

# Transport optimal pour l'apprentissage de modèles génératifs

Antoine Houdard

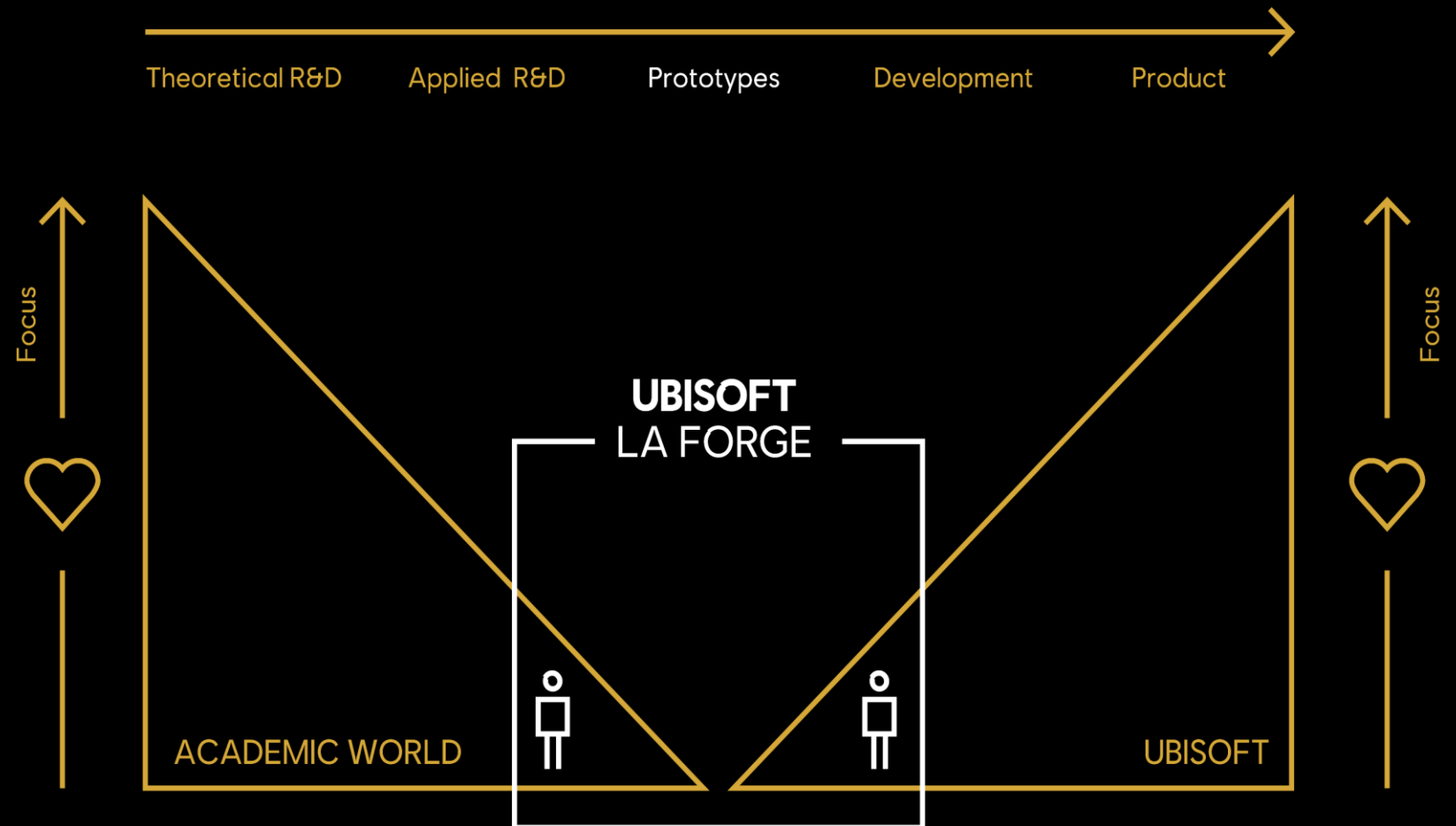
**UBISOFT**  
LA FORGE

Collaboration avec Arthur Leclaire\*, Nicolas Papadakis\* et Julien Rabin\*\*

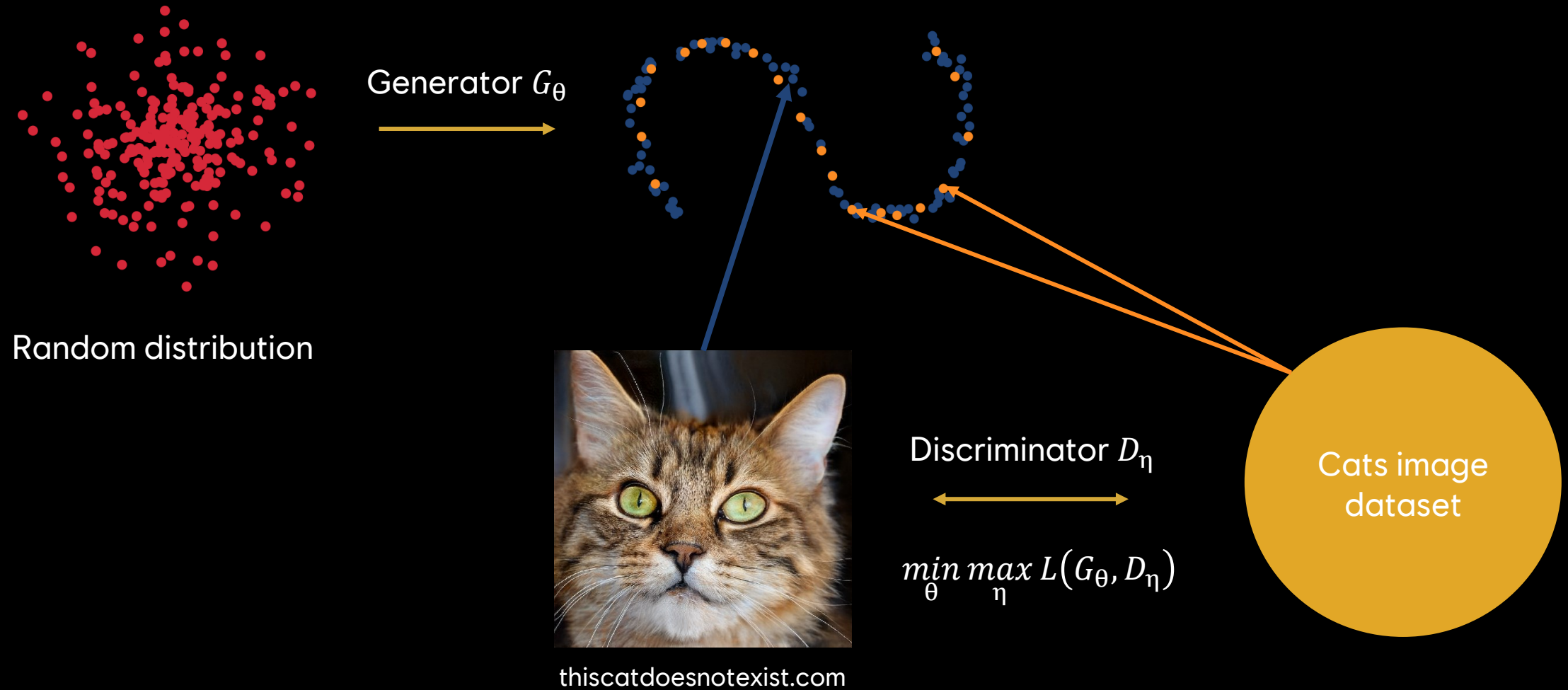


Depuis 2021 — chercheur Ubisoft La Forge → rendu 3D, traitement d'image, ML, super-resolution

2019 – 2021 — post-doc à l'IMB, Université de Bordeaux → transport optimal, modèles génératifs



# Context: Generative Adversarial Networks (GAN)

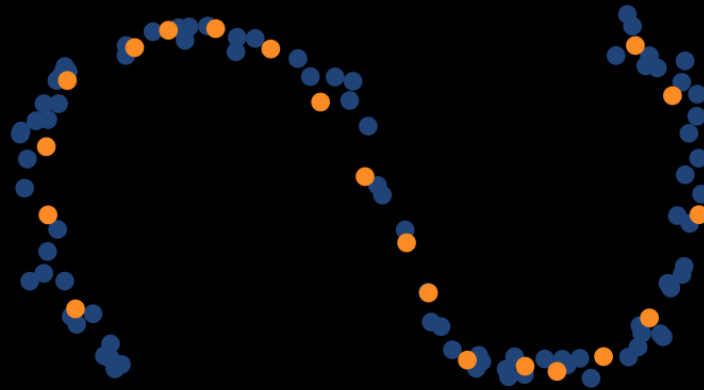


# Modèles génératifs point de vue statistique

- Données  $\{y_1, \dots, y_n\}$  échantillonnées suivant  $Y \sim \nu$
- Distribution image  $\mu_\theta = g_\theta \# \zeta$  définie par un modèle génératif



Échantillons de  $Z \sim \zeta$



Échantillons de  $g_\theta(Z) \sim g_\theta \# \zeta$

**Objectif** : trouver un estimateur  $\theta$  tel que  $\mu_\theta$  proche de  $\nu$



**Idée** : estimateur  $\theta$  tel que  $\mu_\theta$  proche de  $\nu$  au sens du **transport optimal**

$$\hat{\theta} = \min_{\theta} OT_c(\mu_\theta, \nu)$$

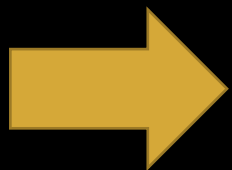
**Formulation semi-duale** du transport optimal

$$OT_c(\mu_\theta, \nu) = \max_{\psi} E_{X \sim \mu_\theta}[\psi^c(X)] + E_{Y \sim \nu}[\psi(Y)]$$

$$\text{Avec } \psi^c(x) = \min_y [c(x, y) - \psi(y)]$$

Avec  $c(x, y) = |x - y|$ , on a  $\psi^c = -\psi$  et on retrouve la formulation WGAN [Arjovsky et al. '17]

Dans ce cas, la variable duale  $\psi$  est encodée par un réseau de neurone



Cas général pour le coût ? Sans réseau de neurone pour  $\psi$  ?



# Apprendre un modèle génératif en minimisant le cout de transport

- On veut minimiser selon  $\theta$  la quantité

$$W(\theta) = \text{OT}_c(g_\theta \# \zeta, \nu) = \max_{\psi} E_{Z \sim \zeta}[\psi^c(g_\theta(Z))] + E_{Y \sim \nu}[\psi(Y)]$$

- La distribution  $\nu$  est connue
- On peut échantillonner  $\mu_\theta = g_\theta \# \zeta$



Peut-on calculer un gradient stochastique de  $W(\theta)$  ?

## Proposition :

Sous hypothèses de régularité de  $g_\theta$  et  $c$ . Si  $\psi_0^*$  est un potentiel optimal pour  $\theta_0$  alors

$$\nabla W(\theta_0) = E_{Z \sim \zeta} \left[ \nabla \psi_0^{*c} \left( g_{\theta_0}(Z) \right) \right]$$

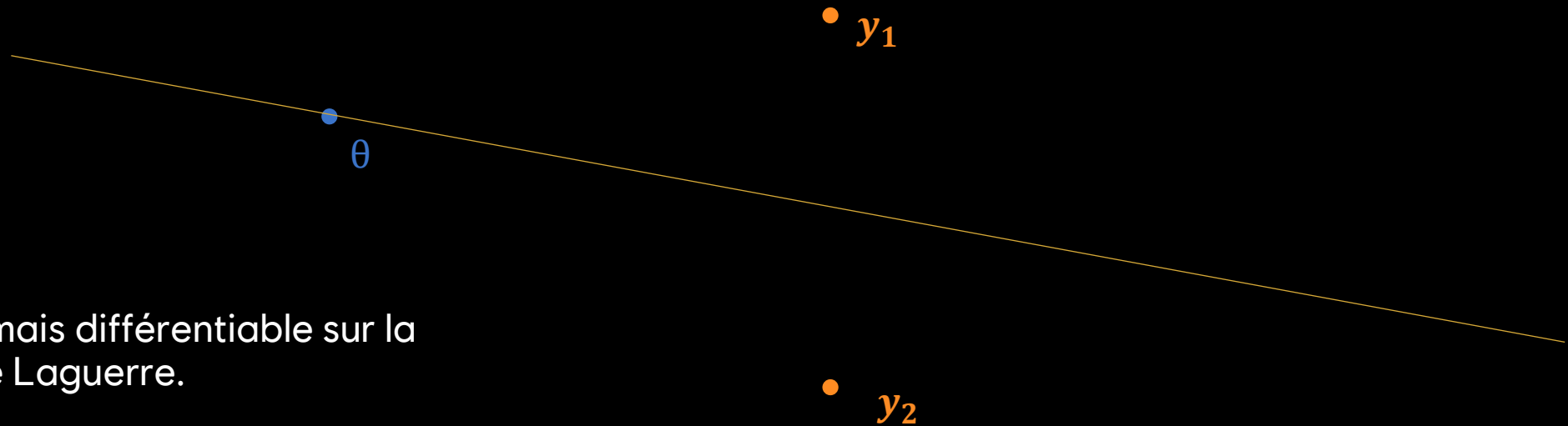
Dès que les deux termes sont définis

**Problème** : « dès que les **deux termes** sont définis »

Il existe des cas où cela n'est vrai pour aucun couple  $(\psi_0^*, \theta_0)$  !

$$\mu_\theta = \delta_\theta$$

$$v = \frac{1}{2} \delta_{y_1} + \frac{1}{2} \delta_{y_2}$$



La c-transform n'est jamais différentiable sur la frontière des cellules de Laguerre.

Or,  $\theta_0$  se retrouve toujours sur la frontière des cellules de Laguerre de  $\psi_0^*$  !



Solutions étudiées et proposées dans notre papier

*On the Gradient Formula for learning Generative Models with Regularized Optimal Transport Costs* (under review)  
[hal.archives-ouvertes.fr/hal-03740368/](https://hal.archives-ouvertes.fr/hal-03740368/)

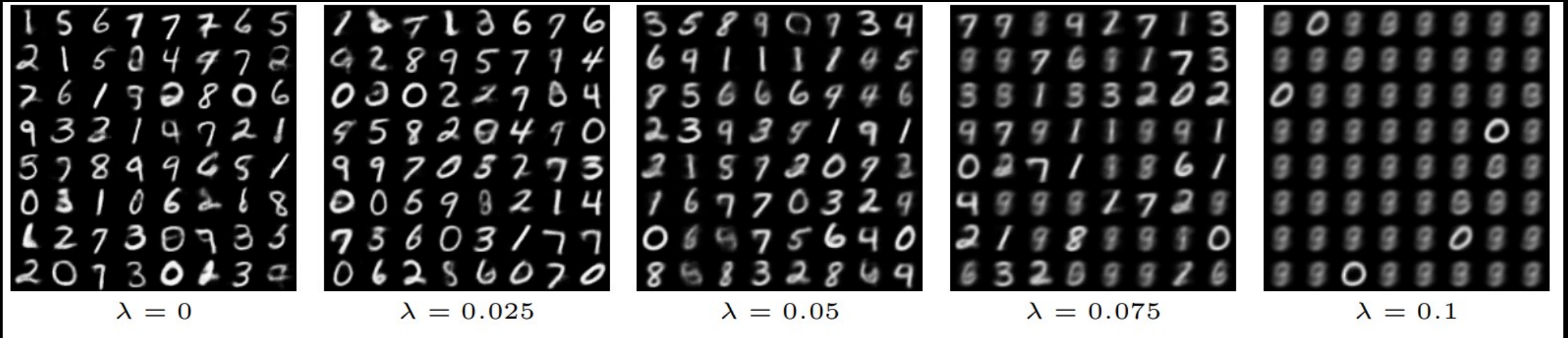
- Conditions d'existence et formulation du gradient dans le **cadre semi-discret** (Théorèmes 3 et 4)
- Conditions d'existence et formulation du gradient dans le cadre du **transport régularisé entropique** (Théorème 5)
- Conditions d'existence et formulation du gradient pour la **divergence de Sinkhorn** (Théorème 6)



Dans toutes ces configurations on peut **apprendre un modèle génératif** en effectuant une **descente de gradient stochastique** sur le coût de transport !



Exemple : modèle génératif appris la base de donnée MNIST avec du transport semi-discret avec et sans régularisation entropique

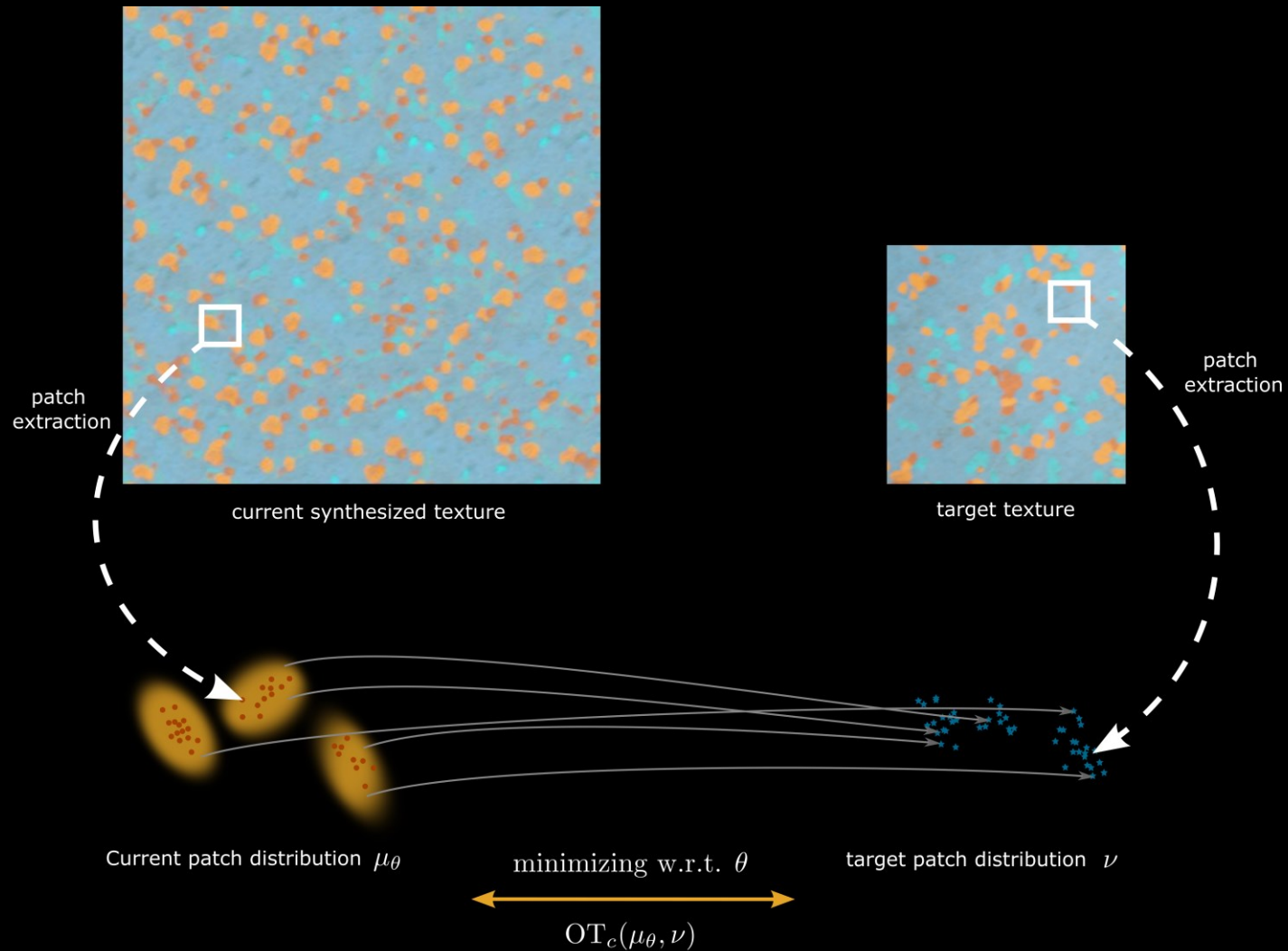


- ✓ Plus besoin d'avoir un discriminateur défini par un réseau de neurones
- ✓ Minimise vraiment le coût de transport
- ❖ Si la base de données est grande la variable duale est de grande dimension (temps de calcul)
- ❖ Le calcul approché du transport optimal à chaque itération limite la méthode

# Une application pertinente : single-image texture synthesis

*Wasserstein generative models for patch-based texture synthesis, SSVM 2021*

*A generative model for texture synthesis based on optimal transport between feature distributions, JMIV 2022*



# Quelques résultats



**Example**

**patch NN**

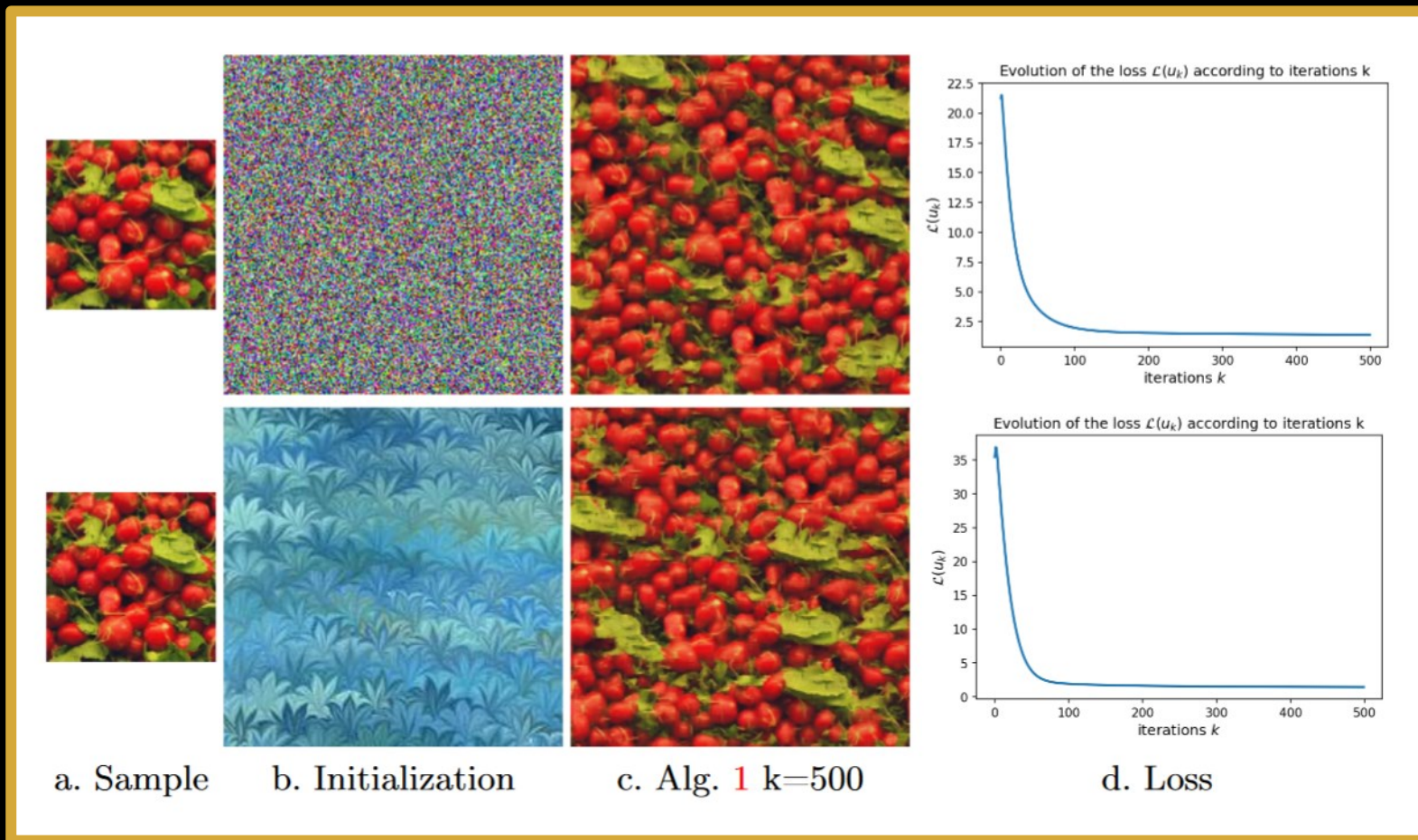
**Gatys'**

**Ours**





# Convergence et robustesse à l'initialisation





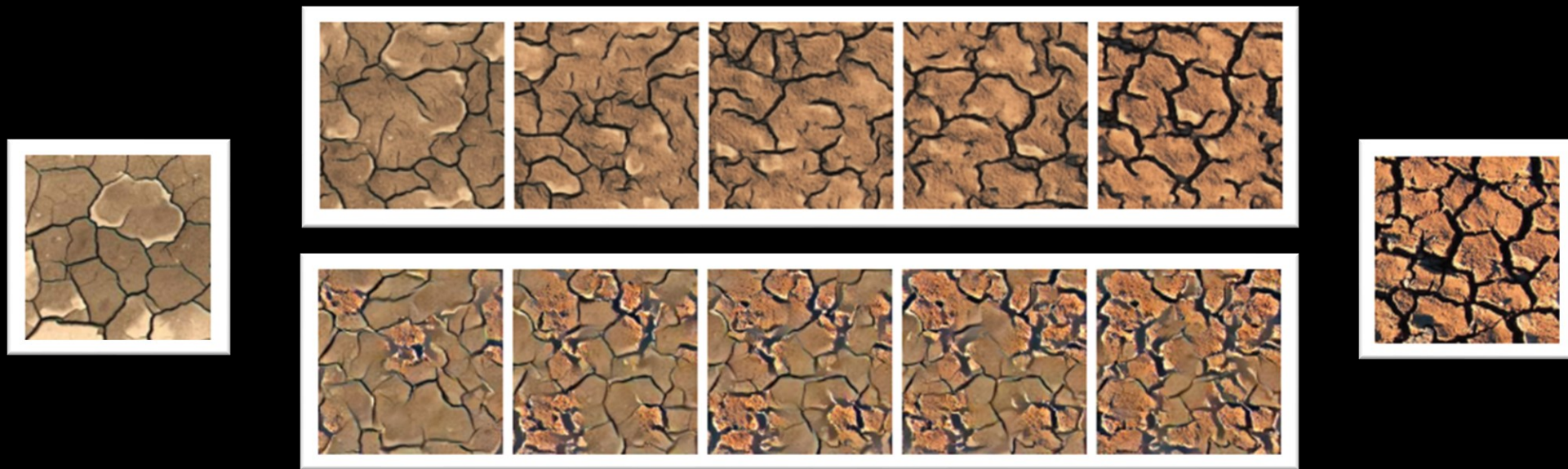
# Inpainting de textures





# Interpolation de textures

**Our — Optimal transport barycenter between patch distributions**

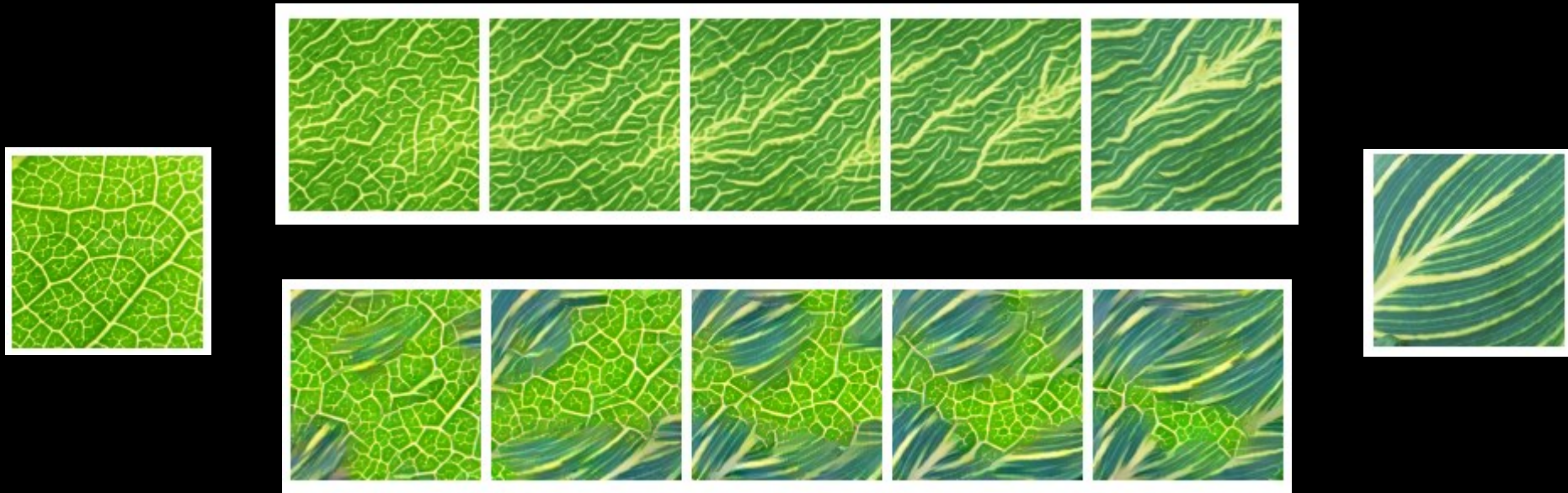


**Gatys — Gram loss interpolation between VGG features**



# Interpolation de textures

**Our — Optimal transport barycenter between patch distributions**



**Gatys — Gram loss interpolation between VGG features**



# Merci !

Papiers en ligne disponibles sur HAL et ArXiv  
Codes sur github [github.com/ahoudard/](https://github.com/ahoudard/)

Page perso [houdard.wp.imt.fr/](http://houdard.wp.imt.fr/)  
Twitter [@AntoineHou](https://twitter.com/AntoineHou)

