A NEURAL TANGENT KERNEL PERSPECTIVE OF GANS

GdR ISIS – Théorie du deep learning

June 28th, 2021

Jean-Yves Franceschi,[1] **Emmanuel de Bézenac**,[1] **Ibrahim Ayed**,[1,2] Mickaël Chen,[3] Sylvain Lamprier,[1] Patrick Gallinari[1,4]

[1]Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
[2]ThereSIS Lab, Thales, Palaiseau, France [3]Valeo.ai, Paris, France
[4]Criteo AI Lab, Paris, France

Outline

- Novel general framework of analysis of GANs taking into account discriminator architecture, unlike prior work.
- Based on the theory of infinite-width neural networks.
- Closes theoretical gaps to better understand GAN training.
- Allows experimental studies of various GAN settings.
- Highlights the interesting properties of neural tangent kernels for generative modeling.

Preprint

`https://arxiv.org/abs/2106.05566`

Principle

Generative model with a generator g competing with a discriminator f trained to distinguish between fake (α_g) and target (β) samples.

Principle

Generative model with a generator g competing with a discriminator f trained to distinguish between fake (α_g) and target (β) samples.

Examples:

- Vanilla GAN (Goodfellow et al., 2014):

$$\min_g \max_f \quad \mathsf{E}_{x \quad \alpha_g} \left[\log \Big(1 \quad \sigma\big(f(x)\big) \Big) \right] + \mathsf{E}_{y \quad \beta} \Big[\log \sigma\big(f(y)\big) \Big].$$

Principle

Generative model with a generator g competing with a discriminator f trained to distinguish between fake (α_g) and target (β) samples.

Examples:

- Vanilla GAN (Goodfellow et al., 2014):

$$\min_g \max_f \quad \mathrm{E}_{x \sim \alpha_g}\Big[\log\Big(1 - \sigma\big(f(x)\big)\Big)\Big] + \mathrm{E}_{y \sim \beta}\Big[\log \sigma\big(f(y)\big)\Big].$$

- IPM, e.g. WGAN (Arjovsky et al., 2017):

$$\min_g \max_f \quad \mathrm{E}_{x \sim \alpha_g}\big[f(x)\big] - \mathrm{E}_{y \sim \beta}\big[f(y)\big].$$

Setting

Generator $g \in G : \mathbb{R}^d \to \mathbb{R}^n$ optimizing:

$$\inf_{g \in G} \left\{ C_{f^\star_{\alpha_g}}(\alpha_g) \ , \ \mathbb{E}_{x \sim \alpha_g} \left[\left(c \circ f^\star_{\alpha_g} \right)(x) \right] \right\},$$

with the discriminator $f^\star_{\alpha_g} \in F : \mathbb{R}^n \to \mathbb{R}$ optimizing:

$$\sup_{f \in F} \left\{ L_{\alpha_g}(f) \ , \ \mathbb{E}_{x \sim \alpha_g} \left[(a \circ f)(x) \right] - \mathbb{E}_{y \sim \beta} \left[(b \circ f)(y) \right] \right\}.$$

Includes many GAN formulations, e.g. WGAN, with $a = b = c = \mathrm{id}$:

$$\min_g \max_f \ \mathbb{E}_{x \sim \alpha_g} \left[f(x) \right] - \mathbb{E}_{y \sim \beta} \left[f(y) \right].$$

GAN: Practice

- G and F are families of neural networks.

- G and F are families of neural networks.
- Generator $g = g_\theta$ optimization:

$$\theta_{j+1} = \theta_j \quad \eta \mathbb{E}_{z \sim p_z} \left[\nabla_\theta g_{\theta_j}(z)^T \nabla_x \left(c(\cdot, g_{\theta_j}(z)) - f^\star_{\alpha_g}(\cdot) \right)(x) \Big|_{x = g_{\theta_j}(z)} \right].$$

$\nabla f^\star_{\alpha_g}$ is crucial for generator training.

- G and F are families of neural networks.
- Generator $g = g_\theta$ optimization:

$$\theta_{j+1} = \theta_j \quad \eta \mathbb{E}_{z \sim p_z} \left[\nabla_\theta g_{\theta_j}(z)^T \nabla_x \left(c \quad f^\star_{\alpha_g} \right)(x) \Big|_{x = g_{\theta_j}(z)} \right].$$

$\nabla f^\star_{\alpha_g}$ is crucial for generator training.

- The discriminator is trained at each step using a finite number of samples from α and β.

The training distribution $\hat{\gamma} = \frac{\hat{\alpha} + \hat{\beta}}{2}$ is a finite mixture of Diracs.

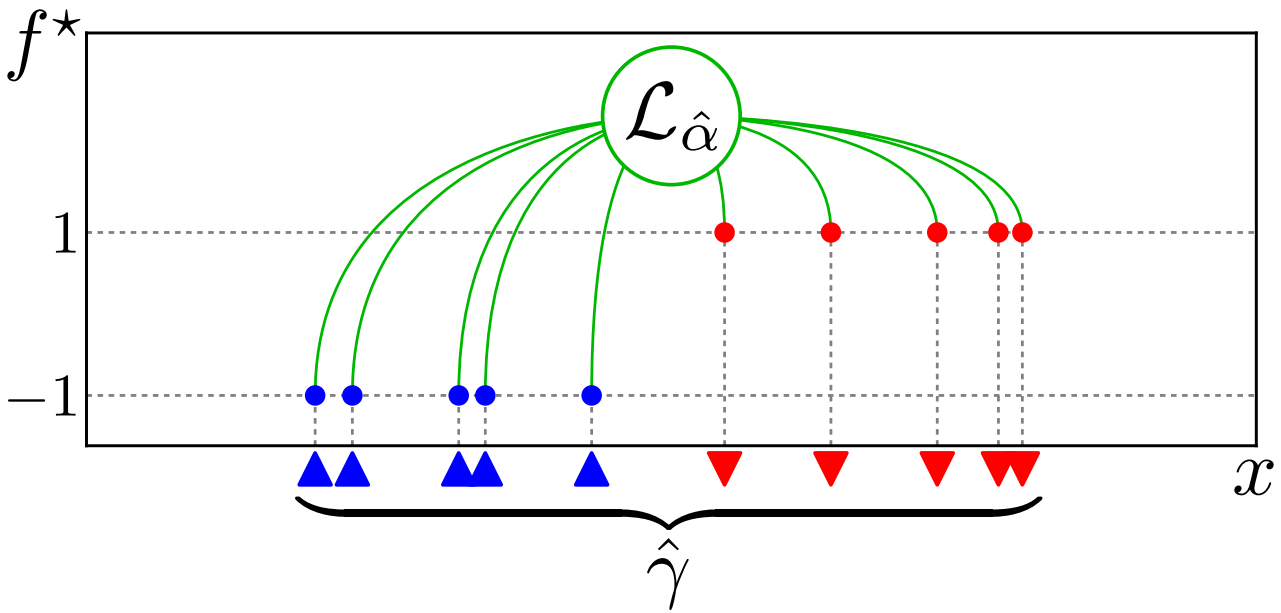
- Analysis by Arjovsky et al. (2017) of LSGAN (Mao et al., 2017):

$$L_{\hat{\alpha}}(f) = \mathbb{E}_{x \sim \hat{\alpha}}\left[\left(f(x) + 1\right)^2\right] \quad \mathbb{E}_{y \sim \hat{\beta}}\left[\left(f(y) \quad 1\right)^2\right].$$

- Analysis by Arjovsky et al. (2017) of LSGAN (Mao et al., 2017):

$$L_{\hat{\alpha}}(f) = \mathbb{E}_{x \sim \hat{\alpha}}\left[(f(x) + 1)^2\right] \quad \mathbb{E}_{y \sim \hat{\beta}}\left[(f(y) \quad 1)^2\right].$$

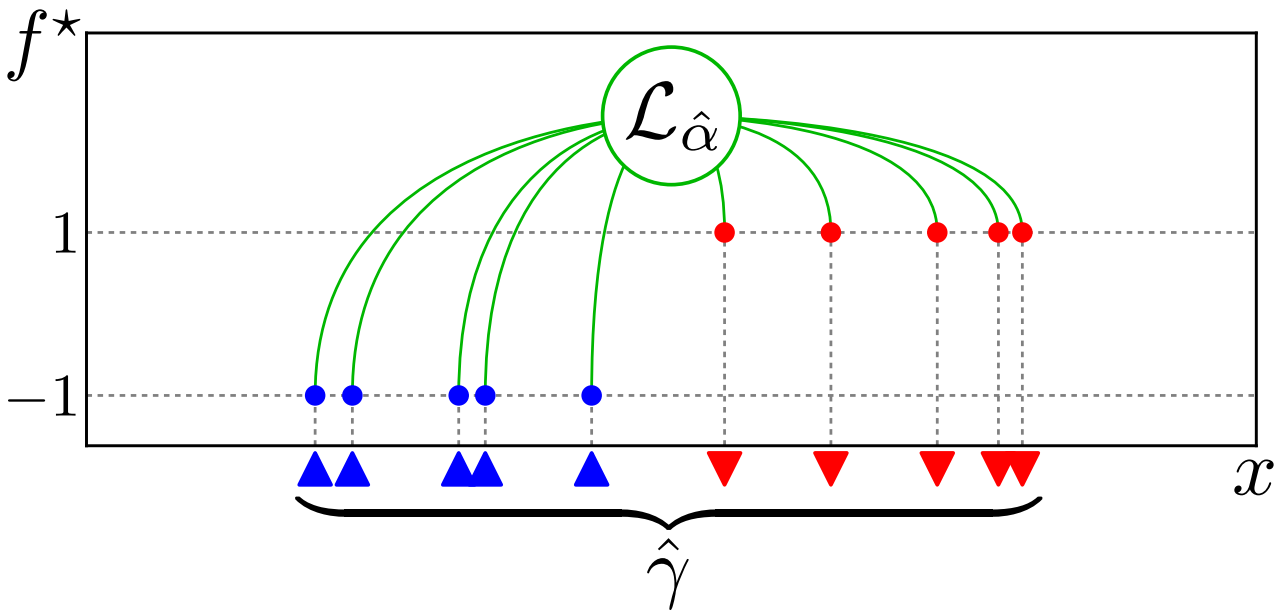
- Considering $f \in L^2(\hat{\gamma})$, the optimal discriminator $f^\star$ has values independent from the generated distribution.



- Analysis by Arjovsky et al. (2017) of LSGAN (Mao et al., 2017):

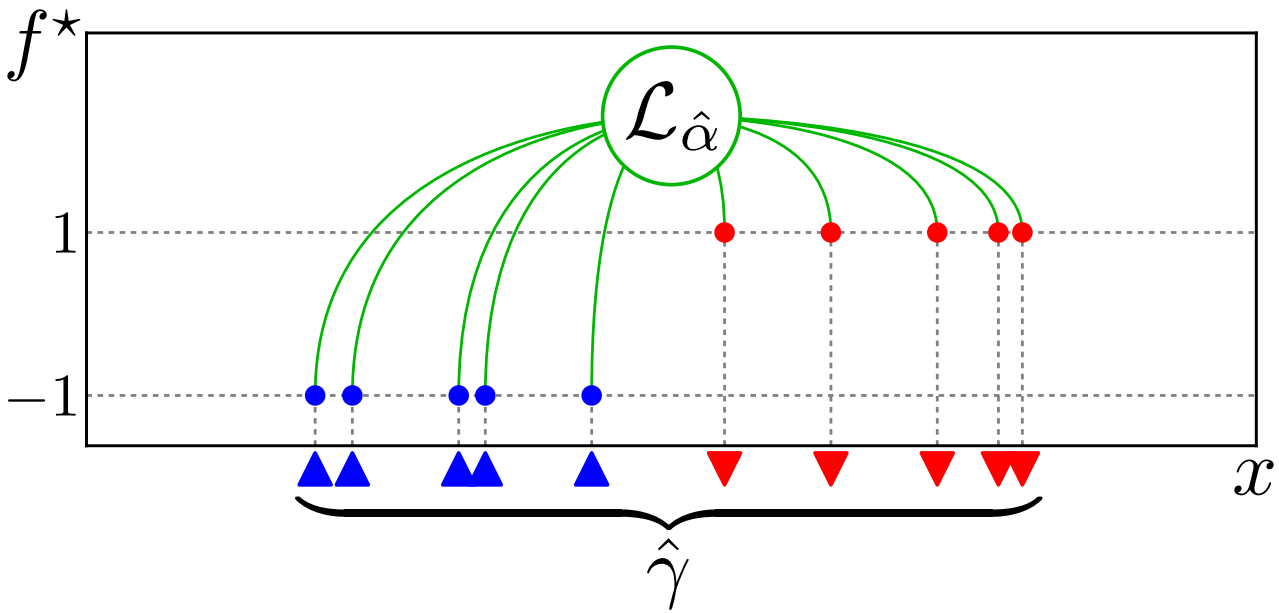
$$L_{\hat{\alpha}}(f) = \mathbb{E}_{x \sim \hat{\alpha}}\left[(f(x) + 1)^2\right] \quad \mathbb{E}_{y \sim \hat{\beta}}\left[(f(y) \quad 1)^2\right].$$

- Considering $f \in L^2(\hat{\gamma})$, the optimal discriminator $f^\star$ has values independent from the generated distribution.
- Conclusion: the generator loss is constant and the generator receives null gradients.



Problem

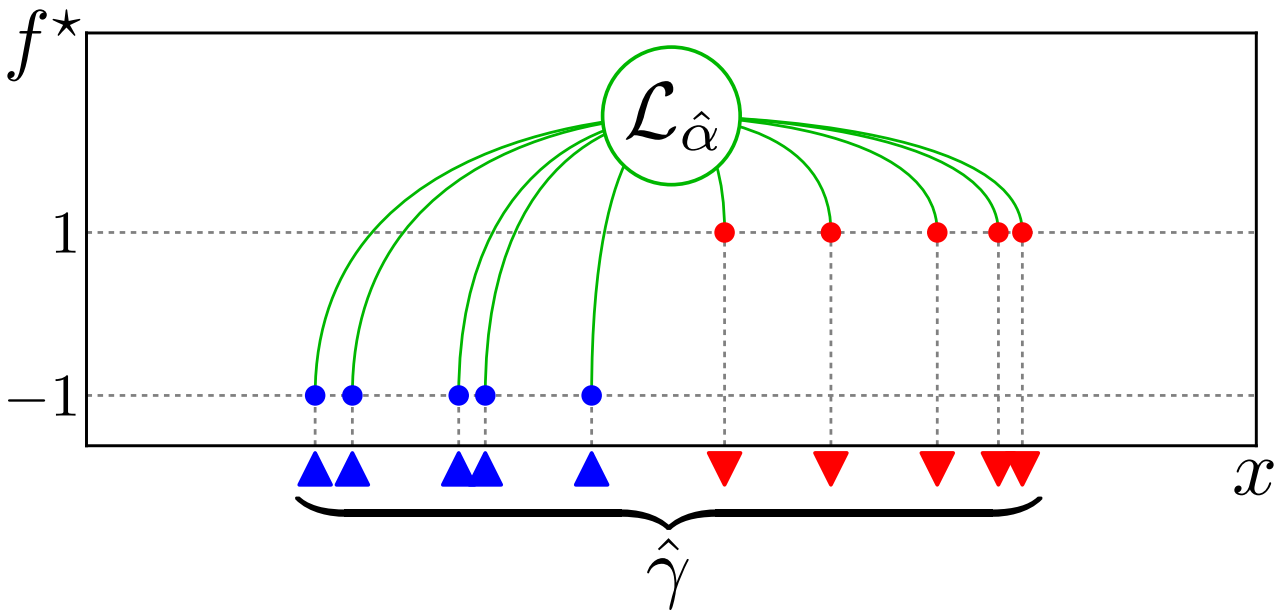
In practice, LSGAN and vanilla GAN are trainable.



Problem

In practice, LSGAN and vanilla GAN are trainable.

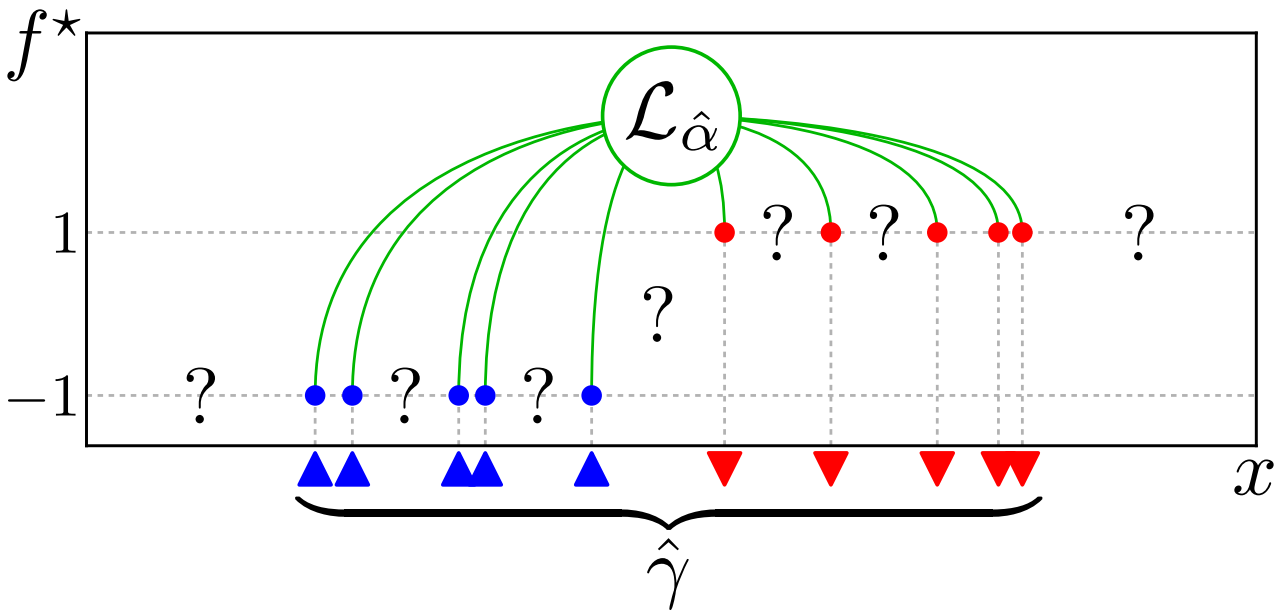
- When updating the generator, the discriminator is assumed fixed. What about $r\ f^\star$?



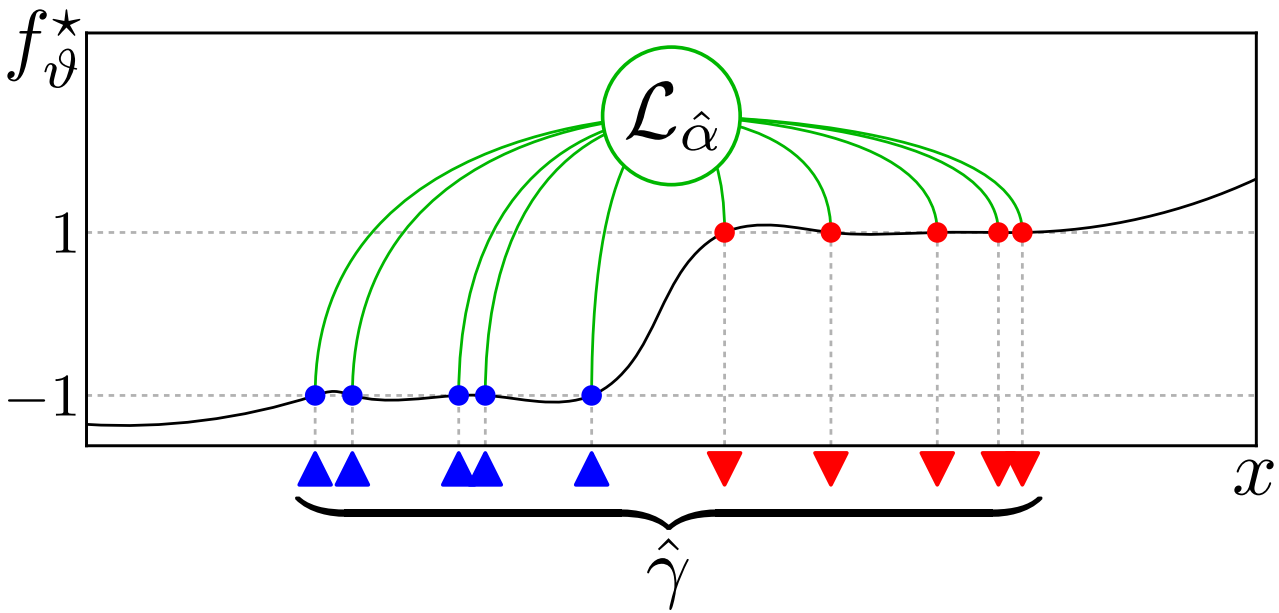
Problem

In practice, LSGAN and vanilla GAN are trainable.

- When updating the generator, the discriminator is assumed fixed. What about $r\, f^\star$?
- Considering $f \in L^2(\hat{\gamma})$, $r\, f^\star$ is undefined as $f^\star$ is only defined over $\operatorname{supp} \hat{\gamma}$.



- Considering $f = f_\vartheta$ as a neural network of a given architecture trained by gradient ascent to maximize $L_{\hat{\alpha}}$.
- With this modelization, the discriminator is defined everywhere.
- In our framework, we model this parameterization in the infinite-width regime using neural tangent kernels (NTKs).



NTK of the Discriminator

- Generally applicable proxy representing the learning capacity of the architecture of f.
- Makes neural networks act like kernel methods and simplifies their theoretical study.

NTK of the Discriminator

- Generally applicable proxy representing the learning capacity of the architecture of f.
- Makes neural networks act like kernel methods and simplifies their theoretical study.

NTK of f_ϑ (Jacot et al., 2018)

$$k_{f_\vartheta}(x, y) \quad , \quad \left\langle \partial_\vartheta f_\vartheta(x), \partial_\vartheta f_\vartheta(y) \right\rangle.$$

NTK of the Discriminator

- Generally applicable proxy representing the learning capacity of the architecture of f.
- Makes neural networks act like kernel methods and simplifies their theoretical study.

NTK of f_ϑ (Jacot et al., 2018)

$$k_{f_\vartheta}(x, y) \quad , \quad \left\langle \partial_\vartheta f_\vartheta(x), \partial_\vartheta f_\vartheta(y) \right\rangle.$$

Constant NTK (Jacot et al., 2018)

In the infinite-width limit of f, during training, $k_{f_\vartheta}(x, y) = k(x, y)$.

Kernel Integral Operator and RKHS

$$T_{k,\hat{\gamma}}: L^2(\hat{\gamma}) \;!\; H_k^{\hat{\gamma}},\; h \;7!\; \int_x k(\,,x)h(x)\,\mathrm{d}\hat{\gamma}(x),$$

where $H_k^{\hat{\gamma}}$ $L^2(\quad)$ is the RKHS of k generated by $\hat{\gamma}$.

Discriminator Training

$$\partial_t f_t = T_{k,\hat{\gamma}}\Big(r^{\hat{\gamma}} L_{\hat{\alpha}}(f_t)\Big).$$

Kernel Integral Operator and RKHS

$$T_{k,\hat{\gamma}}: L^2(\hat{\gamma}) \;!\; H_k^{\hat{\gamma}},\; h \;7!\; \int_x k(\,,x)h(x)\,\mathrm{d}\hat{\gamma}(x),$$

where $H_k^{\hat{\gamma}}$ $L^2(\quad)$ is the RKHS of k generated by $\hat{\gamma}$.

Discriminator Training

$$\partial_t f_t = T_{k,\hat{\gamma}}\Big(r^{\hat{\gamma}} L_{\hat{\alpha}}(f_t)\Big).$$

In short:

- Simple expression of the evolution of f during training.
- Confers strong regularity to f (corresponding to its parameterization as a neural network).

The following results hold under mild assumptions on the architecture of f and L_{α}.

Theorem

f_t is uniquely defined from the previous ODE, and:

$$\forall t \geq 0, f_t = f_0 + T_{k,\gamma}\left(\int_0^t r^{\gamma} L_{\alpha}(f_s)\,\mathrm{d}s\right) - f_0 + H_k^{\gamma}.$$

The following results hold under mild assumptions on the architecture of f and L_{α}.

Theorem

f_t is uniquely defined from the previous ODE, and:

$$\forall t \geq 0, f_t = f_0 + T_{k,\gamma}\left(\int_0^t \gamma L_{\alpha}(f_s)\,\mathrm{d}s\right) - f_0 + H_k^{\gamma}.$$

Theorem

The NTK k of f and f_t are infinitely differentiable (almost) everywhere.

Case Study: IPM loss

$$L_{\hat{\alpha}}(f) = \mathsf{E}_{x \sim \hat{\alpha}}\big[f(x)\big] - \mathsf{E}_{y \sim \hat{\beta}}\big[f(y)\big]$$

$$L_{\hat{\alpha}}(f) = \mathbb{E}_{x \sim \hat{\alpha}}\big[f(x)\big] - \mathbb{E}_{y \sim \hat{\beta}}\big[f(y)\big]$$

IPM Discriminator and MMD

In the infinite-width regime:

$$\begin{aligned} f_t &= f_0 + t f_{\hat{\alpha}}, \\ f_{\hat{\alpha}} &= \mathbb{E}_{x \sim \hat{\alpha}}\big[k(x, \cdot)\big] - \mathbb{E}_{y \sim \hat{\beta}}\big[k(y, \cdot)\big], \\ L_{\hat{\alpha}}(f_t) &= L_{\hat{\alpha}}(f_0) + t \ \mathrm{MMD}_k^2\left(\hat{\alpha}, \hat{\beta}\right). \end{aligned}$$

$$L_{\hat{\alpha}}(f) = \mathbb{E}_{x \sim \hat{\alpha}}\big[f(x)\big] - \mathbb{E}_{y \sim \hat{\beta}}\big[f(y)\big]$$

IPM Discriminator and MMD

In the infinite-width regime:

$$\begin{aligned} f_t &= f_0 + t f_{\hat{\alpha}}, \\ f_{\hat{\alpha}} &= \mathbb{E}_{x \sim \hat{\alpha}}\big[k(x, \cdot)\big] - \mathbb{E}_{y \sim \hat{\beta}}\big[k(y, \cdot)\big], \\ L_{\hat{\alpha}}(f_t) &= L_{\hat{\alpha}}(f_0) + t \ \mathrm{MMD}_k^2\left(\hat{\alpha}, \hat{\beta}\right). \end{aligned}$$

Consequence

The generator minimizes the MMD, defined with the NTK of f, between fake and target distributions (thanks to Arbel et al. (2019)).

$$L_{\hat{\alpha}}(f) = \mathbb{E}_{x \sim \hat{\alpha}}\left[(f(x) + 1)^2\right] \quad \mathbb{E}_{y \sim \hat{\beta}}\left[(f(y) \quad 1)^2\right]$$

LSGAN Discriminator

In the infinite-width regime:

$$f_t = \exp(\quad 4tT_{k,\hat{\gamma}})(f_0 \quad \rho) + \rho, \quad \rho(x) = \begin{cases} 1 & \text{if } x \in \text{supp}\,\hat{\alpha} \\ 1 & \text{if } x \in \text{supp}\,\hat{\beta} \end{cases}.$$

Consequence

f_1 is optimal when k is strictly positive definite ($f_1|_{\text{supp}\,\hat{\gamma}} = \rho$) but $T_{k,\hat{\gamma}}$ smoothes ρ on the whole domain.

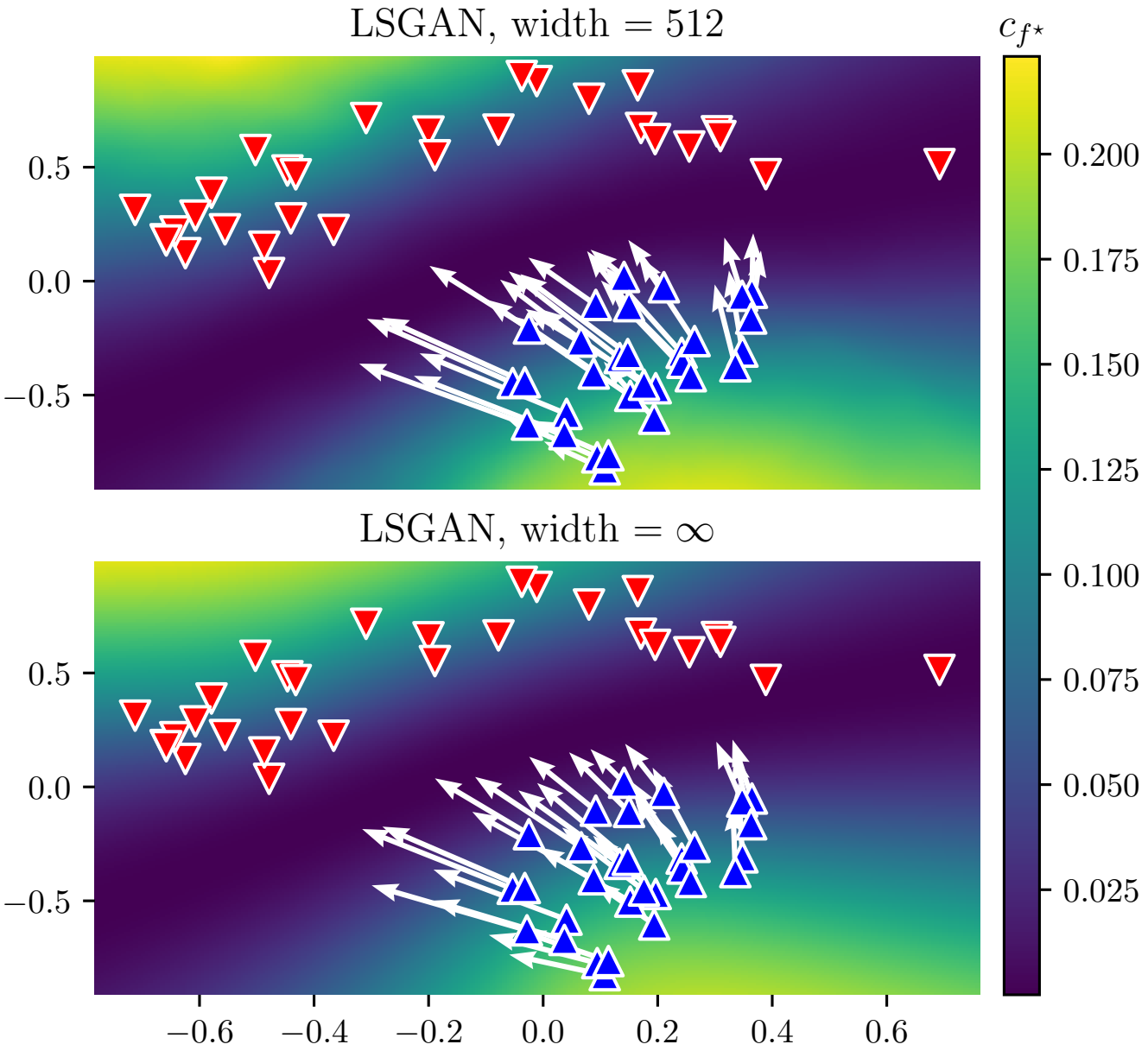
GAN Analysis Toolkit GAN(TK)^2

Code

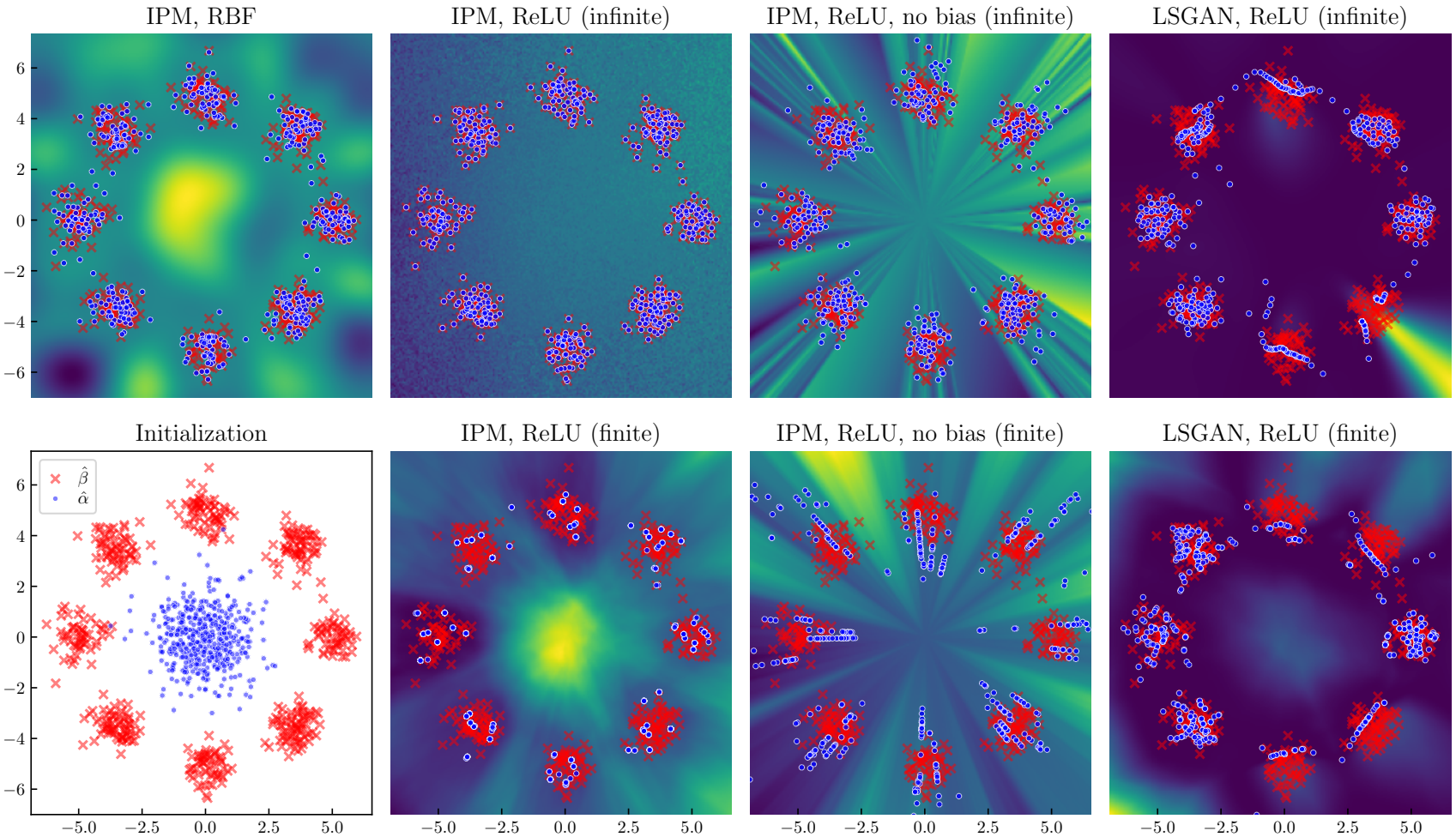
`https://github.com/emited/gantk2`

- Various finite- and infinite-width implementations of GANs.
- Based on the Neural Tangents library (Novak et al., 2020).
- Could be used for further experimental studies.

GAN Analysis Toolkit: Adequacy



GAN Analysis Toolkit: Convergence



References

Arbel, Michael et al. (2019). "Maximum Mean Discrepancy Gradient Flow". In: *Advances in Neural Information Processing Systems*. Ed. by Hanna Wallach et al. Vol. 32. Curran Associates, Inc., pp. 6481–6491.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (Aug. 2017). "Wasserstein Generative Adversarial Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 214–223.

Goodfellow, Ian et al. (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Zoubin Ghahramani et al. Vol. 27. Curran Associates, Inc., pp. 2672–2680.

Jacot, Arthur, Franck Gabriel, and Clement Hongler (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by Samy Bengio et al. Vol. 31. Curran Associates, Inc., pp. 8580–8589.

Mao, Xudong et al. (Oct. 2017). "Least Squares Generative Adversarial Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2813–2821.

Novak, Roman et al. (2020). "Neural Tangents: Fast and Easy Infinite Neural Networks in Python". In: *International Conference on Learning Representations*.