\right) + \mathbf{h}_{\ell-1}, \qquad \ell = 1, \ldots, L \tag{2}$$

$$\mathbf{h}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{h}'_\ell\right)\right) + \mathbf{h}'_\ell, \qquad \ell = 1, \ldots, L \tag{3}$$

$$\mathbf{y} = \text{LN}\left(\mathbf{h}_L^0\right) \tag{4}$$

Equações do ViTGAN:

$$\mathbf{h}_0 = \mathbf{E}_{\text{pos}}, \qquad \mathbf{E}_{\text{pos}} \in \mathbb{R}^{L \times D} \tag{5}$$

$$\mathbf{h}'_\ell = \text{MSA}\left(\text{SLN}\left(\mathbf{h}_{\ell-1}, \mathbf{w}\right)\right) + \mathbf{h}_{\ell-1}, \qquad \ell = 1, \ldots, L, \quad \mathbf{w} \in \mathbb{R}^D \tag{6}$$

$$\mathbf{h}_\ell = \text{MLP}\left(\text{SLN}\left(\mathbf{h}'_\ell, \mathbf{w}\right)\right) + \mathbf{h}'_\ell, \qquad \ell = 1, \ldots, L \tag{7}$$

$$\mathbf{y} = \text{SLN}\left(\mathbf{h}_L, \mathbf{w}\right) = \left[\mathbf{y}^1, \cdots, \mathbf{y}^L\right], \qquad \mathbf{y}^1, \ldots, \mathbf{y}^L \in \mathbb{R}^D \tag{8}$$

$$\mathbf{x} = \left[f_\theta\left(\mathbf{E}_{\text{fou}}, \mathbf{y}^1\right), \ldots, f_\theta\left(\mathbf{E}_{\text{fou}}, \mathbf{y}^L\right)\right], \qquad \mathbf{x}_p^i \in \mathbb{R}^{P^2 \times C}, \quad \mathbf{x} \in \mathbb{R}^{H \times W \times C} \tag{9}$$

Equações do gerador

No ViT é utilizada a classificação, com x_{class} e E_{pos} formulando o patch embedding h_0; no ViTGAN $w = MLP(z)$ é utilizada para substituir o embedding de classificação h_L^0:

Equações do ViT:

$$\mathbf{h}_0 = \left[\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^L\mathbf{E}\right] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{\left(P^2 \cdot C\right) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(L+1) \times D} \tag{1}$$

$$\mathbf{h}'_\ell = \text{MSA}\left(\text{LN}\left(\mathbf{h}_{\ell-1}\right)\right) + \mathbf{h}_{\ell-1}, \qquad \ell = 1, \ldots, L \tag{2}$$

$$\mathbf{h}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{h}'_\ell\right)\right) + \mathbf{h}'_\ell, \qquad \ell = 1, \ldots, L \tag{3}$$

$$\mathbf{y} = \text{LN}\left(\mathbf{h}_L^0\right) \tag{4}$$

Equações do ViTGAN:

$$\mathbf{h}_0 = \mathbf{E}_{\text{pos}}, \qquad \mathbf{E}_{\text{pos}} \in \mathbb{R}^{L \times D} \tag{5}$$

$$\mathbf{h}'_\ell = \text{MSA}\left(\text{SLN}\left(\mathbf{h}_{\ell-1}, \mathbf{w}\right)\right) + \mathbf{h}_{\ell-1}, \qquad \ell = 1, \ldots, L, \quad \mathbf{w} \in \mathbb{R}^D \tag{6}$$

$$\mathbf{h}_\ell = \text{MLP}\left(\text{SLN}\left(\mathbf{h}'_\ell, \mathbf{w}\right)\right) + \mathbf{h}'_\ell, \qquad \ell = 1, \ldots, L \tag{7}$$

$$\mathbf{y} = \text{SLN}\left(\mathbf{h}_L, \mathbf{w}\right) = \left[\mathbf{y}^1, \cdots, \mathbf{y}^L\right], \qquad \mathbf{y}^1, \ldots, \mathbf{y}^L \in \mathbb{R}^D \tag{8}$$

$$\mathbf{x} = \left[f_\theta\left(\mathbf{E}_{\text{fou}}, \mathbf{y}^1\right), \ldots, f_\theta\left(\mathbf{E}_{\text{fou}}, \mathbf{y}^L\right)\right], \qquad \mathbf{x}_p^i \in \mathbb{R}^{P^2 \times C}, \quad \mathbf{x} \in \mathbb{R}^{H \times W \times C} \tag{9}$$

Equações do gerador

No ViT é utilizada a classificação, com x_{class} e E_{pos} formulando o patch embedding h_0; no ViTGAN $w = MLP(z)$ é utilizada para substituir o embedding de classificação h_L^0:

Equações do ViT:

$$\mathbf{h}_0 = \left[\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^L\mathbf{E}\right] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{\left(P^2 \cdot C\right) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(L+1) \times D} \tag{1}$$

$$\mathbf{h}'_\ell = \text{MSA}\left(\text{LN}\left(\mathbf{h}_{\ell-1}\right)\right) + \mathbf{h}_{\ell-1}, \qquad \ell = 1, \ldots, L \tag{2}$$

$$\mathbf{h}_\ell = \text{MLP}\left(\text{LN}\left(\mathbf{h}'_\ell\right)\right) + \mathbf{h}'_\ell, \qquad \ell = 1, \ldots, L \tag{3}$$

$$\mathbf{y} = \text{LN}\left(\mathbf{h}_L^0\right) \tag{4}$$

Equações do ViTGAN:

$$\mathbf{h}_0 = \mathbf{E}_{\text{pos}}, \qquad \mathbf{E}_{\text{pos}} \in \mathbb{R}^{L \times D} \tag{5}$$

$$\mathbf{h}'_\ell = \text{MSA}\left(\text{SLN}\left(\mathbf{h}_{\ell-1}, \mathbf{w}\right)\right) + \mathbf{h}_{\ell-1}, \qquad \ell = 1, \ldots, L, \quad \mathbf{w} \in \mathbb{R}^D \tag{6}$$

$$\mathbf{h}_\ell = \text{MLP}\left(\text{SLN}\left(\mathbf{h}'_\ell, \mathbf{w}\right)\right) + \mathbf{h}'_\ell, \qquad \ell = 1, \ldots, L \tag{7}$$

$$\mathbf{y} = \text{SLN}\left(\mathbf{h}_L, \mathbf{w}\right) = \left[\mathbf{y}^1, \cdots, \mathbf{y}^L\right], \qquad \mathbf{y}^1, \ldots, \mathbf{y}^L \in \mathbb{R}^D \tag{8}$$

$$\mathbf{x} = \left[f_\theta\left(\mathbf{E}_{\text{fou}}, \mathbf{y}^1\right), \ldots, f_\theta\left(\mathbf{E}_{\text{fou}}, \mathbf{y}^L\right)\right], \qquad \mathbf{x}_p^i \in \mathbb{R}^{P^2 \times C}, \quad \mathbf{x} \in \mathbb{R}^{H \times W \times C} \tag{9}$$

Operação de LayerNorm automodulada

Na equação abaixo utiliza-se z para modular a operação de LayerNorm, ao invés de enviar o vetor de ruído z como input do **ViT**. Essa operação é chamada de automodulação por não depender de informações externas.

$$\mathrm{SLN}(h_\ell, w) = \mathrm{SLN}(h_\ell, \mathrm{MLP}(\mathbf{z})) = \gamma_\ell(w) \cdot \frac{h_\ell - \mu}{\sigma} + \beta_\ell(w)$$

onde

- μ corresponde a média das entradas somadas dentro da camada
- σ e registra a variância dessas entradas somadas
- γ_ℓ e β_ℓ computam parâmetros de normalização adaptativas controlados pelo vetor latente derivado de z

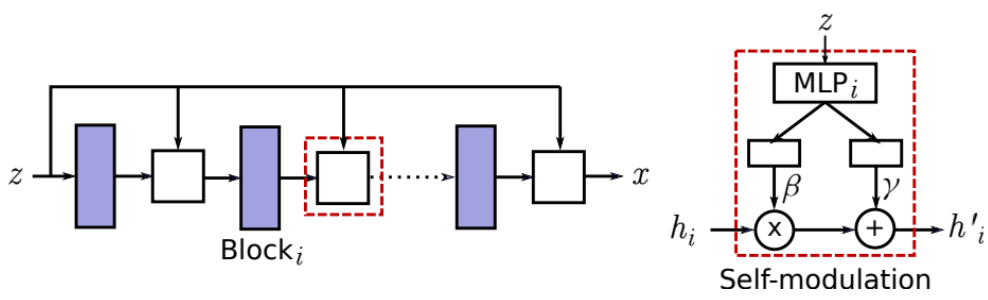


Figura: Automodulação que, a partir de z produzem os parâmetros β e γ por meio de uma MLP (Chen et al. [4]).

Representação neural implícita para geração de patches

- Para aprender um mapeamento contínuo de um patch embedding $y^i \in \mathbb{R}^D$ para valores de patch pixel $x_p^i \in \mathbb{P} \times \mathbb{C}$, utiliza-se uma **representação neural implícita**.
- As representações implícitas podem restringir o espaço das amostras geradas ao espaço de sinais naturais de variação suave
  - caso sejam combinadas com features de Fourrier ou
  - funções de ativação senoidal.
- $x_p^i = f_\theta(\mathrm{E}_{fou}, y^i)$, onde $\mathrm{E}_{fou} \in \mathbb{R}^{P^2 \cdot D}$ são codificação de Fourier (função de ativação seno) para localizações espaciais $P \times P$
  - $f_\theta(.,.)$ trata-se de uma MPL de 2 camadas.

w Embedding	Output Mapping	FID ↓	IS ↑
Fig 2 (A)	Linear	14.3	8.60
Fig 2 (A)	NeurRep	11.3	9.05
Fig 2 (B)	Linear	328	1.01
Fig 2 (B)	NeurRep	285	2.46
Fig 2 (C)	Linear	15.1	8.58
Fig 2 (C)	NeurRep	**6.66**	**9.30**

(a) **Generator Ablation Studies.** NeurRep denotes implicit neural representation.

Figura: A representação neural implícita é particularmente útil para treinar GANs com geradores baseados em ViT

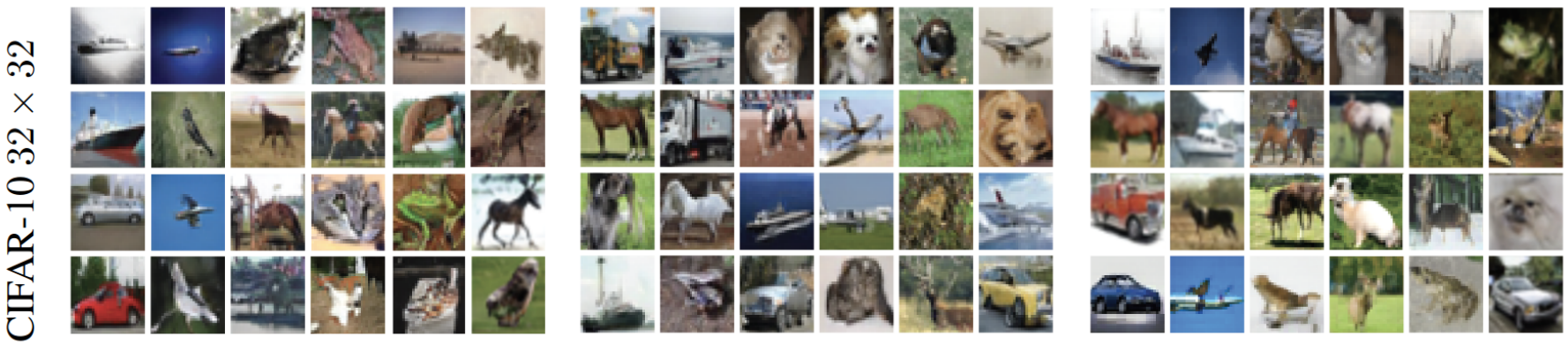
Experimentos

Datasets

Os datasets utilizados nos experimentos foram:

- CIFAR-10: benchmark padrão para geração de imagem, contendo 50K imagens de treino e 10K imgens de teste.
- CelebA: dataset de rostos humanos, contendo 162.770 imagens de teste e 18.962 imagens de treino.
- LSUN bedroom: contém em 3M de imagens de treino e 300 imagens de validação.

Comparação qualitativa no dataset CIFAR-10



(a) StyleGAN2 (FID = 11.1) (b) Vanilla-ViT (FID = 12.7) (c) ViTGAN (**FID = 6.66**)

Figura: Comparação qualitativa no dataset CIFAR-10 (Lee et al. [1]).

Comparação qualitativa no dataset CelebA

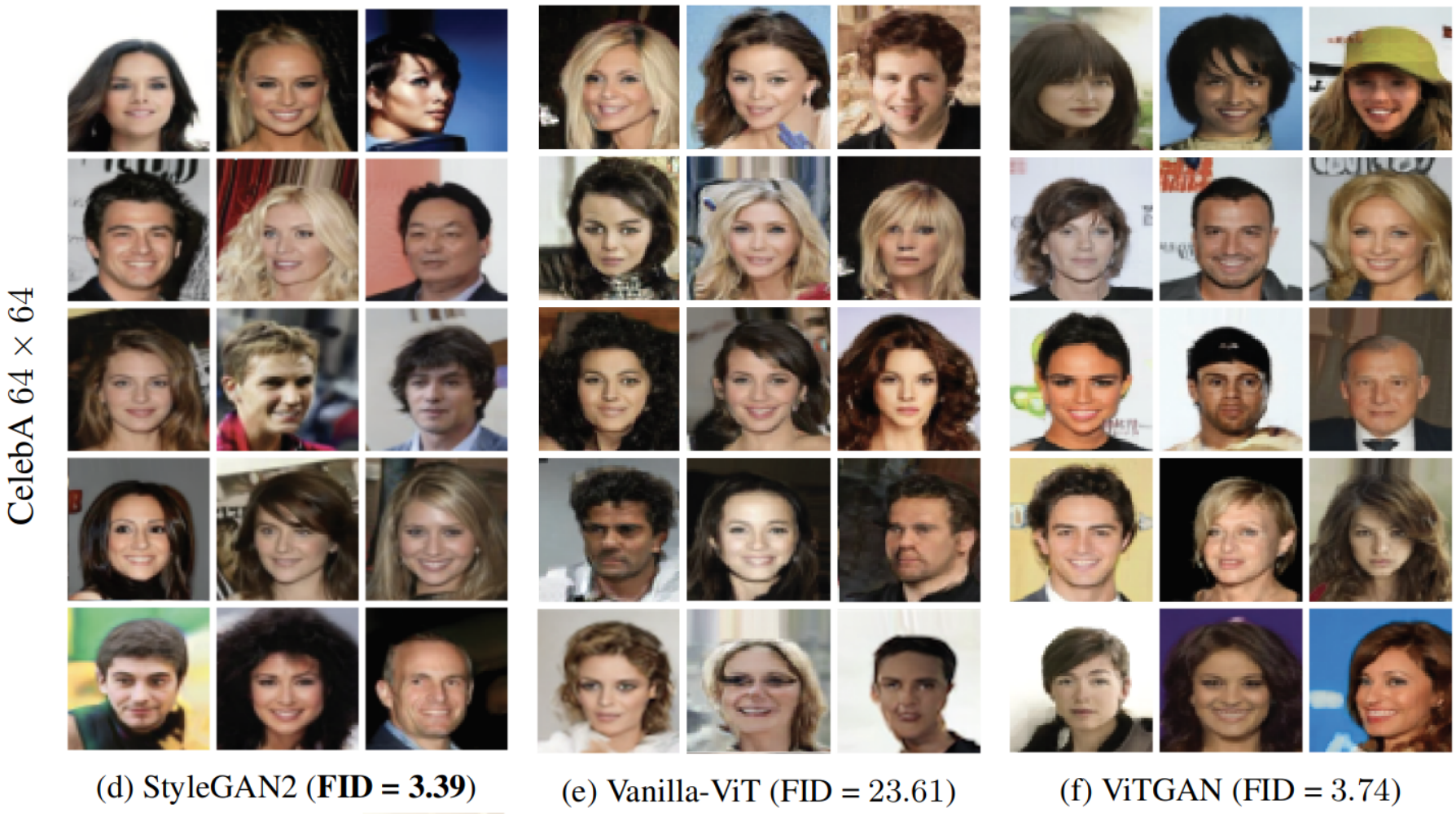


Figura: Comparação qualitativa no dataset CelebA (Lee et al. [1]).

Comparação qualitativa no dataset LSUN bedroom

Figura: Comparação qualitativa no dataset LSUN bedroom (Lee et al. [1]).

Resultados e Conclusões

O modelo ViTGAN alcança performance comparável com GANs de última geração baseados em CNN.

Architecture	Conv	Pool	CIFAR		CelebA		LSUN Bedroom	
			FID ↓	IS ↑	FID ↓	IS ↑	FID ↓	IS ↑
BigGAN [6] + DiffAug [59]	✓	✓	8.59*	9.25*	-	-	-	-
StyleGAN2 [27]	✓	✓	11.1*	9.18*	**3.39**	**3.43**	3.25	**2.45**
TransGAN-XL [23]	✗	✓	11.9*	8.63*	-	-	-	-
Vanilla-ViT	✗	✗	12.7	8.40	20.2	2.57	218.1	2.20
ViTGAN (Ours)	✗	✗	**6.66**	**9.30**	3.74	3.21	**2.65**	2.36

Figura: Comparação com arquiteturas GAN representativas de diferentes benchmarks de geração de imagens incondicionais. (Lee et al. [1]).

Entretanto, ainda não consegue superar o melhor modelo GAN baseado em CNN disponível com técnicas sofisticadas. Sugere-se, para melhorar a performance, Implementar:

- discriminador com contrastive representation learning, para ter o treinamento de GANs com data augmentation mais fortes, sem aumentar a instabilidade, previnindo o overfitting no discriminador [5].

Referências I

[1] Kwonjoon Lee et al. *ViTGAN: Training GANs with Vision Transformers*. 9 de jul. de 2021. arXiv: 2107.04589[cs,eess]. URL: http://arxiv.org/abs/2107.04589 (acesso em 08/11/2023).

[2] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 3 de jun. de 2021. DOI: 10.48550/arXiv.2010.11929. arXiv: 2010.11929[cs]. URL: http://arxiv.org/abs/2010.11929 (acesso em 09/11/2023).

[3] Hyunjik Kim, George Papamakarios e Andriy Mnih. *The Lipschitz Constant of Self-Attention*. 9 de jun. de 2021. DOI: 10.48550/arXiv.2006.04710. arXiv: 2006.04710[cs,stat]. URL: http://arxiv.org/abs/2006.04710 (acesso em 12/11/2023).

[4] Ting Chen et al. *On Self Modulation for Generative Adversarial Networks*. 2019. arXiv: 1810.01365 [cs.LG].

[5] Jongheon Jeong e Jinwoo Shin. *Training GANs with Stronger Augmentations via Contrastive Discriminator*. 2021. arXiv: 2103.09742 [cs.LG].

[6] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 10 de jun. de 2014. DOI: 10.48550/arXiv.1406.2661. arXiv: 1406.2661[cs,stat]. URL: http://arxiv.org/abs/1406.2661 (acesso em 09/11/2023).

Referências II

[7] Ashish Vaswani et al. *Attention Is All You Need*. 1 de ago. de 2023. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762[cs]. URL: http://arxiv.org/abs/1706.03762 (acesso em 09/11/2023).

[8] Edgar Schönfeld, Bernt Schiele e Anna Khoreva. "A U-Net Based Discriminator for Generative Adversarial Networks". Em: *CoRR* abs/2002.12655 (2020). arXiv: 2002.12655. URL: https://arxiv.org/abs/2002.12655.

[9] Yihe Dong, Jean-Baptiste Cordonnier e Andreas Loukas. *Attention is Not All You Need: Pure Attention Loses Rank Doubly Exponentially with Depth*. 1 de ago. de 2023. DOI: 10.48550/arXiv.2103.03404. arXiv: 2103.03404[cs]. URL: http://arxiv.org/abs/2103.03404 (acesso em 13/11/2023).