Visual Tranformers & GANs

L. Le Boudec, N. Olivain, P. Liautaud

- Sorbonne University -

Stabilization Problem

GANs are a SOTA technique for generation tasks, which are based on CNN and studied for the past years (stability, performances). Recently, attention mechanisms and (visual) transformers have shown great performances on several classical tasks, but they are showing some difficulties to be adapted on classical GANs architectures. The goal of the paper is to design a new appropriate regularization swiping out unstable training with visual transformers GANs.

Model & Basics

Vision Transformers (ViTs)

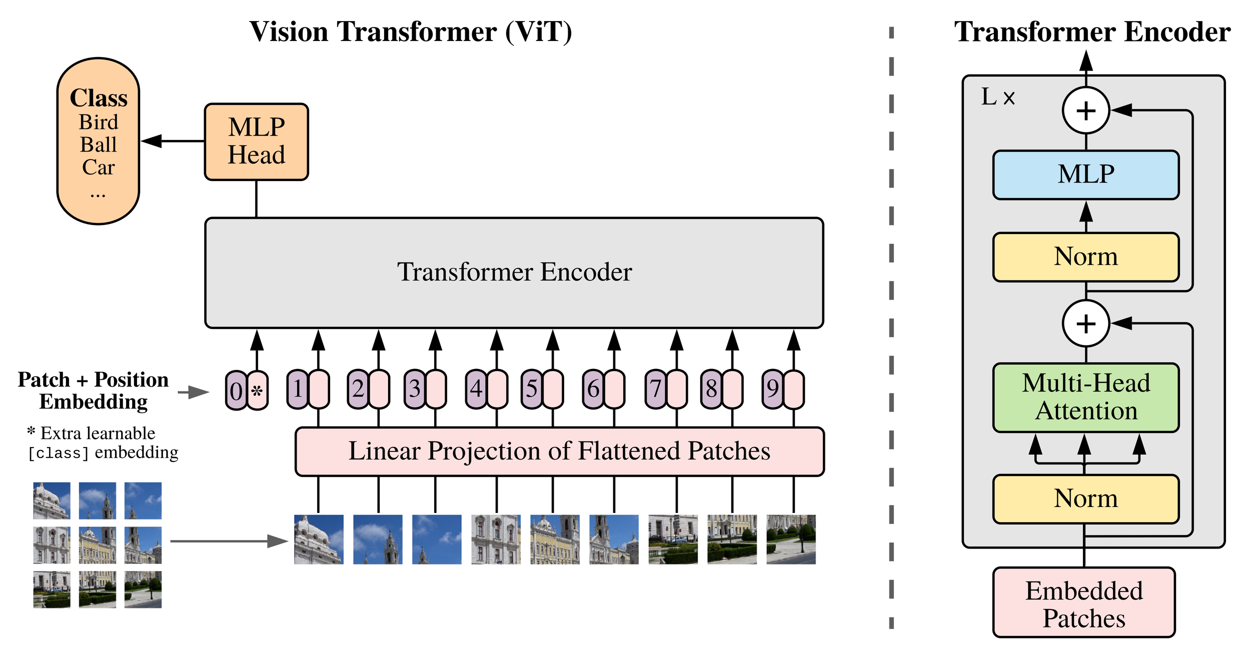

Fig. 1: ViT overview.

❶ Split an image into patches 2D image $\mathbf{x}$ of size $H \times W$, on C channels flattened into a sequence $\mathbf{x}_p$ of image patches:

$$\mathbf{x} \in \mathbb{R}^{H\times W\times C} \longrightarrow \mathbf{x}_p \in \left(\mathbb{R}^{P\times P\times C}\right)^N, P^2 \times C \text{ dim. of the } N = (H\times W)/P^2 \text{ patches.}$$

❷ Flatten patches $\to$ lower-dimensional linear embeddings of constant size d + positional info

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

$\mathbf{x}_{\text{class}} = *$ in Fig. 1

❸ Transformer encoder : Layernorm, Attention & MLP block

$$\begin{aligned} \mathbf{h}_{n+1/2} &= \mathrm{MSA}(\mathrm{LN}(\mathbf{h}_{n-1})) + \mathbf{h}_{n-1}; && 1 \le n \le L \\ \mathbf{h}_{n+1} &= \mathrm{MLP}(\mathrm{LN}(\mathbf{h}_{n+1/2}) + \mathbf{h}_{n+1/2}; && 1 \le n \le L \\ \mathbf{y} &= \mathrm{LN}(\mathbf{x}_{\text{class},L}); && \mathbf{x}_{\text{class},L} = \mathbf{h}_L^{(0)} \in \mathbb{R}^d \end{aligned}$$

Self-Attention

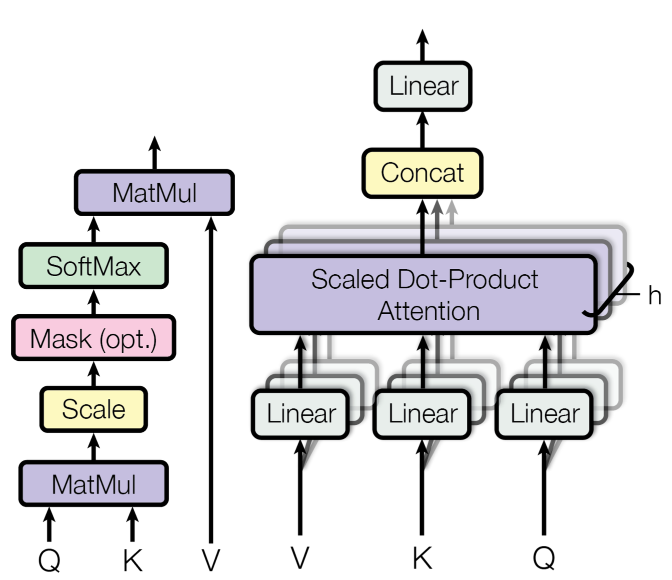

Fig. 2: Attention : Scaled Dot-Product (left) & Multi-Head (right).

Given 3 learnable matrices $\mathbf{W}_q$ (query), $\mathbf{W}_k$ (key), $\mathbf{W}_v$ (value), standard dot-product self-attention is:

$$\boxed{\mathrm{Attention}_h(\mathbf{X}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_h}}\right)\mathbf{V}}$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}_q, \mathbf{K} = \mathbf{X}\mathbf{W}_k$ and $\mathbf{V} = \mathbf{X}\mathbf{W}_v$.
`dot=torch.einsum("bid, bjd -> bij",q,k)`

Multi-headed self-attention (MSA) aggregates $H \ge 1$ single self-attention:

$$\boxed{\mathrm{MSA}(\mathbf{X}) = \mathrm{concat}_{h=1}^{H}[\mathrm{Attention}_h(\mathbf{X})]\mathbf{W} + \mathbf{b}}$$

with $\mathbf{W}$, $\mathbf{b}$ learnable parameters in the last linear projection.

GAN paradigm

Generative Adversarial Network includes a Generator G and a Discriminator D whose goals are:

$$\boxed{\max_D \min_G \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))]}$$

ViTGAN

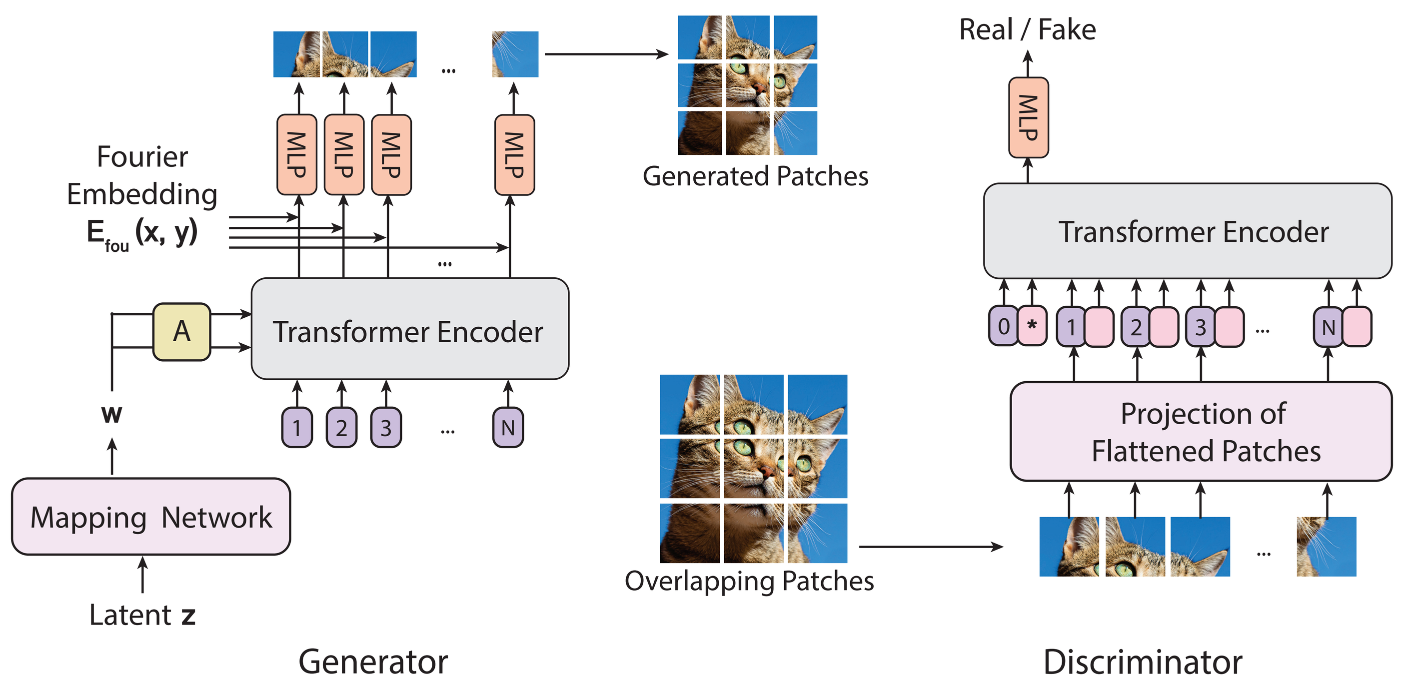

Fig. 3: ViTGan framework.

Regularization on ViT Discriminator

(Fig. 3, right)

It was proved that a Lipschitz Discriminator guarantees optimality in discriminative function and unicity in Nash equilibrium. 2 methods have been introduced in [4] to strengthen "Lipschitzianity":

⋆ **New Attention: from $\langle\cdot,\cdot\rangle$ to $\|\cdot\|_{\ell_2}$**

$$\boxed{\mathrm{Attention}_h(\mathbf{X}) = \mathrm{Softmax}\left(\frac{\|\mathbf{X}\mathbf{W}_q - \mathbf{X}\mathbf{W}_k\|_{\ell_2}}{\sqrt{d_h}}\right)\mathbf{X}\mathbf{W}_v}$$

⋆ **Improved Spectral Normalization (ISN)**

$$\boxed{\bar{\mathbf{W}} = \frac{\lambda_{\max}\left(\mathbf{W}^{(0)}\right)}{\lambda_{\max}(\mathbf{W})}\mathbf{W}}$$

⋆ **Overlap in Image Patches**

Including overlap $o \in \mathbb{N}^*$ slightly prevents D from memorizing local cues and provides meaningful loss for G. Extension of each border edge of a patch will lead to a patch size $(P + 2o)$ and the following sequence :

$$\mathbf{x}_p \in \left(\mathbb{R}^{(P+2o)^2\times C}\right)^N$$

New Generator Architecture

(Fig. 3, left)

⋆ **Principle**

Tansformer encoder architecture is mainly based on Fig. 1 (right) with a few changes. For $\mathbf{z}$ Gaussian noise:

$$\begin{aligned} \mathbf{h}_0 &= \mathbf{E}_{\text{pos}}; && \mathbf{E}_{\text{pos}} \in \mathbb{R}^{N\times d} \text{ positional emb.} \\ \mathbf{h}_{n+1/2} &= \mathrm{MSA}(\mathrm{SLN}(\mathbf{h}_n, \mathbf{w})) + \mathbf{h}_n; && 1 \le n \le L, \mathbf{w} = \mathrm{MLP}(\mathbf{z}) \in \mathbb{R}^d \\ \mathbf{h}_{n+1} &= \mathrm{SLN}(\mathbf{h}_{n+1/2}, \mathbf{w}) + \mathbf{h}_{n+1/2}; && 1 \le n \le L \\ \mathbf{y} &= \left[\mathbf{y}^{(1)}; \cdots; \mathbf{y}^{(N)}\right] = \mathrm{SLN}(\mathbf{h}_L, w); && \mathbf{y}^{(i)} \in \mathbb{R}^d \\ \mathbf{x} &= \left[\mathbf{x}_p^{(1)}; \cdots; \mathbf{y}^{(N)}\right] = \left[f_\theta\left(\mathbf{E}_{\text{sir}}, \mathbf{y}^{(1)}\right); \cdots; f_\theta\left(\mathbf{E}_{\text{sir}}, \mathbf{y}^{(N)}\right)\right] && \mathbf{x}_p^{(i)} \in \mathbb{R}^{P^2\times C}, \mathbf{x} \in \mathbb{R}^{H\times W\times C} \end{aligned}$$

⋆ **Self-modulated LayerNorm** SLN

Denoted as `A` in Fig. 3 , it uses noise input $\mathbf{z}$ to modulate the normalization LN in ❸, for each step n:

$$\boxed{\mathbf{w} = \mathrm{MLP}(\mathbf{z}) \in \mathbb{R}^d; \mathbf{h}_n \mapsto \mathrm{SLN}(\mathbf{h}, \mathbf{w}) = \gamma_n(\mathbf{w}) \odot \frac{\mathbf{h}_n - \mu}{\sigma} + \beta_n(\mathbf{w})}$$

where $\gamma_n(\mathbf{w}), \beta_n(\mathbf{w})$ are learnable parameters.

⋆ **From Implicit Neural Representation to patch pixel**

Implicit Neural Representation allows to learn continuous mapping : $\mathbf{y}^{(i)} \in \mathbb{R}^d \mapsto \mathbf{x}_p^{(i)}$. A key was to use `SIREN` *sinunoidal activation functions* $\mathbf{E}_{\text{sir}}$ (or *Fourier features* $\mathbf{E}_{\text{fou}}$ in Fig. 3) coupled with implicit representations $\mathbf{y}^{(i)}$. Concretely, patch pixel i is computed as:

$$\boxed{\mathbf{x}_p^{(i)} = f_\theta\left(\mathbf{E}_{\text{sir}}, \mathbf{y}^{(i)}\right)}$$

with $f_\theta(\mathbf{E}_{\text{sir}}, \cdot)$ a 2-`SIREN`-layer MLP: `SIREN(input) = torch.sin(constante * Linear(input))`

XP & Comparison

MNIST

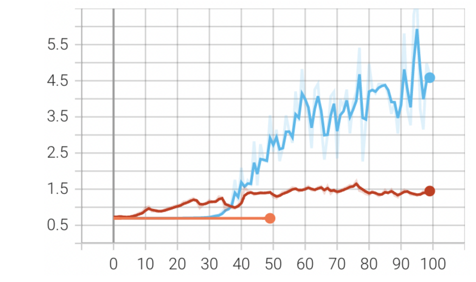

Fig. 4: Generator loss.

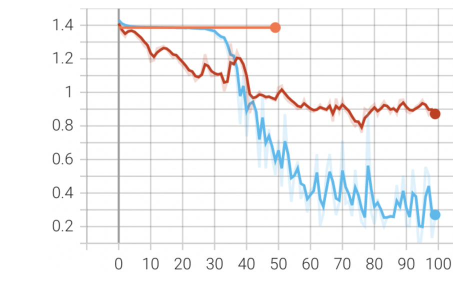

Fig. 5: Discriminator loss.

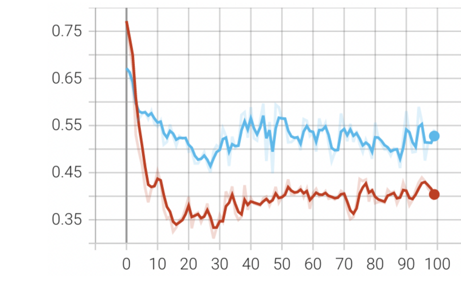

Fig. 6: Frechet Inception Distance (FID).

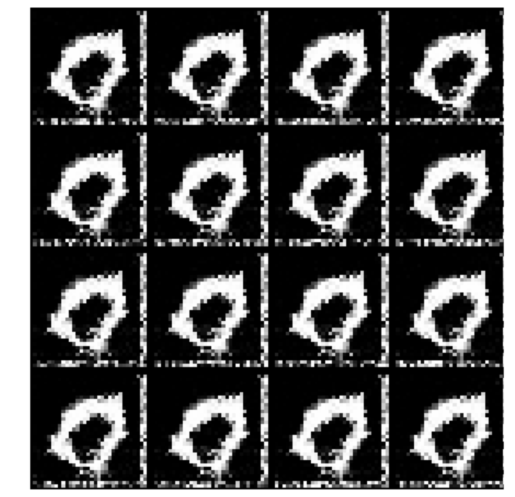

Fig. 7: Vanilla ViT fake samples.

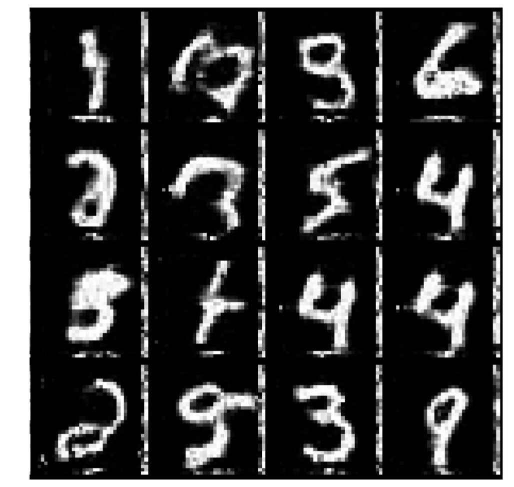

Fig. 8: ViTGAN fake samples.

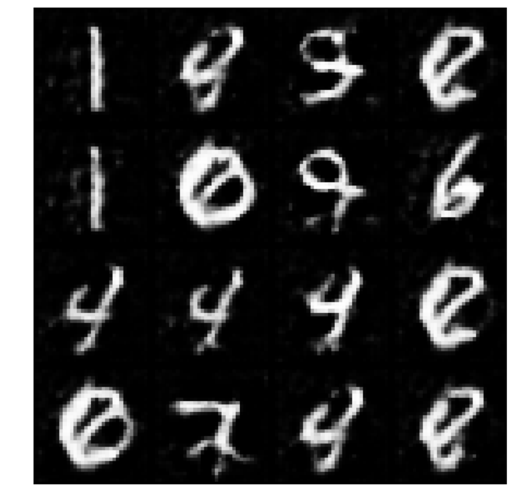

Fig. 9: Convolutional GAN fake samples.

Graphs' legend [Fig. 4 to 6] - `Red:` *fully regulated ViTGAN model ;* `Blue:` *Vanilla ViT without* SLN*, neither* L^2*-Att. nor Spectral Norm ;* `Orange:` ReLU *in place of* GELU *for* MLP *final activation.* 4 *blocks of* 4 *attention heads* $\approx 30\times 10^6$ *parameters trained over* 100 *epochs,* `lr`$= 2\times 10^{-5}$.

CelebA

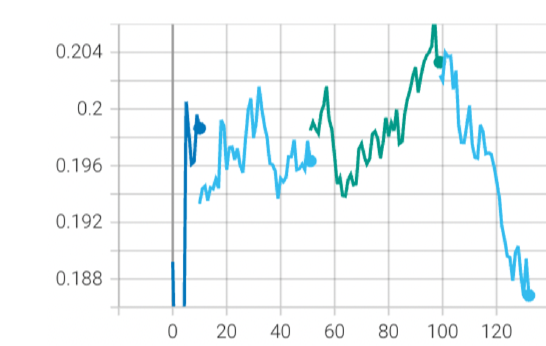

Fig. 10: Frechet Inception Distance (FID).

Fig. 11: ViTGAN fake samples.

Remarks

↪ FID measures difference between 2 data distribution featured by $(\mu_1, \Sigma_1), (\mu_2, \Sigma_2)$ as:

$$\mathrm{FID} = |\mu_1 - \mu_2| + \mathrm{Tr}\left(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2}\right)$$

↪ Position embeddings added to patch embeddings are 1D standard variable since no significant performance gains are observed from using 2D position embeddings [2] ;

↪ ReLU vs GELU non-linearity : a gradient vanishing tradeoff ;

↪ Number of patches dealing with the Discriminator's transformer can be increased to get better performances (do not need to do so with Generator's transformer) [4] ;

↪ Setting overlap $o = P/2$ could be seen as a convolution operation with kernel $(P + 2o)^2$ and stride $P \times P$. Increasing sequence length of feature dimension on D is sufficient when scaling on high resolution images.

References

[1] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: `1810.04805 [cs.CL]`.

[2] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: `2010.11929 [cs.CV]`.

[3] Hyunjik Kim, George Papamakarios, and Andriy Mnih. *The Lipschitz Constant of Self-Attention*. 2021. arXiv: `2006.04710 [stat.ML]`.

[4] Kwonjoon Lee et al. *ViTGAN: Training GANs with Vision Transformers*. 2021. arXiv: `2107.04589 [cs.CV]`.

[5] Vincent Sitzmann et al. *Implicit Neural Representations with Periodic Activation Functions*. 2020. arXiv: `2006.09661 [cs.CV]`.

[6] Matthew Tancik et al. *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*. 2020. arXiv: `2006.10739 [cs.CV]`.

[7] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: `1706.03762 [cs.CL]`.