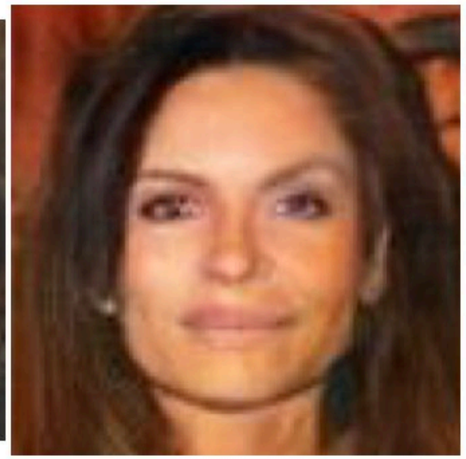
Apprentissage Automatique I
MATH60629

Modèles génératifs modernes
— Semaine #11

Évolution rapide des capacités à générer

2014

2015



2016

2017

2018

P(image)

https://twitter.com/goodfellow_ian/status/1084973596236144640/photo/1

Génération conditionnelle

texte

Une photo d'un raton laveur portant un casque d'astronaute qui regarde à travers une fenêtre la nuit

$P(\text{image} \mid \text{texte})$

image

https://imagen.research.google/

Pourquoi générer des images?

- Pendant un moment, c’était plutôt question de la curiosité scientifique
- Maintenant, on imagine que l’on pourrait utiliser ces images à plusieurs endroits où les images sont utiles (visualisations, jeux vidéos, présentations, publicité, etc…)
- Utilisateur dans la boucle

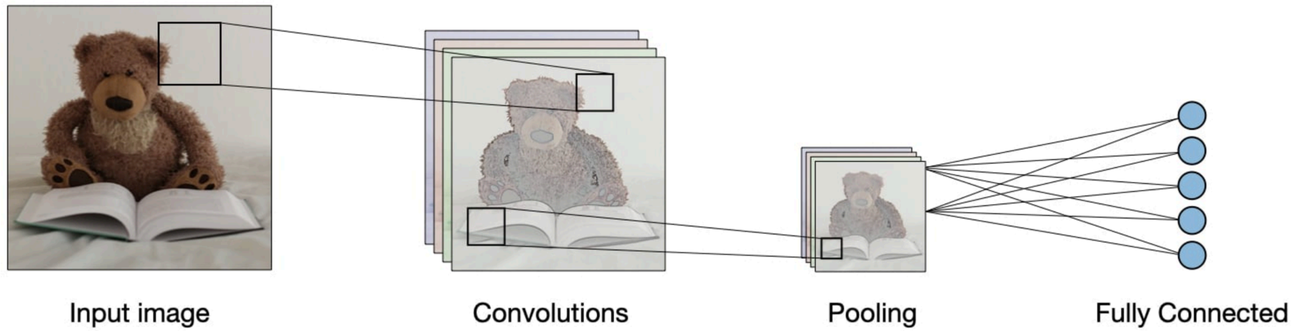
Plan de la séance*

- “Prédire” une image
- Modèle génératifs
  - Images
    - Auto-encodeurs variationnels, GANs
  - Images conditionnelles au texte
    - Dall-E (2), Imagen

* Les diapos viennent en majorité de David Berger

Réseaux à Convolution (CNN) — Rappel

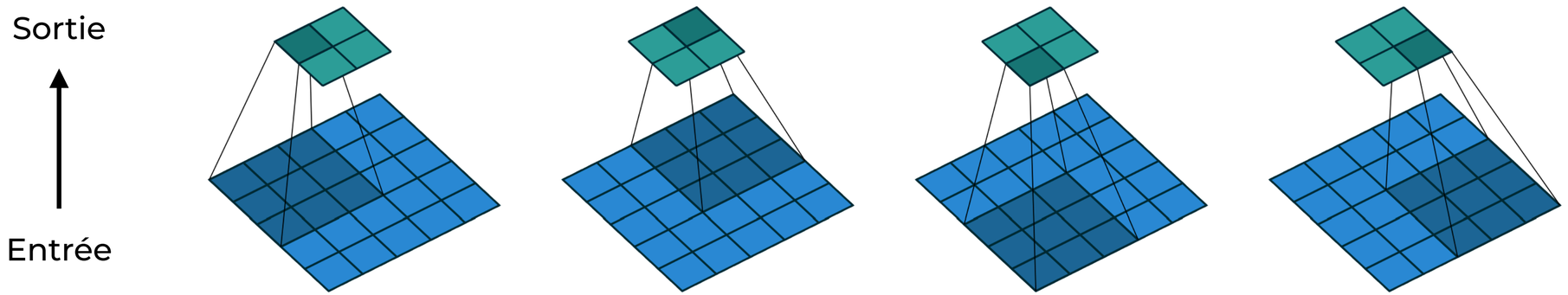
- Série de couches: convolutions + pooling
  - Chaque couche diminue la dimensionalité de la représentation



 https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

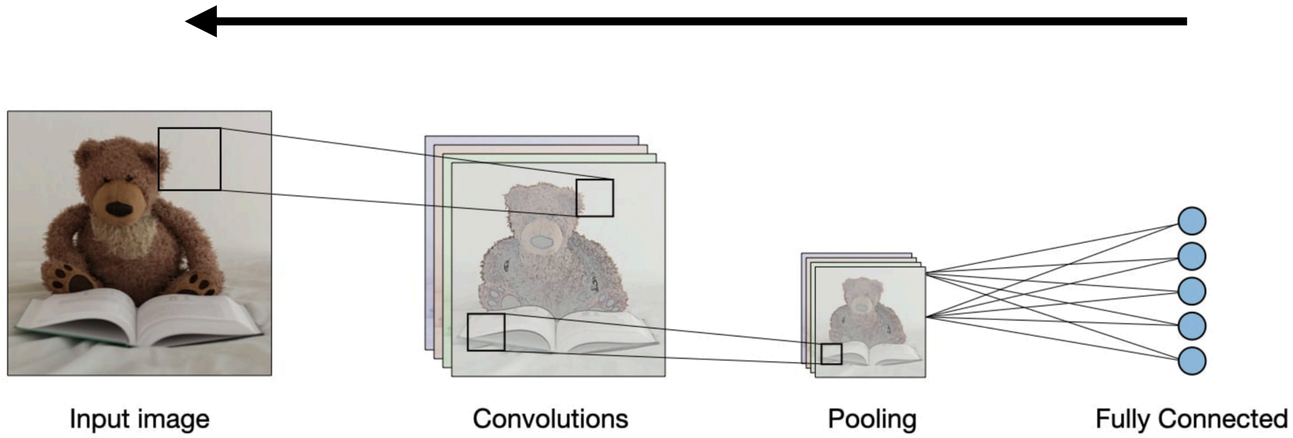
Convolution

- **Entrée: 5 x 5**
- **Filtre: 3 x 3. Pas 2.**
- **Sortie: 2 x 2**



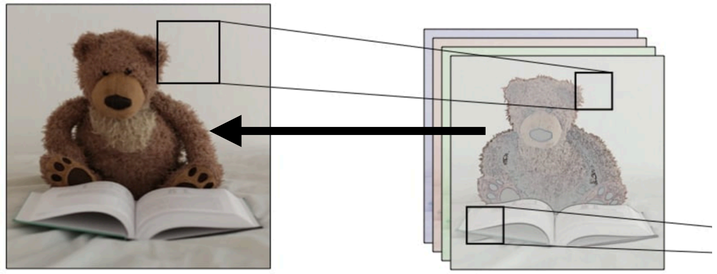
“Inverser” un réseau à convolution

- **Chaque couche augmente la dimensionnalité de la représentation**



- **(Note: c’est d’ailleurs l’opération effectuée par la rétropropagation)**

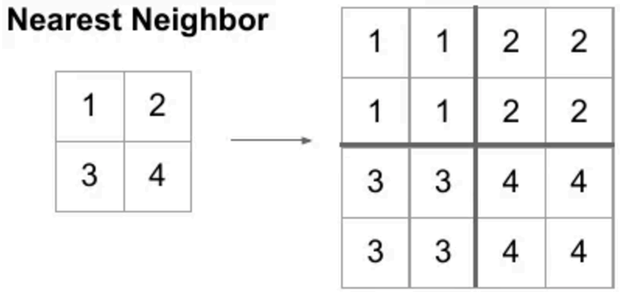
https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks



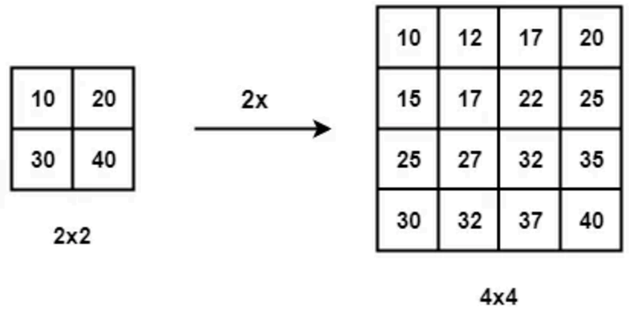
Input image Convolutions

Interpolation

1. Répétition



2. Interpolation (bi-)linéaire

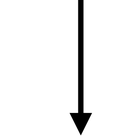


- Pas de paramètres à apprendre

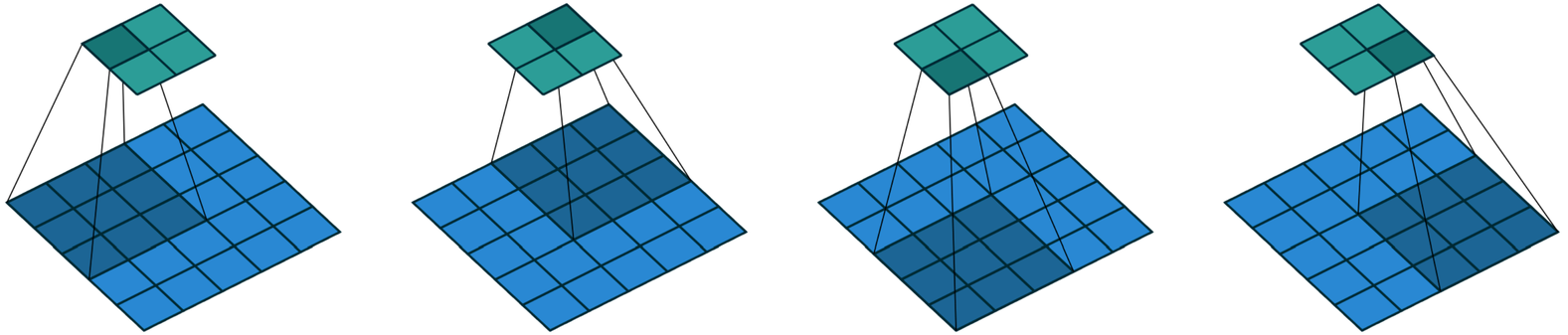
Convolution transposée

- Utile pour générer des images ou même augmenter la résolution d'une image (comme dans les films!)
- Entrée: 2 x 2
- Filtre: 3 x 3. Pas 1.
- Sortie: 3 x 3

Entrée

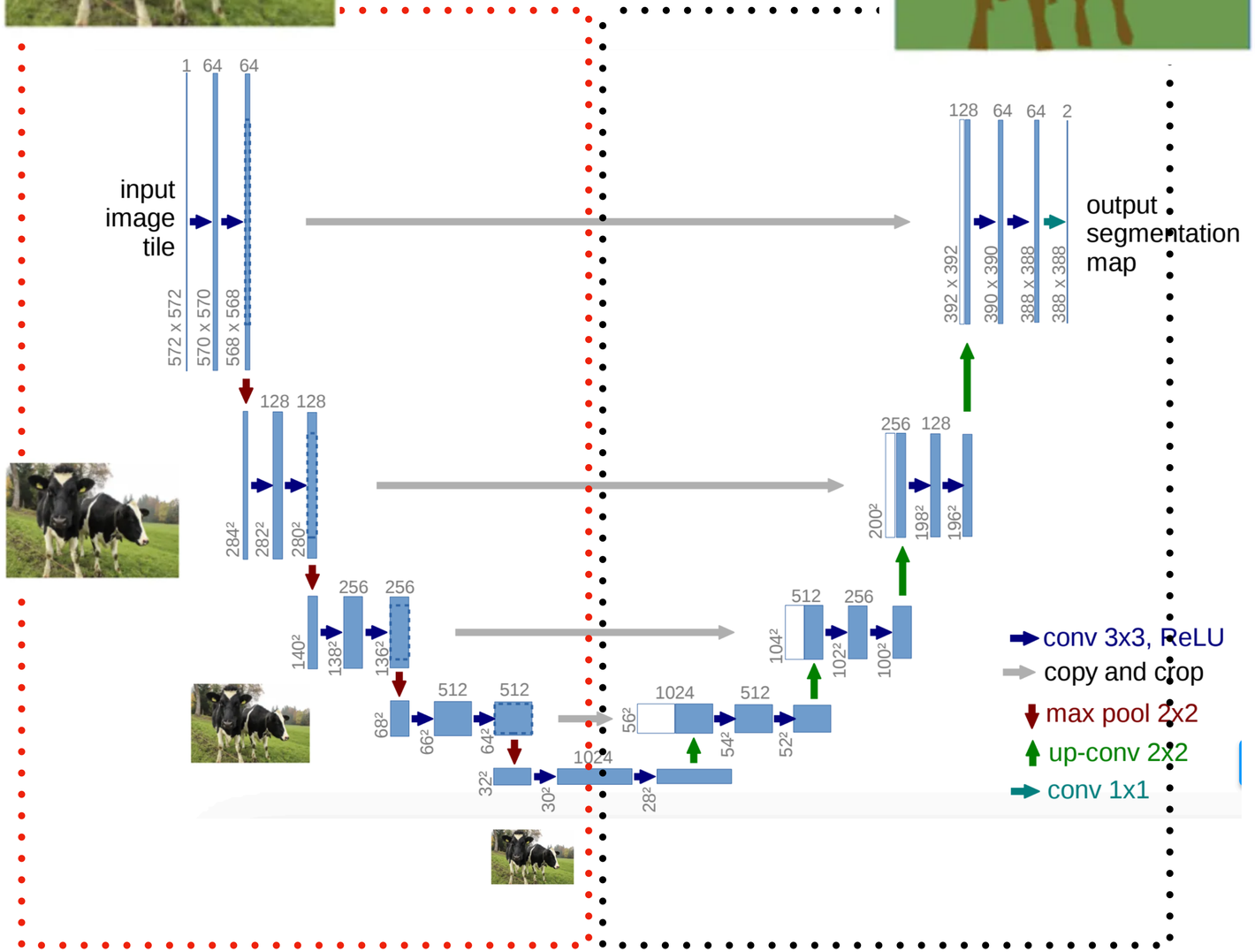


Sortie



- Si l'on écrit une convolution comme une multiplication de matrice. L'opération "dans l'autre sens" devient une multiplication par sa transposée.

U-Nets



- Architecture Encodeur-Décodeur
  - Encodeur: Obtient une représentation de l'image (CNN classique)
  - Décodeur: À partir, de la représentation obtient une image
  - Connexions entre les deux permettent de conserver une certaine information sémantique
- up-conv -> convolution transposée (ou autre)
- Proposé pour la segmentation
- Devenu l'architecture de choix pour la génération (conditionnelle) d'images

U-Nets: https://arxiv.org/pdf/1505.04597

https://wiki.cloudfactory.com/docs/mp-wiki/key-principles-of-computer-vision/upsampling-and-downsampling-techniques-in-machine-learning

Modèles générateurs

- Un U-Net permet d’obtenir une image à partir d’une représentation, mais n’a pas d’interprétation probabiliste

- Un modèle générateur est une façon de paramétrer $P(x)$ — non supervisé

Pourquoi les modèles générateurs sont-ils (souvent) probabilistes?

- Permet l'évaluation
  - P. ex., la probabilité d'une image selon le modèle: $P_\theta(x_{\textbf{new}})$
- Permet l'échantillonnage $x \sim P_\theta(x)$
- Quantifie l'incertitude
- À l'occasion on donne aussi une interprétation probabiliste à un modèle. Ça ne fait automatiquement un modèle générateur.

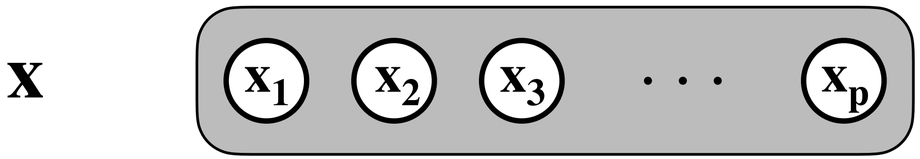
Auto-Encodeur

Auto-encodeur - Formalisme

Pour un AE avec une seule couche cachée:

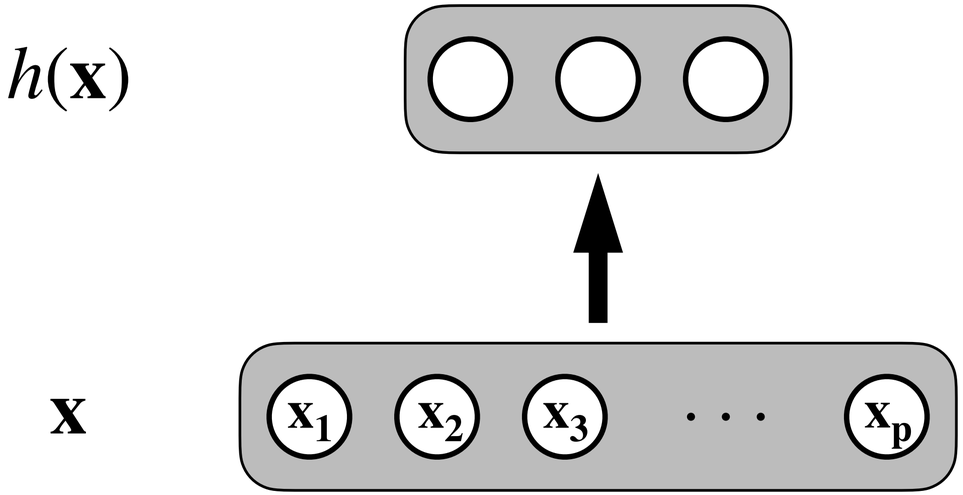
Auto-encodeur - Formalisme

Pour un AE avec une seule couche cachée:



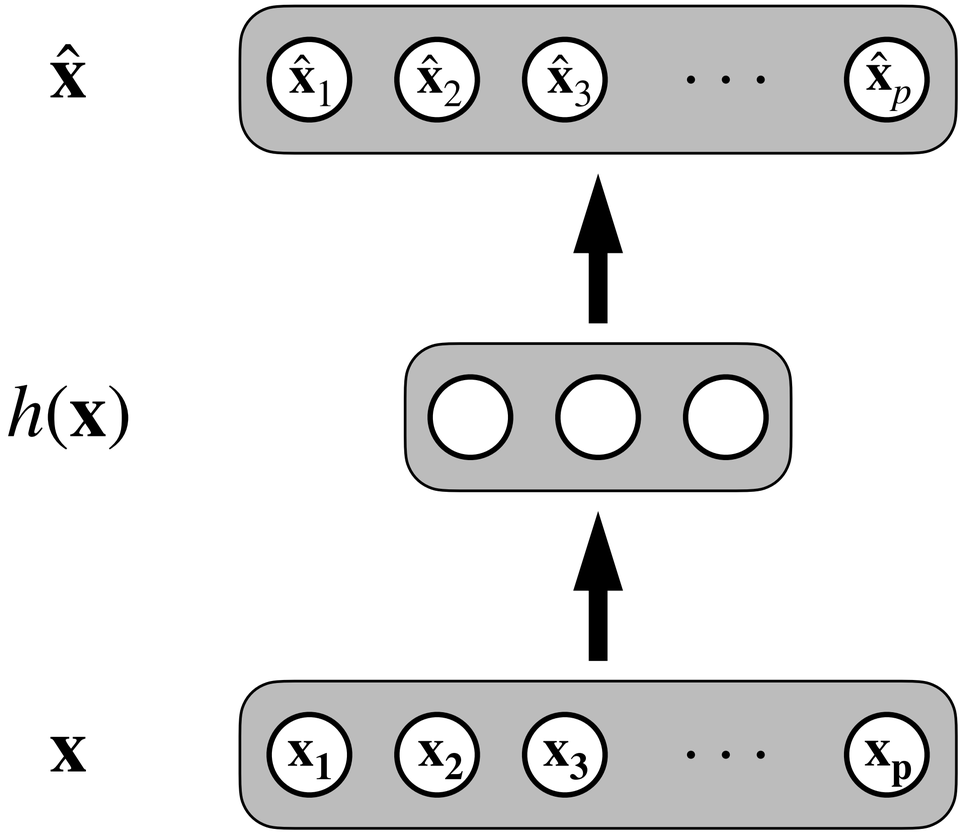
Auto-encodeur - Formalisme

Pour un AE avec une seule couche cachée:



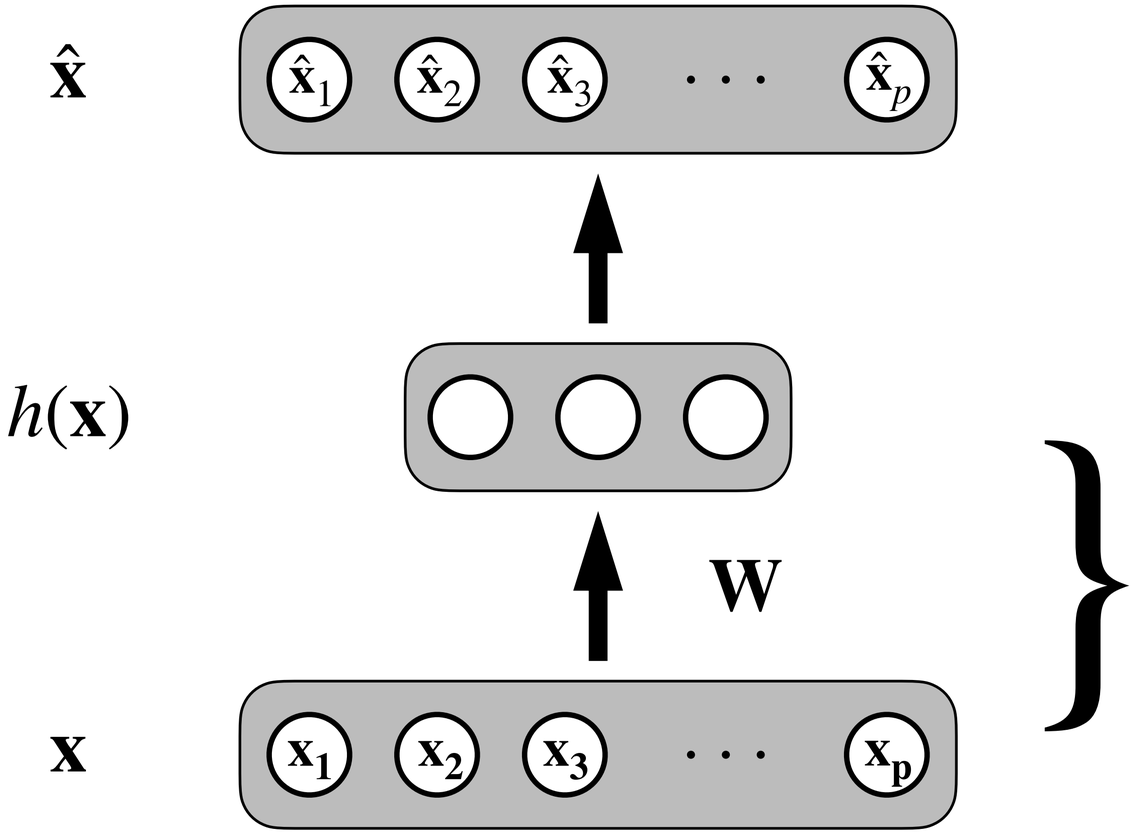
Auto-encodeur - Formalisme

Pour un AE avec une seule couche cachée:



Auto-encodeur - Formalisme

Pour un AE avec une seule couche cachée:

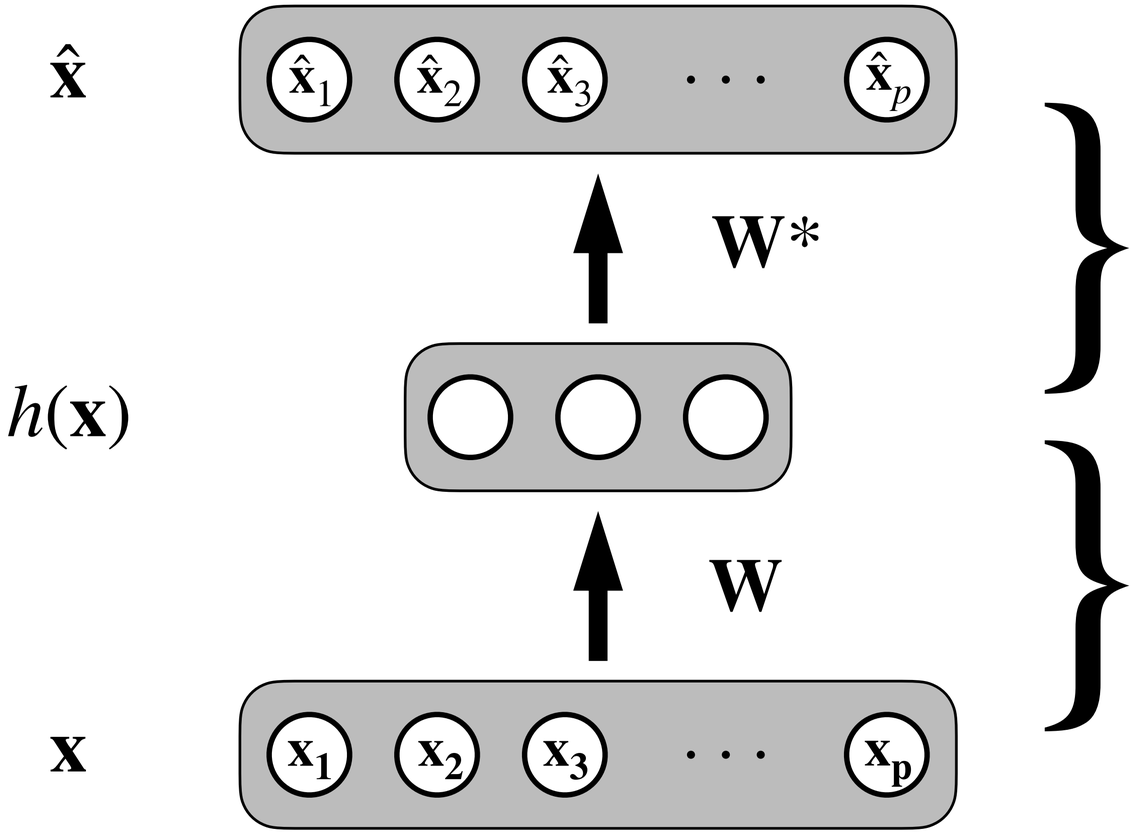


Encodeur

$$h(\mathbf{x}) = g(\mathbf{a}(\mathbf{x}))$$
$$= g(\mathbf{W}\mathbf{x} + \mathbf{b})$$

Auto-encodeur - Formalisme

Pour un AE avec une seule couche cachée:



Décodeur

$$\hat{\mathbf{x}} = o(\hat{\mathbf{a}}(\mathbf{x})) = o(\mathbf{W}^* h(\mathbf{x}) + \mathbf{b}^*)$$

Encodeur

$$h(\mathbf{x}) = g(\mathbf{a}(\mathbf{x})) = g(\mathbf{W}\mathbf{x} + \mathbf{b})$$

Auto-encodeur variationnel

*La compréhension exacte de ces modèles requièrent des notions qui sont plus avancées que celles vues en cours.
Ici, on tente de vous donner une intuition.

Auto-encodeur variationnels - Motivations

Idée:

- Les données observées sont générées à partir d'une variable latente:

$$\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z})$$

Z est une variable aléatoire (random variable), Vous pouvez y pensez comme étant un « embedding », une représentation que nous allons apprendre pour générer de bons x.

Auto-encodeur variationnels - Motivations

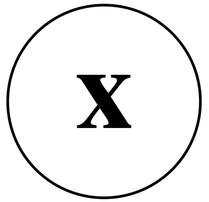
Idée:

- Les données observées sont générées à partir d'une variable latente:

$$\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z})$$

Z est une variable aléatoire (random variable), Vous pouvez y pensez comme étant un « embedding », une représentation que nous allons apprendre pour générer de bons x.

Graphiquement



Auto-encodeur variationnels - Motivations

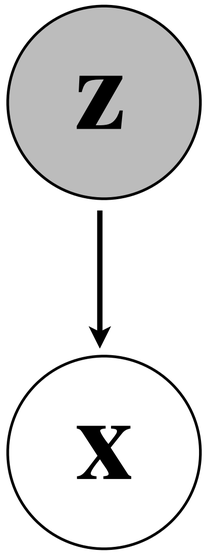
Idée:

- Les données observées sont générées à partir d'une variable latente:

$$\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z})$$

Z est une variable aléatoire (random variable), Vous pouvez y pensez comme étant un « embedding », une représentation que nous allons apprendre pour générer de bons x.

Graphiquement



Auto-encodeur variationnels - Motivations

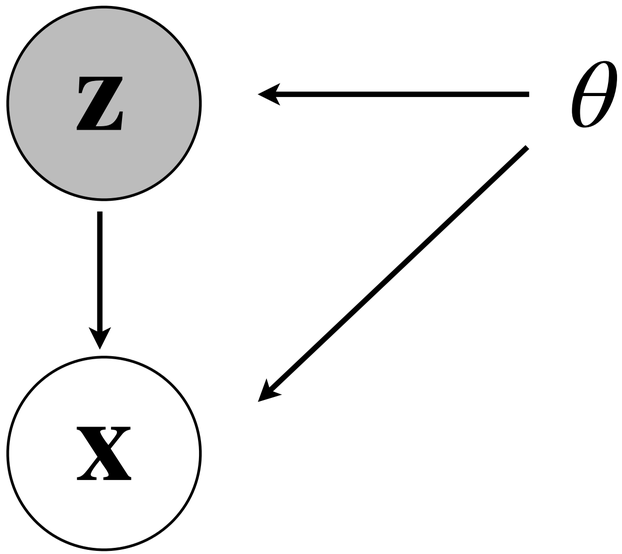
Idée:

- Les données observées sont générées à partir d'une variable latente:

$$\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z})$$

Z est une variable aléatoire (random variable), Vous pouvez y pensez comme étant un « embedding », une représentation que nous allons apprendre pour générer de bons x.

Graphiquement



Auto-encodeur variationnels - Motivations

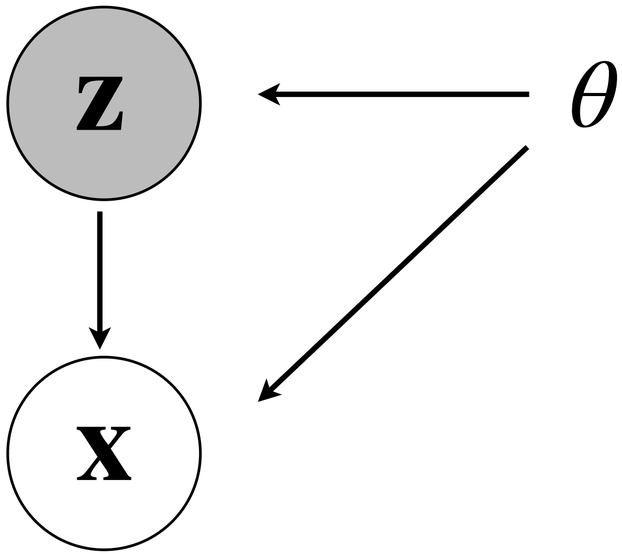
Idée:

- Les données observées sont générées à partir d'une variable latente:

$$\mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z})$$

Z est une variable aléatoire (random variable), Vous pouvez y pensez comme étant un « embedding », une représentation que nous allons apprendre pour générer de bons x.

Graphiquement



Question:

- Comment faire pour apprendre pareille distribution?

Auto-encodeur variationnels - Motivations

Auto-encodeur variationnels - Motivations

Réponse partielle:

- Un «bon» modèle latent devrait maximiser la vraisemblance des données observées.
- Par Bayes, deux options se présentent:

Auto-encodeur variationnels - Motivations

Réponse partielle:

- Un «bon» modèle latent devrait maximiser la vraisemblance des données observées.
- Par Bayes, deux options se présentent:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) \mathrm{d}\mathbf{z}$$

Auto-encodeur variationnels - Motivations

Réponse partielle:

- Un «bon» modèle latent devrait maximiser la vraisemblance des données observées.
- Par Bayes, deux options se présentent:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) \mathrm{d}\mathbf{z}$$

Problème:

Auto-encodeur variationnels - Motivations

Réponse partielle:

- Un «bon» modèle latent devrait maximiser la vraisemblance des données observées.
- Par Bayes, deux options se présentent:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) \mathrm{d}\mathbf{z}$$

Problème:

- Le posterior peut être intractable.

Auto-encodeur variationnels - Motivations

Réponse partielle:

- Un «bon» modèle latent devrait maximiser la vraisemblance des données observées.
- Par Bayes, deux options se présentent:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Problème:

- Le posterior peut être intractable.

Auto-encodeur variationnels - Motivations

Réponse partielle:

- Un «bon» modèle latent devrait maximiser la vraisemblance des données observées.
- Par Bayes, deux options se présentent:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z}) \mathbb{P}_\theta(\mathbf{z}) d\mathbf{z}$$

Problème:

- Le posterior peut être intractable.

Auto-encodeur variationnels - Motivations

Réponse partielle:

- Un «bon» modèle latent devrait maximiser la vraisemblance des données observées.
- Par Bayes, deux options se présentent:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z}) \mathbb{P}_\theta(\mathbf{z}) d\mathbf{z}$$

Problème:

- Le posterior peut être intractable.
- L'intégrale est intractable.

Auto-encodeur variationnels - Motivations

Réponse partielle:

- Un «bon» modèle latent devrait maximiser la vraisemblance des données observées.
- Par Bayes, deux options se présentent:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{x} \mid \mathbf{z}) \mathbb{P}_\theta(\mathbf{z}) d\mathbf{z}$$

Problème:

- Le posterior peut être intractable.
- L'intégrale est intractable.

Auto-encodeur variationnels - Motivations

Explorons le premier problème:

- Le posterior peut être intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Auto-encodeur variationnels - Motivations

Explorons le premier problème:

- Le posterior peut être intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Idée:

- Nous pourrions proposer une approximation du vrai posterior, de façon telle à ce que cette dernière soit estimable.

Auto-encodeur variationnels - Motivations

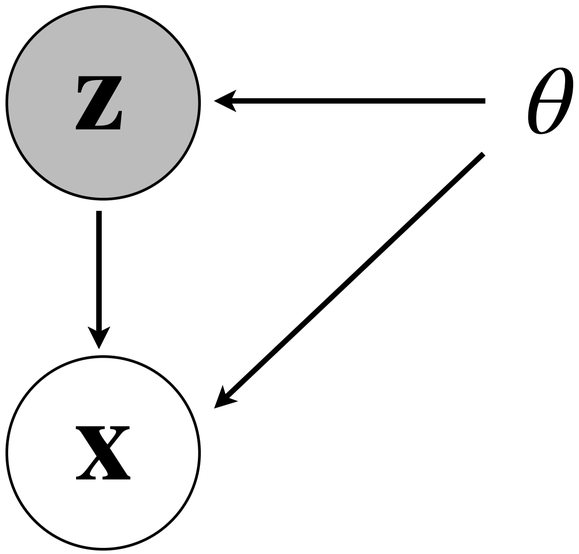
Explorons le premier problème:

- Le posterior peut être intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Idée:

- Nous pourrions proposer une approximation du vrai posterior, de façon telle à ce que cette dernière soit estimable.



Auto-encodeur variationnels - Motivations

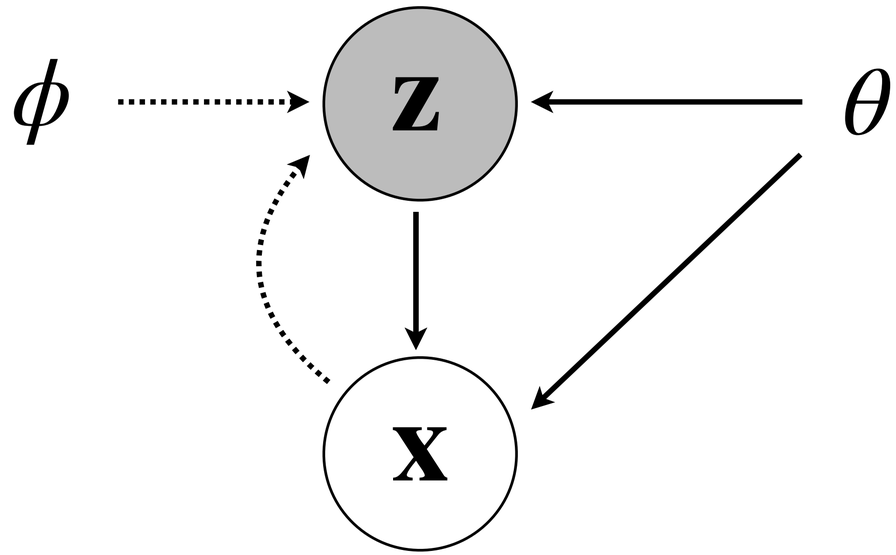
Explorons le premier problème:

- Le posterior peut être intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Idée:

- Nous pourrions proposer une approximation du vrai posterior, de façon telle à ce que cette dernière soit estimable.



Auto-encodeur variationnels - Motivations

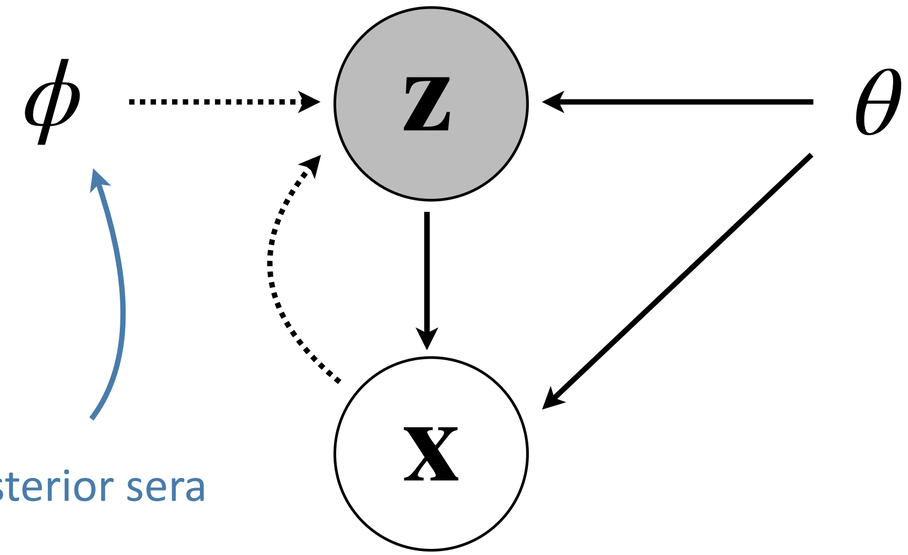
Explorons le premier problème:

- Le posterior peut être intractable:

$$\mathbb{P}_\theta(\mathbf{x}) = \int_{\mathbf{z}} \mathbb{P}_\theta(\mathbf{z} \mid \mathbf{x}) \mathbb{P}_\theta(\mathbf{x}) d\mathbf{z}$$

Idée:

- Nous pourrions proposer une approximation du vrai posterior, de façon telle à ce que cette dernière soit estimable.



L'approximation du posterior sera paramétrique par ϕ

Auto-encodeur variationnels - Formalisme

Nous avons obtenu une borne sur la log-vraisemblance de $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Auto-encodeur variationnels - Formalisme

Nous avons obtenu une borne sur la log-vraisemblance de $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

où:

Auto-encodeur variationnels - Formalisme

Nous avons obtenu une borne sur la log-vraisemblance de $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

où:

- nous chercherons à minimiser la distance entre le posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ et l'apriori sur $q_\phi(\mathbf{z})$.

Auto-encodeur variationnels - Formalisme

Nous avons obtenu une borne sur la log-vraisemblance de $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

où:

- nous chercherons à minimiser la distance entre le posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ et l'apriori sur $q_\phi(\mathbf{z})$.

Auto-encodeur variationnels - Formalisme

Nous avons obtenu une borne sur la log-vraisemblance de $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

où:

- nous chercherons à minimiser la distance entre le posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ et l'apriori sur $q_\phi(\mathbf{z})$.
- tout en maximisant la log-vraisemblance conditionnel de $\mathbf{x}$.

Auto-encodeur variationnels - Formalisme

Nous avons obtenu une borne sur la log-vraisemblance de $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

où:

- nous chercherons à minimiser la distance entre le posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ et l'apriori sur $q_\phi(\mathbf{z})$.
- tout en maximisant la log-vraisemblance conditionnel de $\mathbf{x}$.

Auto-encodeur variationnels - Formalisme

Nous avons obtenu une borne sur la log-vraisemblance de $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \mid\mid q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

où:

- nous chercherons à minimiser la distance entre le posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ et l'apriori sur $q_\phi(\mathbf{z})$.
- tout en maximisant la log-vraisemblance conditionnel de $\mathbf{x}$.

Question:

Auto-encodeur variationnels - Formalisme

Nous avons obtenu une borne sur la log-vraisemblance de $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

où:

- nous chercherons à minimiser la distance entre le posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ et l'apriori sur $q_\phi(\mathbf{z})$.
- tout en maximisant la log-vraisemblance conditionnel de $\mathbf{x}$.

Question:

- Comment procède-t-on en pratique?

Auto-encodeur variationnels - Formalisme

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l’inégalité ci-bas:

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \;\geq -\,\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \;||\; q_\phi(\mathbf{z})\} \;+\; \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

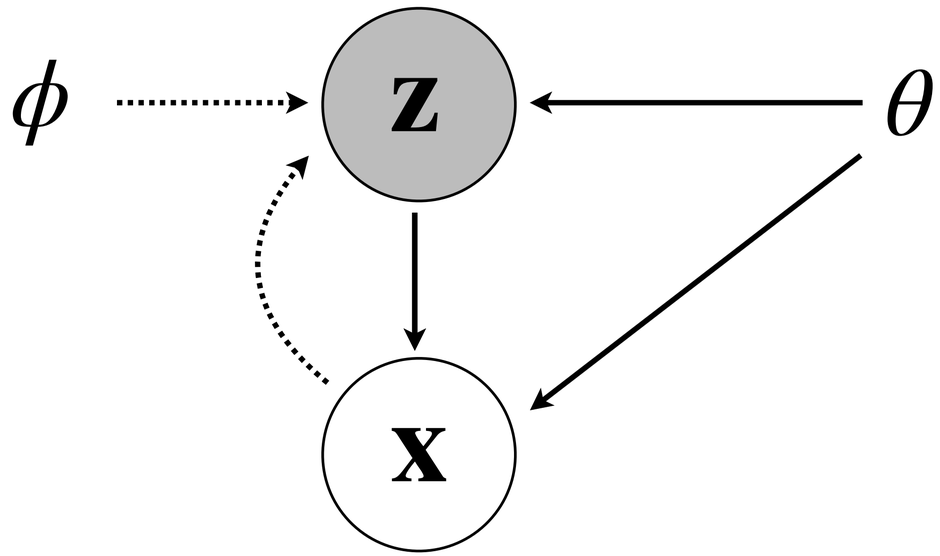
et en se rappelant notre petit graphe introduit précédemment:

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

et en se rappelant notre petit graphe introduit précédemment:

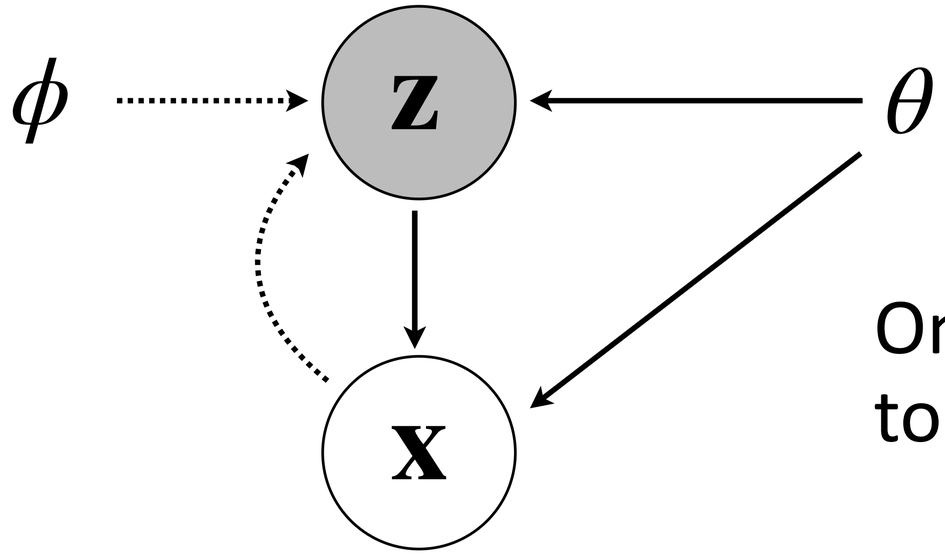


Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \;\geq -\,\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \;||\; q_\phi(\mathbf{z})\} \;+\; \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

et en se rappelant notre petit graphe introduit précédemment:



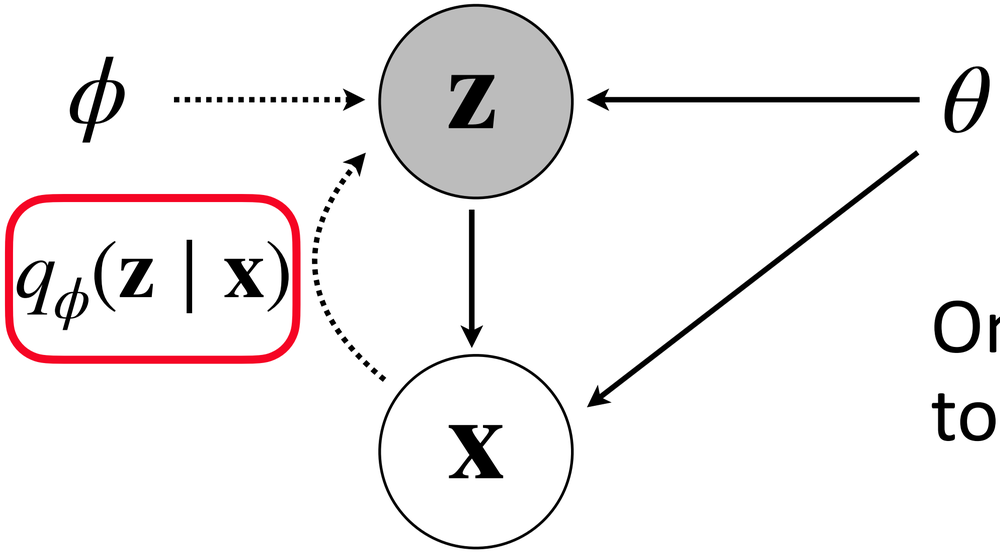
On se rend bien compte que nous avons tout ce qu'il faut pour faire de l'inférence!

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

et en se rappelant notre petit graphe introduit précédemment:



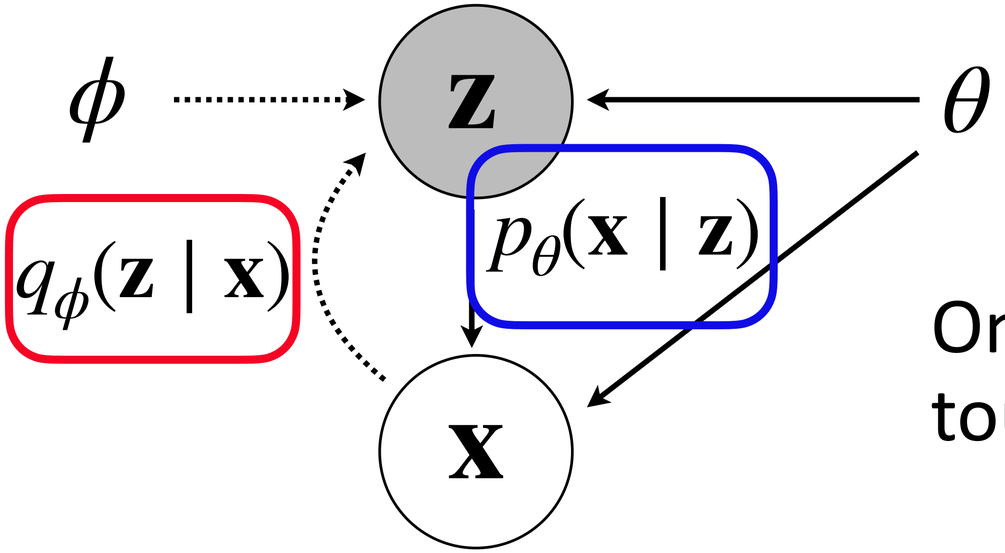
On se rend bien compte que nous avons tout ce qu'il faut pour faire de l'inférence!

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

et en se rappelant notre petit graphe introduit précédemment:



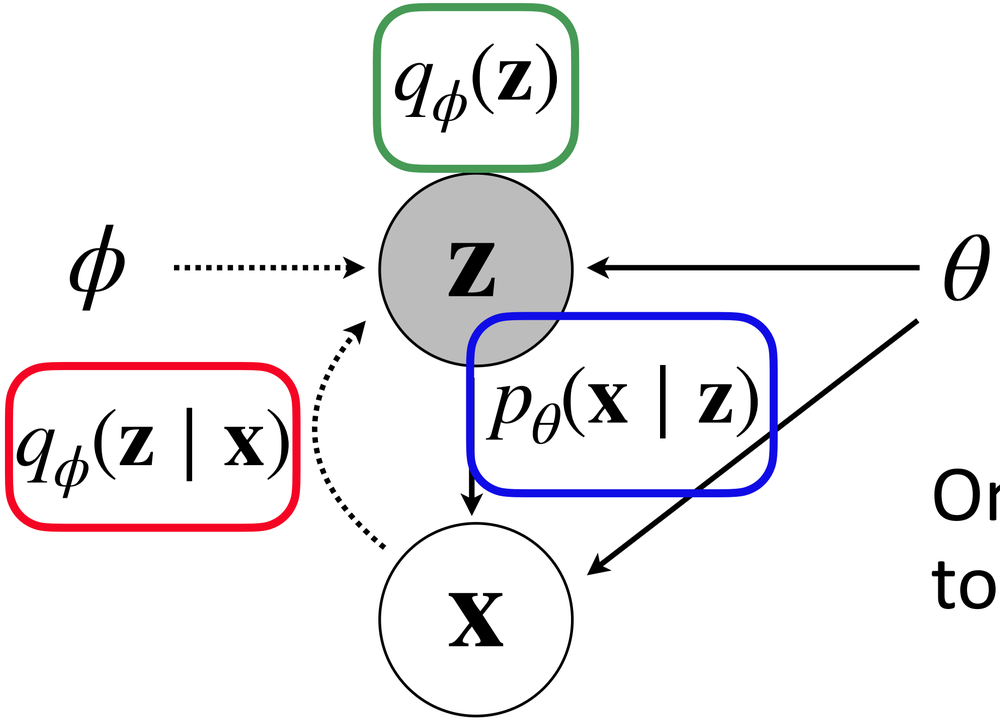
On se rend bien compte que nous avons tout ce qu'il faut pour faire de l'inférence!

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

et en se rappelant notre petit graphe introduit précédemment:



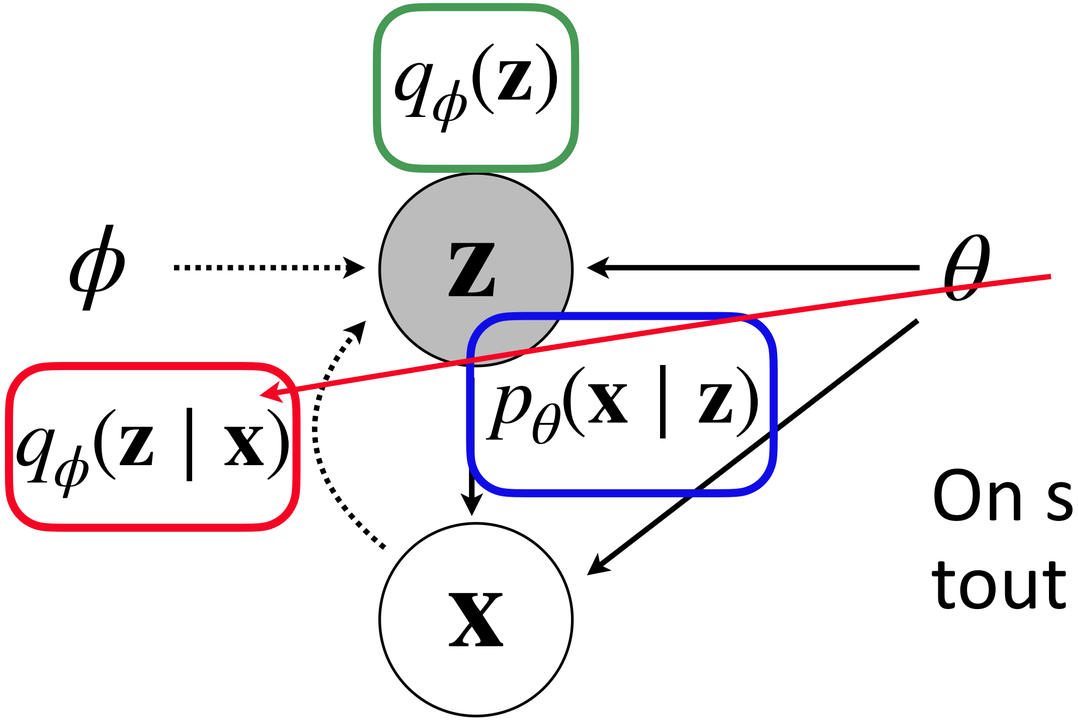
On se rend bien compte que nous avons tout ce qu'il faut pour faire de l'inférence!

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

et en se rappelant notre petit graphe introduit précédemment:



Où l'on suppose que le posterior suit une distribution dont les paramètres sont inconnus

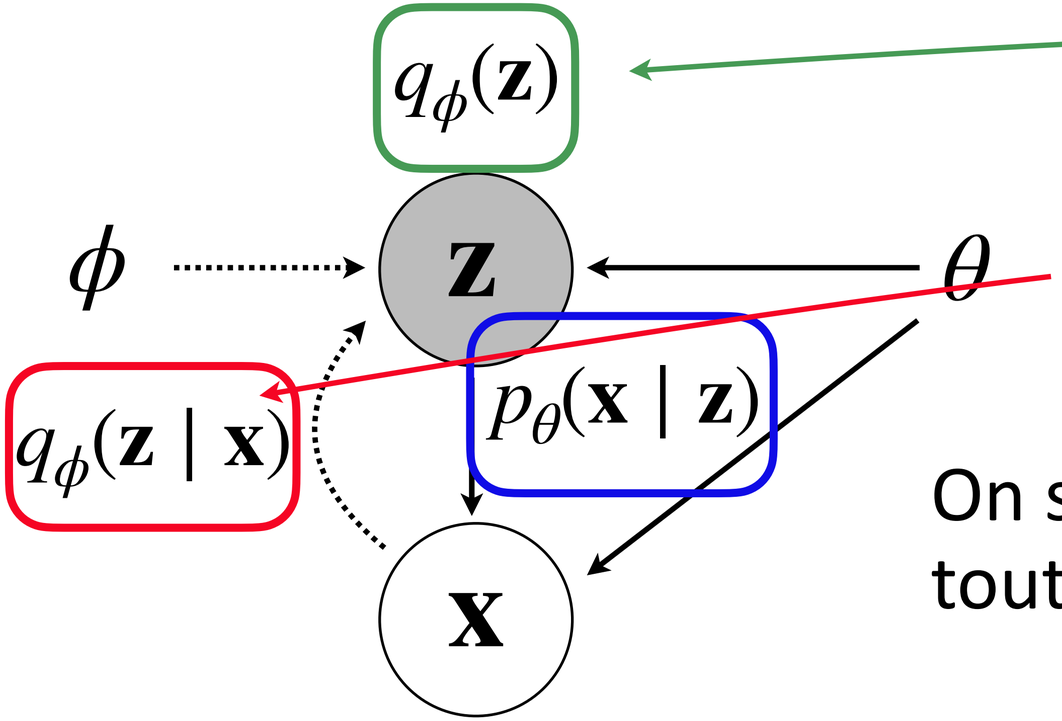
On se rend bien compte que nous avons tout ce qu'il faut pour faire de l'inférence!

Auto-encodeur variationnels - Formalisme

Faut juste pas paniquer! En considérant l'inégalité ci-bas:

$$\log p_\theta(\mathbf{x}) \geq -\mathrm{KL}\{q_\phi(\mathbf{z} \mid \mathbf{x}) \,||\, q_\phi(\mathbf{z})\} + \log p_\theta(\mathbf{x} \mid \mathbf{z}),$$

et en se rappelant notre petit graphe introduit précédemment:



Où l'on suppose que la marginale suit une distribution choisie (par exemple une normale centrée réduite)

Où l'on suppose que le posterior suit une distribution dont les paramètres sont inconnus

On se rend bien compte que nous avons tout ce qu'il faut pour faire de l'inférence!

Auto-encodeur variationnels

Auto-encodeur variationnels

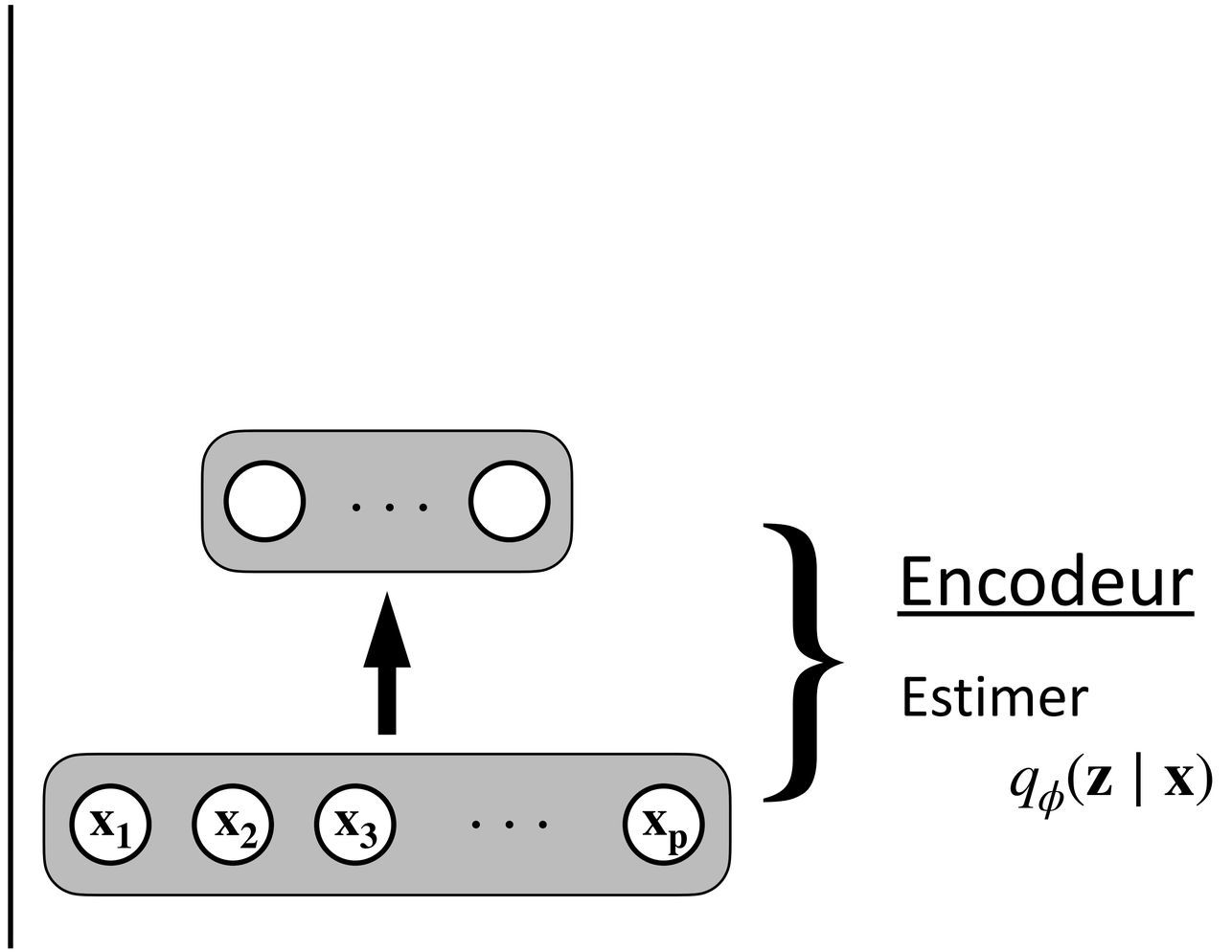
Les auto-encodeurs présentent l’architecture parfaite pour estimer pareil modèle:

Auto-encodeur variationnels

Les auto-encodeurs présentent l’architecture parfaite pour estimer pareil modèle:

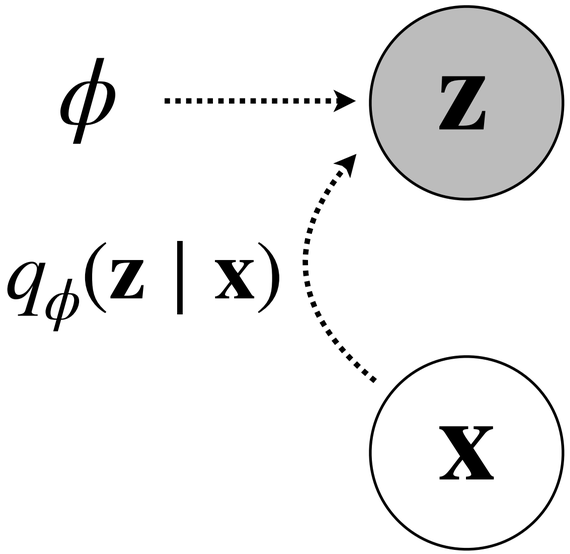
Auto-encodeur variationnels

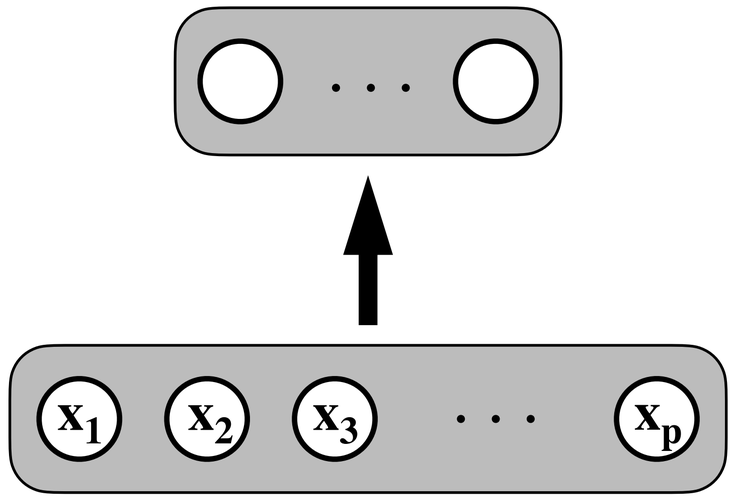
Les auto-encodeurs présentent l'architecture parfaite pour estimer pareil modèle:



Auto-encodeur variationnels

Les auto-encodeurs présentent l'architecture parfaite pour estimer pareil modèle:



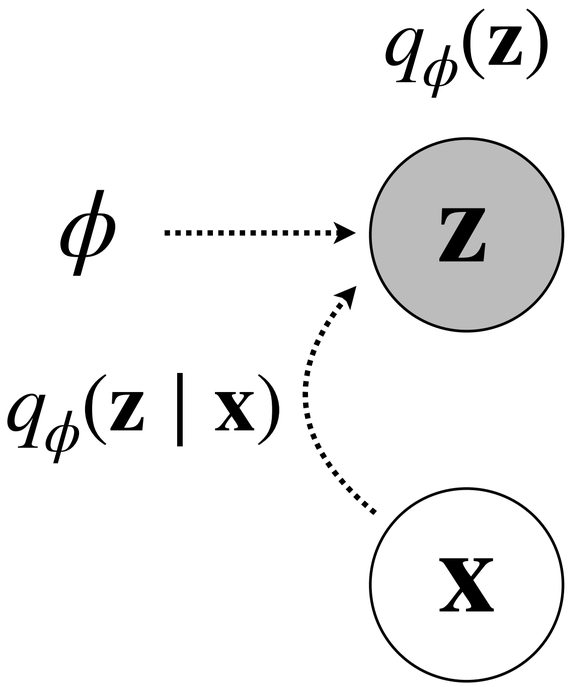


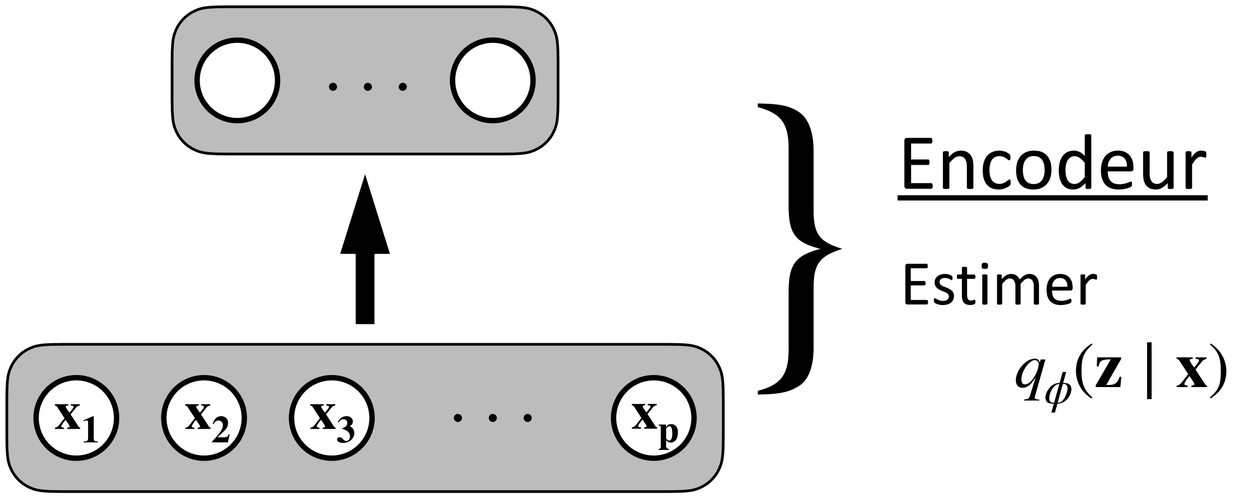
Encodeur

Estimer $q_\phi(\mathbf{z} \mid \mathbf{x})$

Auto-encodeur variationnels

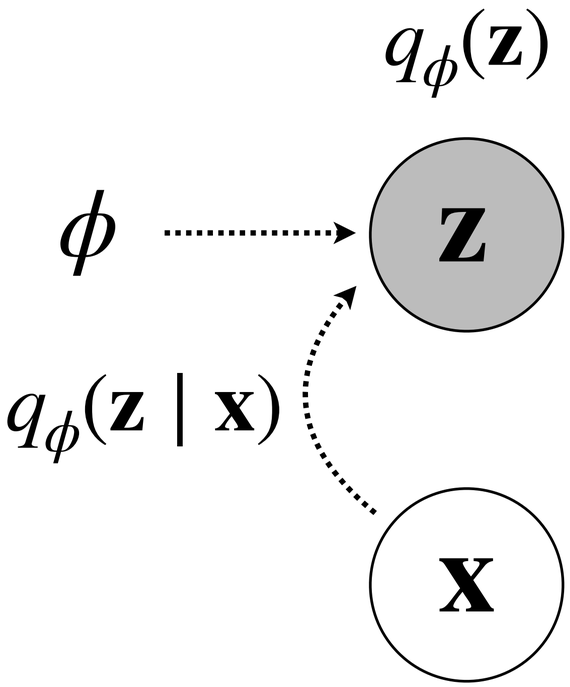
Les auto-encodeurs présentent l'architecture parfaite pour estimer pareil modèle:

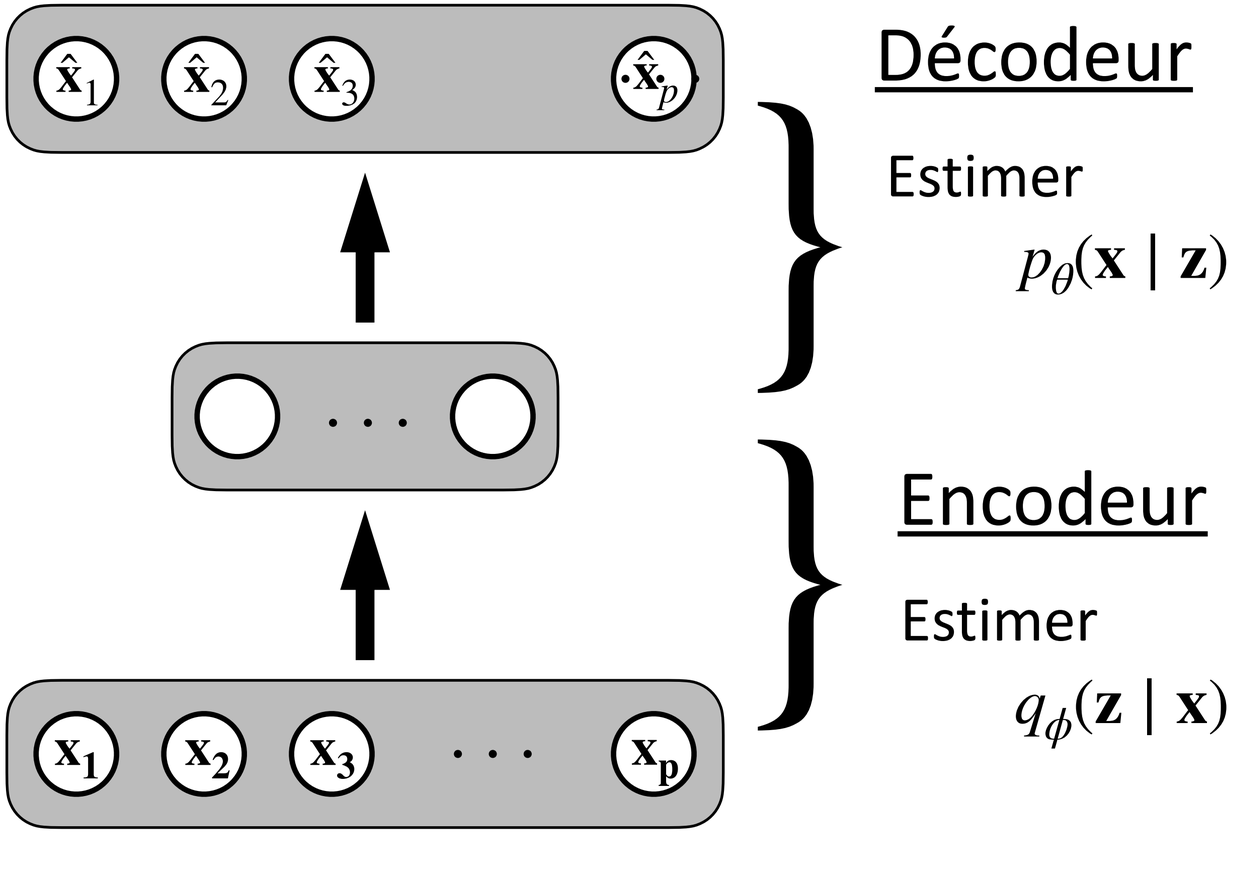




Auto-encodeur variationnels

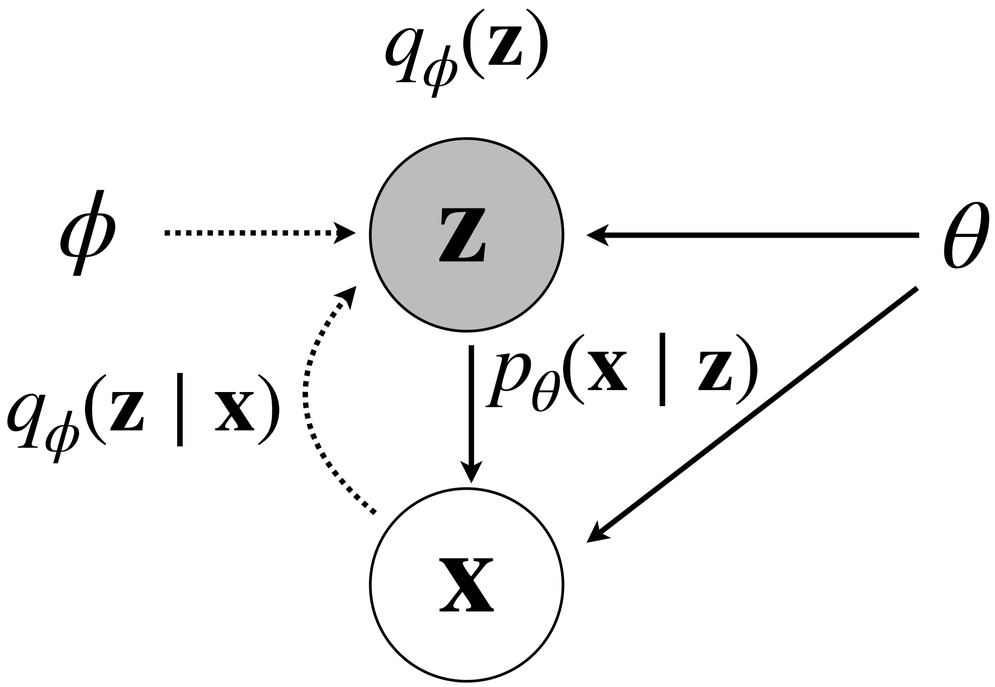
Les auto-encodeurs présentent l'architecture parfaite pour estimer pareil modèle:

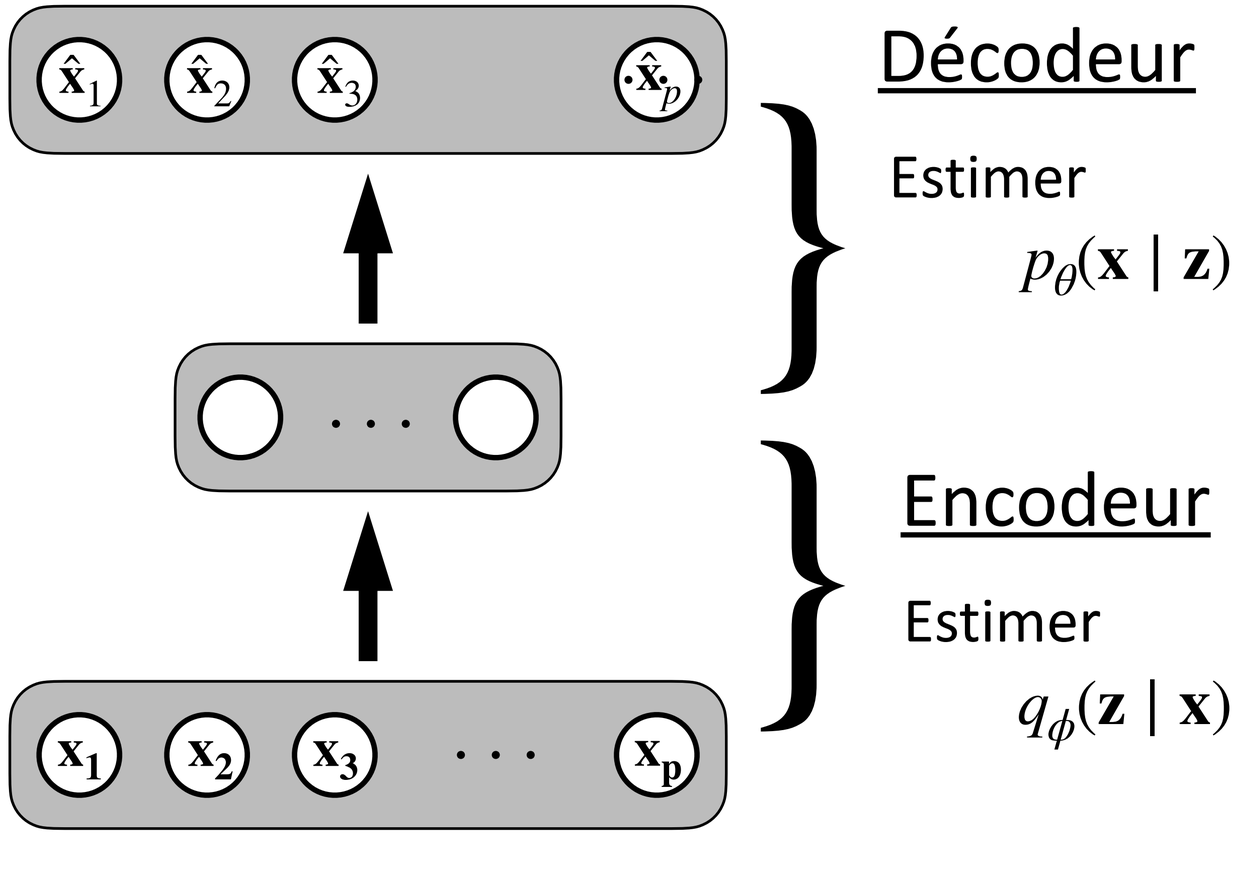




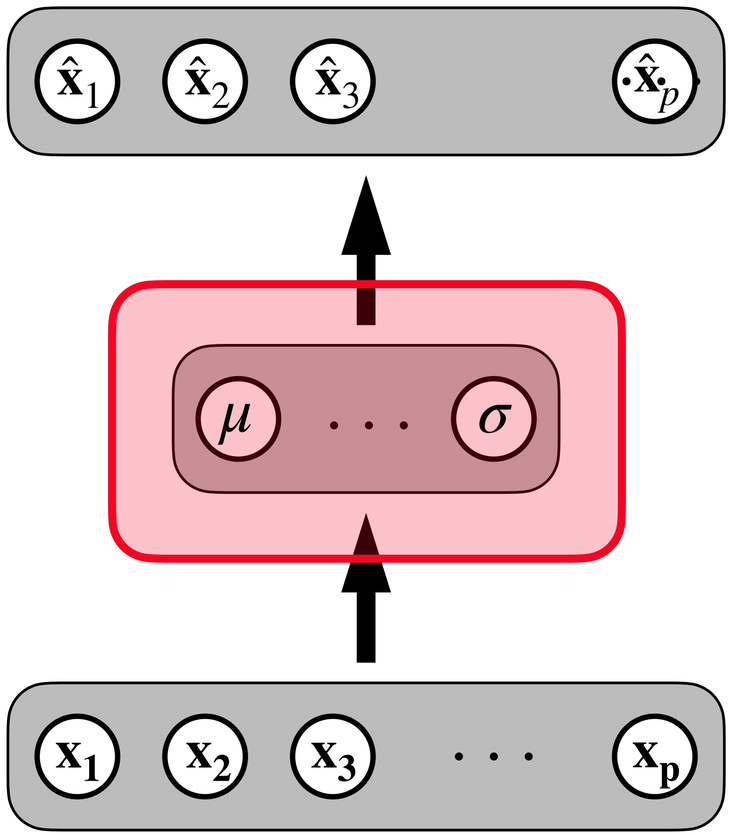
Auto-encodeur variationnels

Les auto-encodeurs présentent l'architecture parfaite pour estimer pareil modèle:





Auto-encodeur variationnels



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

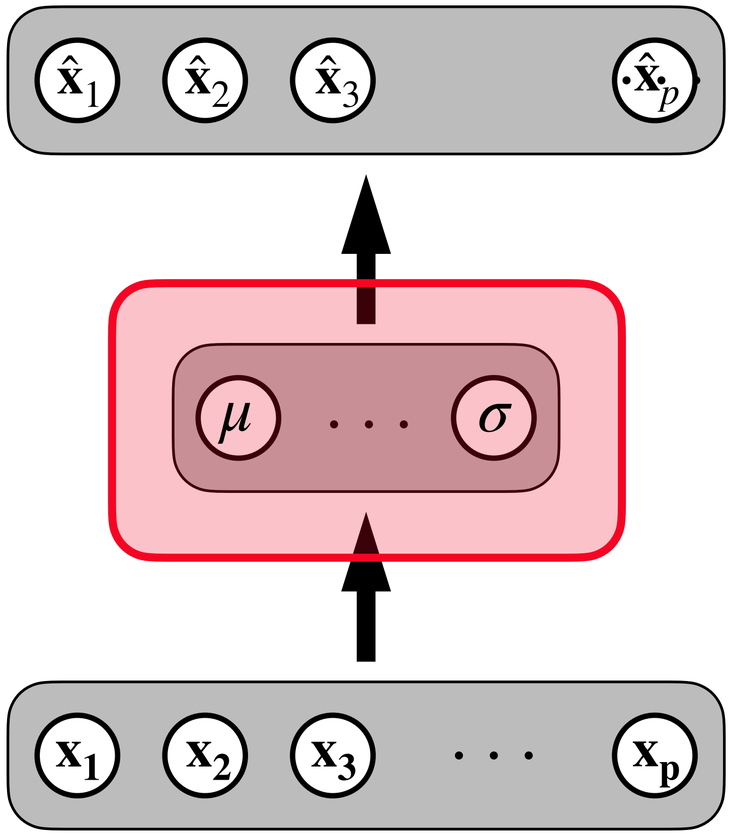
Encodeur

Estimer

$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Auto-encodeur variationnels

Reparamétrisation de la couche cachée:



<u>Décodeur</u>

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

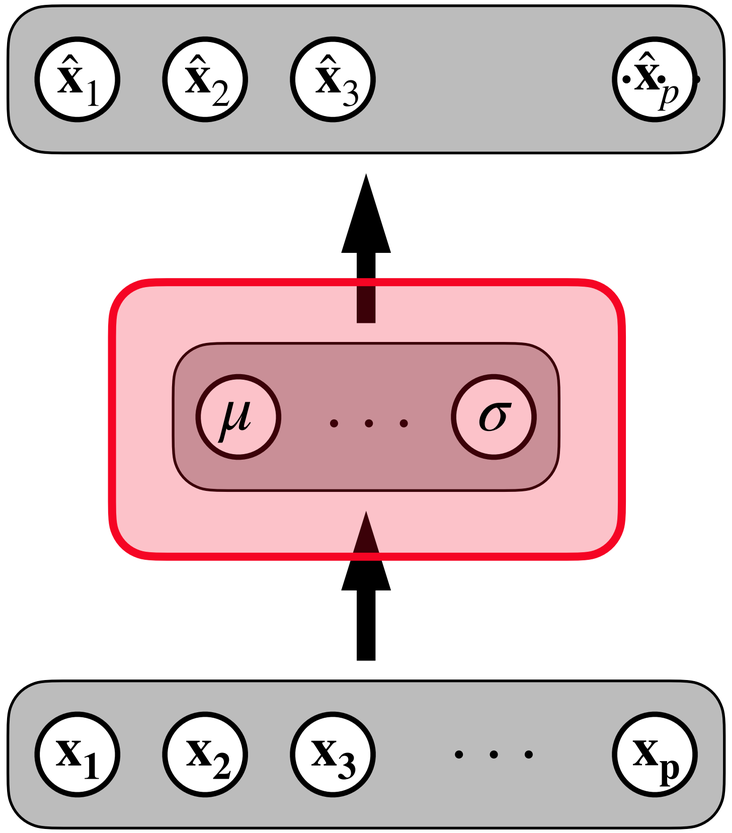
<u>Encodeur</u>

Estimer

$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Auto-encodeur variationnels

Reparamétrisation de la couche cachée:



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

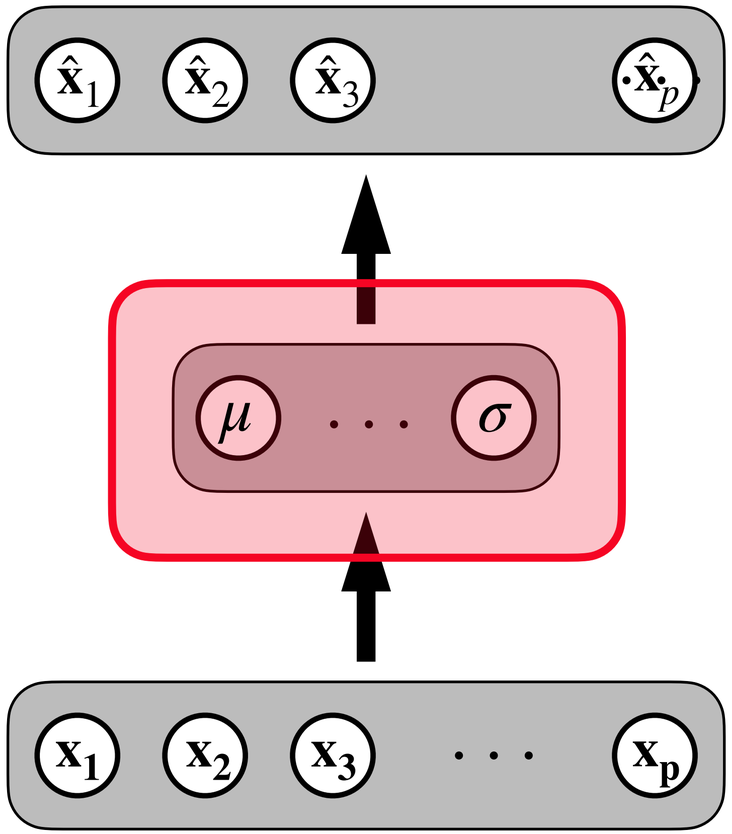
Estimer

$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Supposons que $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

Auto-encodeur variationnels

Reparamétrisation de la couche cachée:



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

Estimer

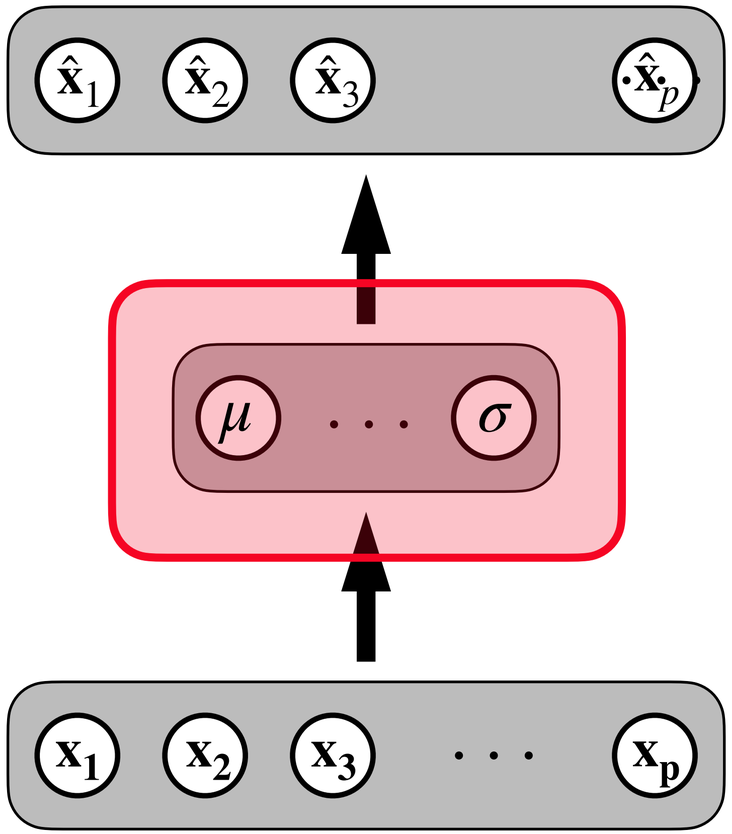
$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Supposons que $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

Alors on peut écrire $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0,1)$.

Auto-encodeur variationnels

Reparamétrisation de la couche cachée:



Décodeur

Estimer

$p_\theta(\mathbf{x} \mid \mathbf{z})$

Encodeur

Estimer

$q_\phi(\mathbf{z} \mid \mathbf{x})$

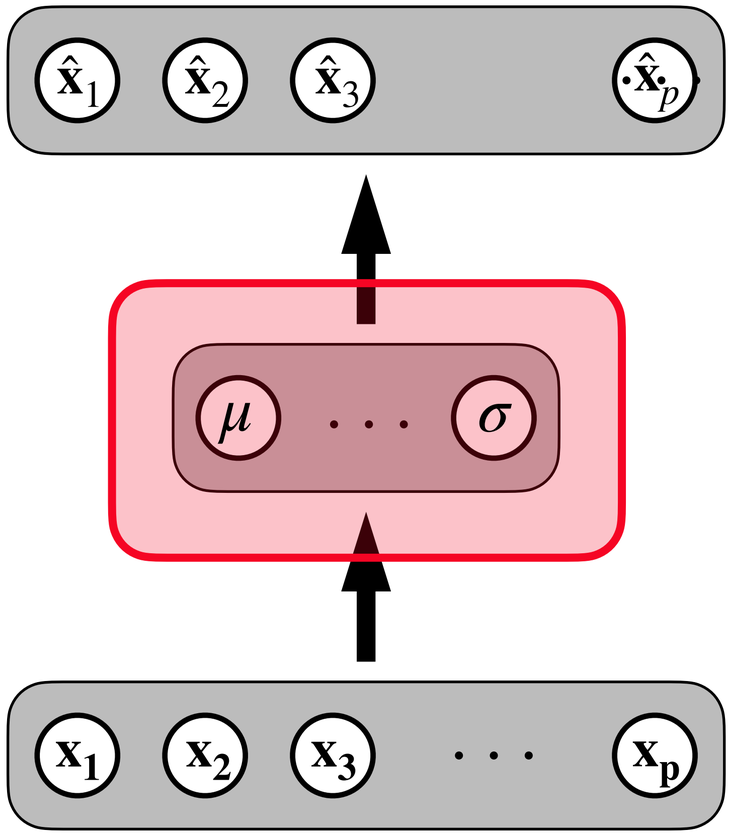
Supposons que $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

Alors on peut écrire $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0,1)$.

Il ne suffit qu'à apprendre les paramètres μ et σ!

Auto-encodeur variationnels

Reparamétrisation de la couche cachée:



Décodeur

Estimer

$p_\theta(\mathbf{x} \mid \mathbf{z})$

Encodeur

Estimer

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Supposons que $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

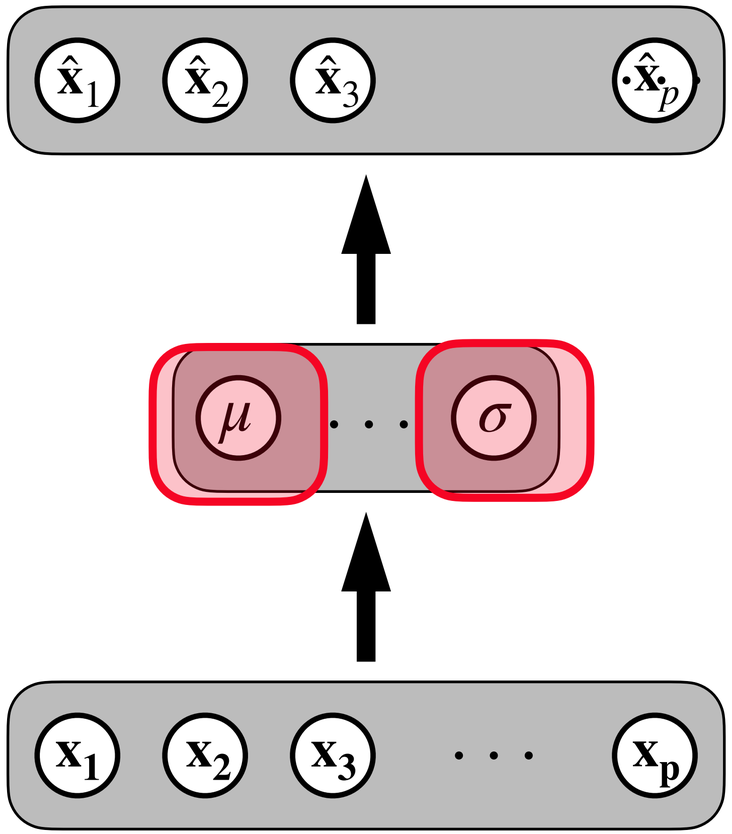
Alors on peut écrire $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0,1)$.

Il ne suffit qu'à apprendre les paramètres μ et σ!

Le réseau de neurones va s'en occuper!

Auto-encodeur variationnels

Reparamétrisation de la couche cachée:



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

Estimer

$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Supposons que $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

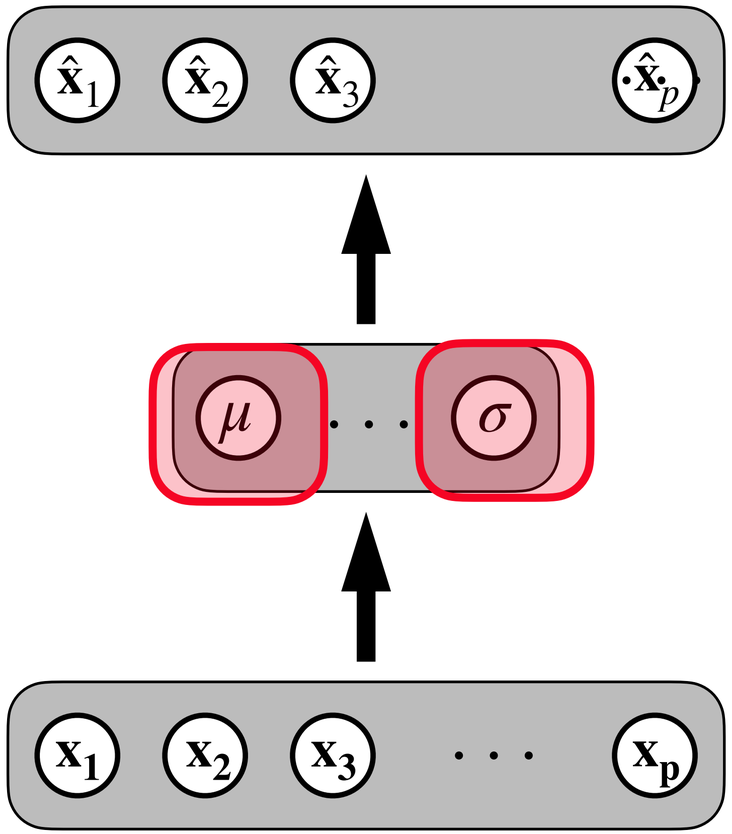
Alors on peut écrire $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0,1)$.

Il ne suffit qu'à apprendre les paramètres μ et σ!

Le réseau de neurones va s'en occuper!

Auto-encodeur variationnels

Reparamétrisation de la couche cachée:



Décodeur

Estimer

$p_\theta(\mathbf{x} \mid \mathbf{z})$

Encodeur

Estimer

$q_\phi(\mathbf{z} \mid \mathbf{x})$

Supposons que $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$.

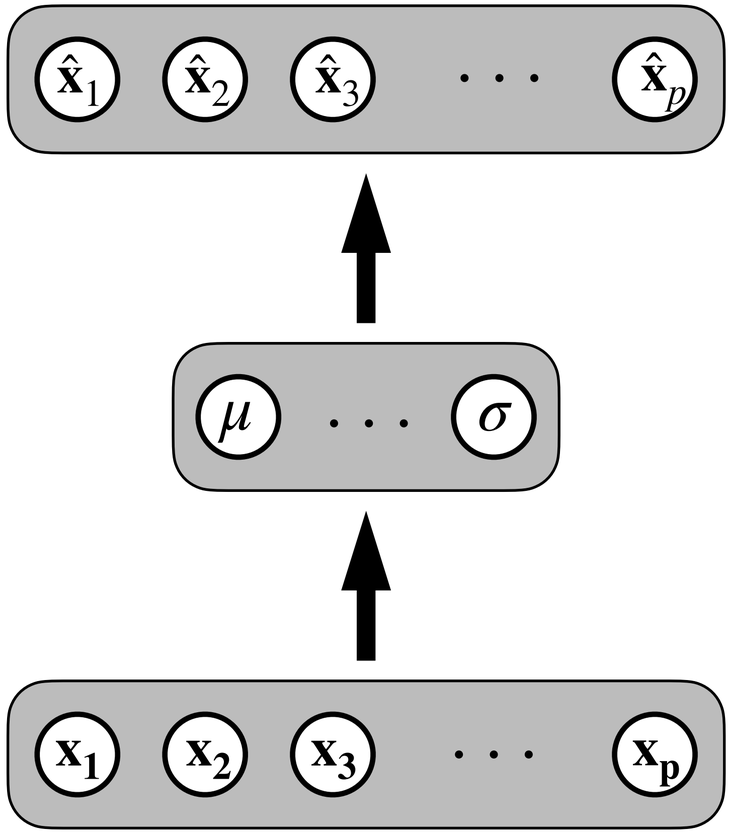
Alors on peut écrire $z = \mu + \sigma\epsilon$ où $\epsilon \sim \mathcal{N}(0,1)$.

Il ne suffit qu'à apprendre les paramètres μ et σ!

Le réseau de neurones va s'en occuper!

Nous pouvons dès lors échantillonner directement des variables latentes, et donc générer de nouvelles observations!

Auto-encodeur variationnels



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

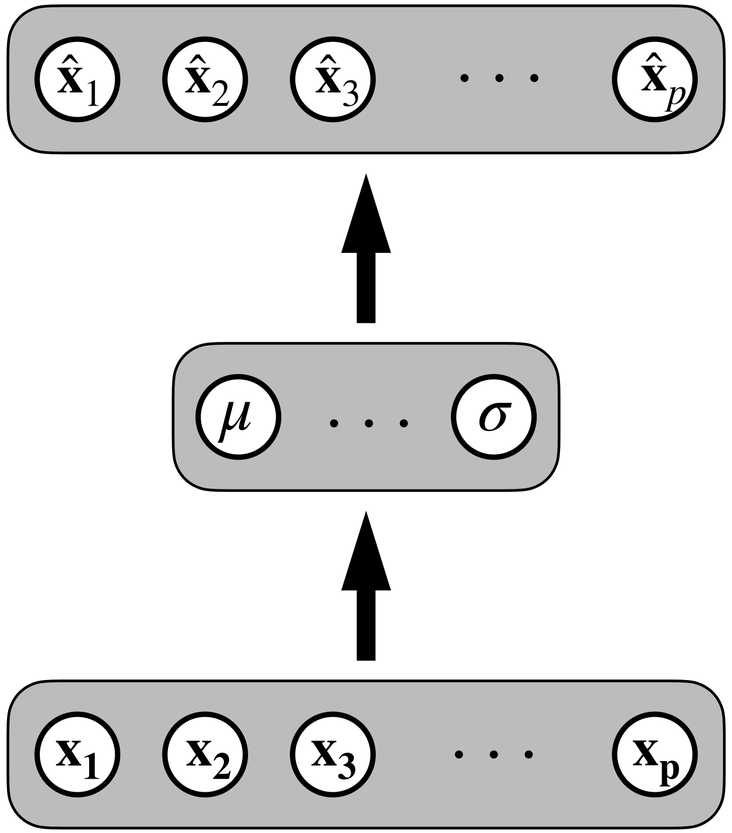
Encodeur

Estimer

$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Auto-encodeur variationnels

Les VAEs sont des modèles génératifs.



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

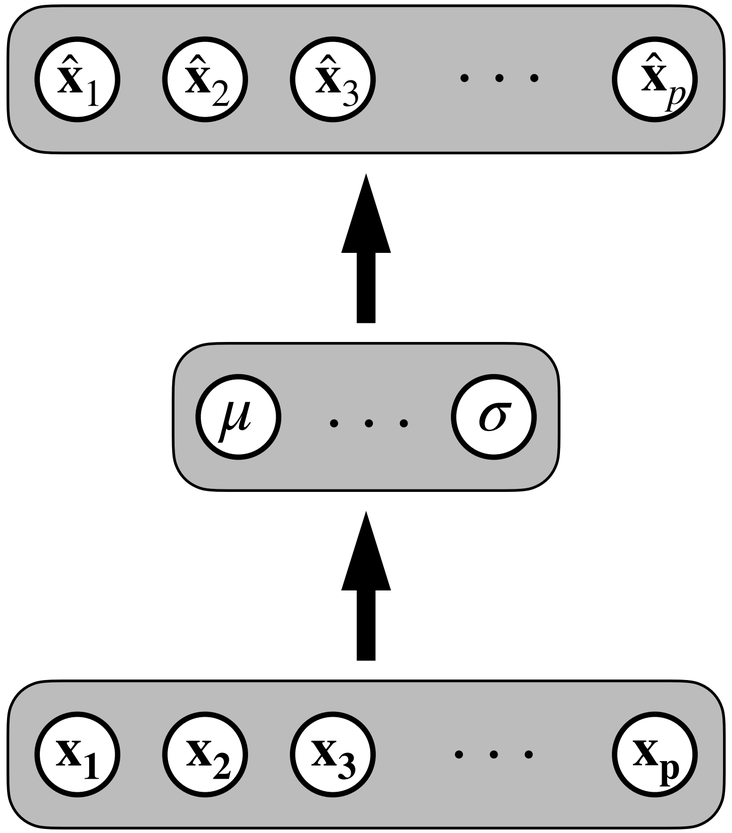
Encodeur

Estimer

$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Auto-encodeur variationnels

Les VAEs sont des modèles génératifs.



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

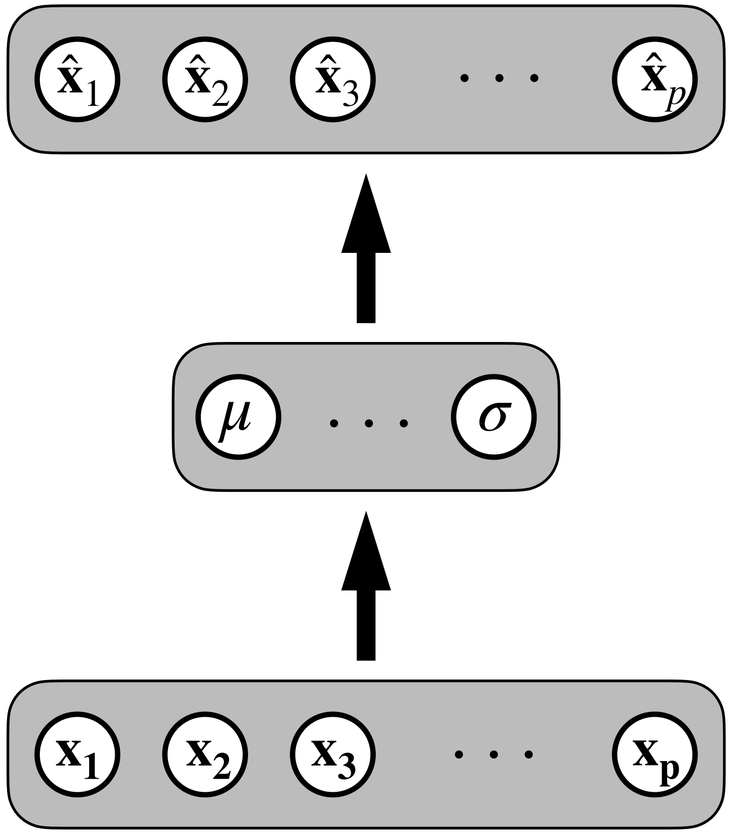
Estimer

$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Marche à suivre

Auto-encodeur variationnels

Les VAEs sont des modèles génératifs.



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

Estimer

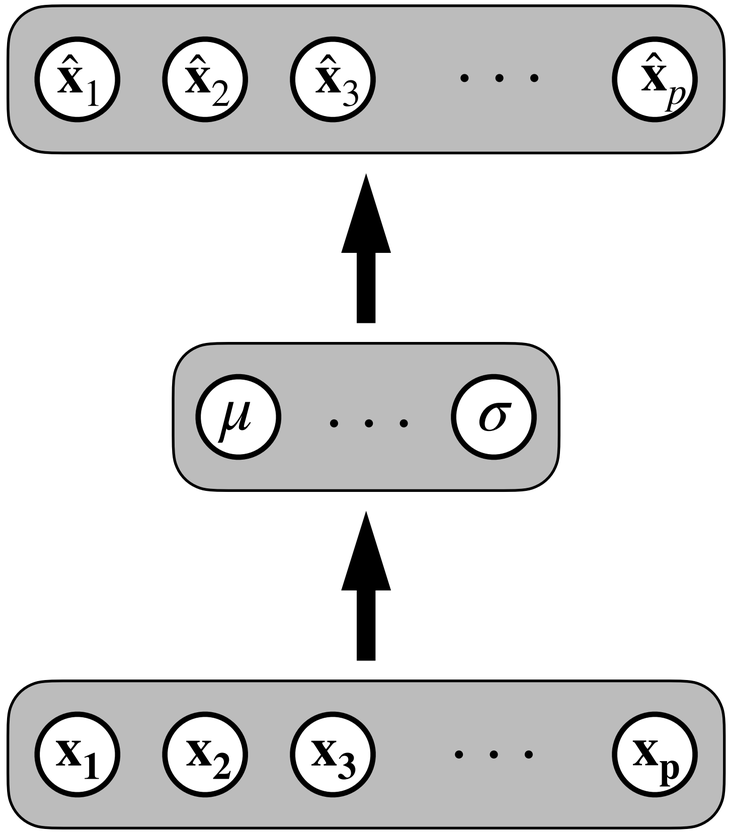
$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Marche à suivre

1. Entraîner le modèle pour apprendre:

Auto-encodeur variationnels

Les VAEs sont des modèles génératifs.



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

Estimer

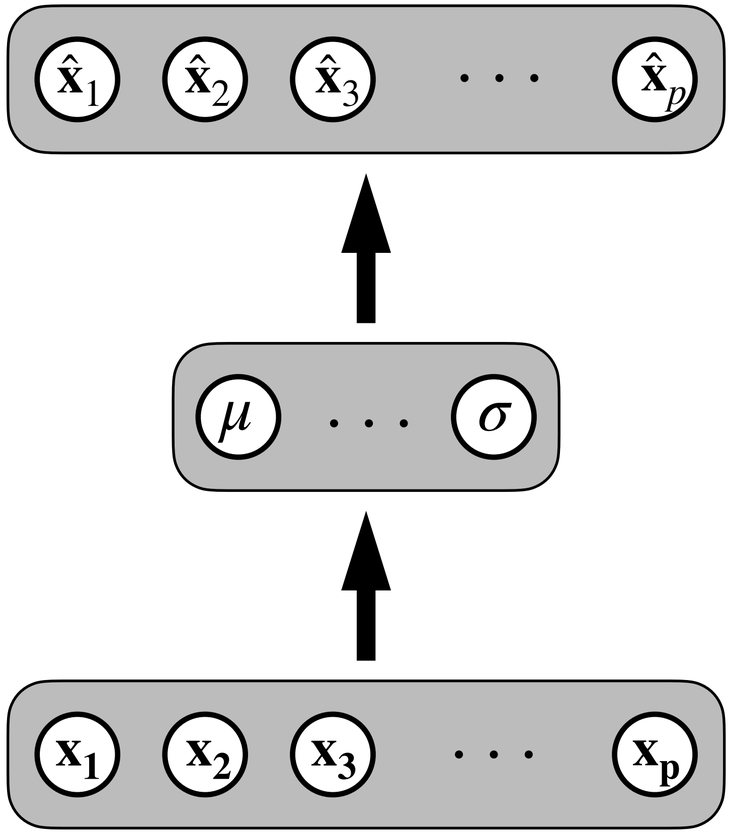
$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Marche à suivre

1. Entraîner le modèle pour apprendre:
   - $p_\theta(\mathbf{x} \mid \mathbf{z})$

Auto-encodeur variationnels

Les VAEs sont des modèles génératifs.



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

Estimer

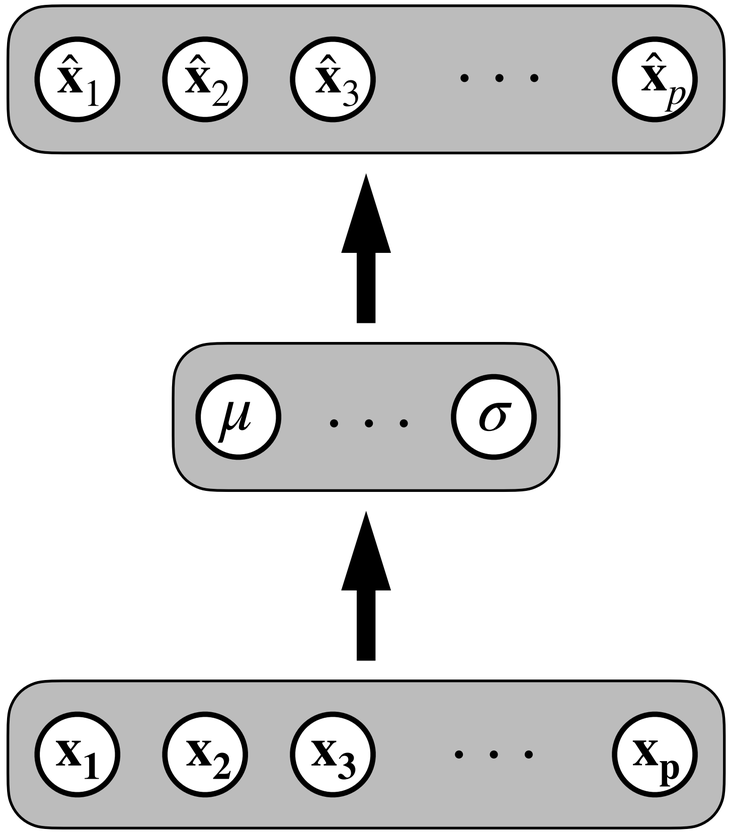
$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Marche à suivre

1. Entraîner le modèle pour apprendre:
   - $p_\theta(\mathbf{x} \mid \mathbf{z})$
   - $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$

Auto-encodeur variationnels

Les VAEs sont des modèles génératifs.



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

Estimer

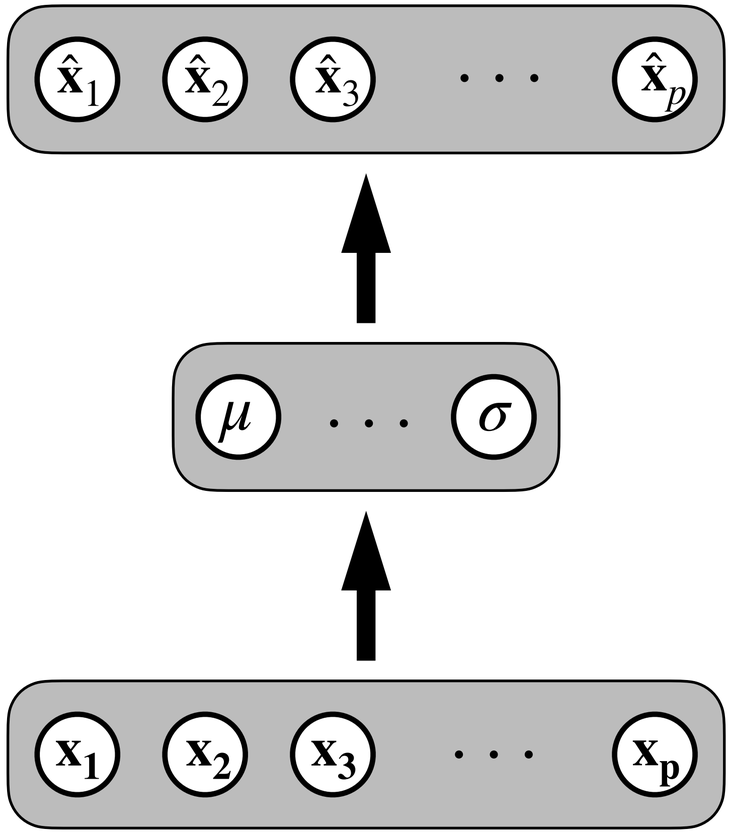
$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Marche à suivre

1. Entraîner le modèle pour apprendre:
   - $p_\theta(\mathbf{x} \mid \mathbf{z})$
   - $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$
2. Générer des ϵ et après construire $\mathbf{z} \sim \mathbf{P}(\mathbf{z})$.

Auto-encodeur variationnels

Les VAEs sont des modèles génératifs.



Décodeur

Estimer

$$p_\theta(\mathbf{x} \mid \mathbf{z})$$

Encodeur

Estimer

$$q_\phi(\mathbf{z} \mid \mathbf{x})$$

Marche à suivre

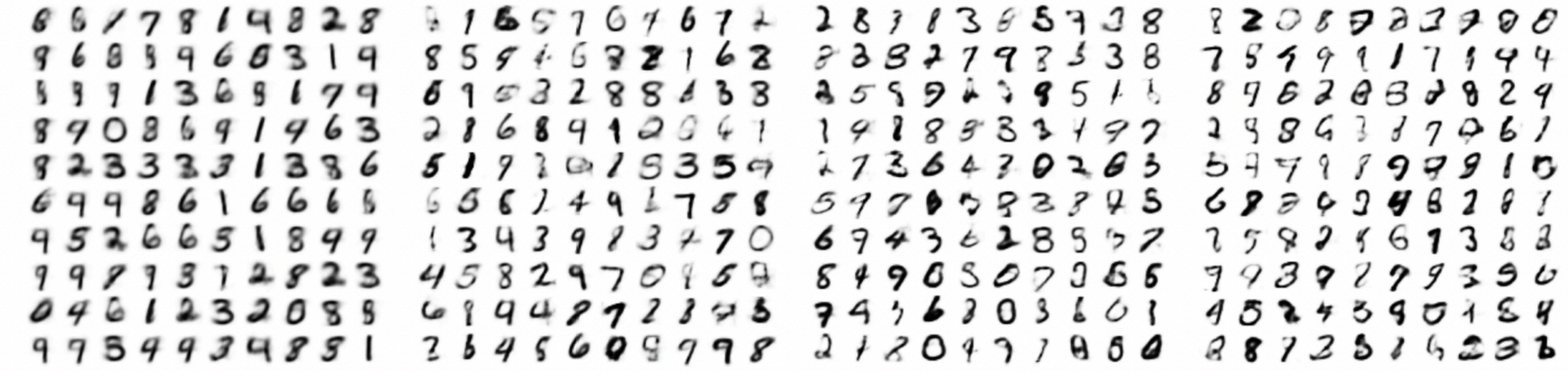
1. Entraîner le modèle pour apprendre:
   - $p_\theta(\mathbf{x} \mid \mathbf{z})$
   - $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2)$
2. Générer des ϵ et après construire $\mathbf{z} \sim \mathbf{P}(\mathbf{z})$.
3. Construire $\mathbf{x}$ à partir de $p_\theta(\mathbf{x} \mid \mathbf{z})$.

Auto-encodeur variationnels

Un exemple: Génération de nombres

Auto-encodeur variationnels

Un exemple: Génération de nombres



Normale multivariée ($\mathbb{R}^2$)

Normale multivariée ($\mathbb{R}^5$)

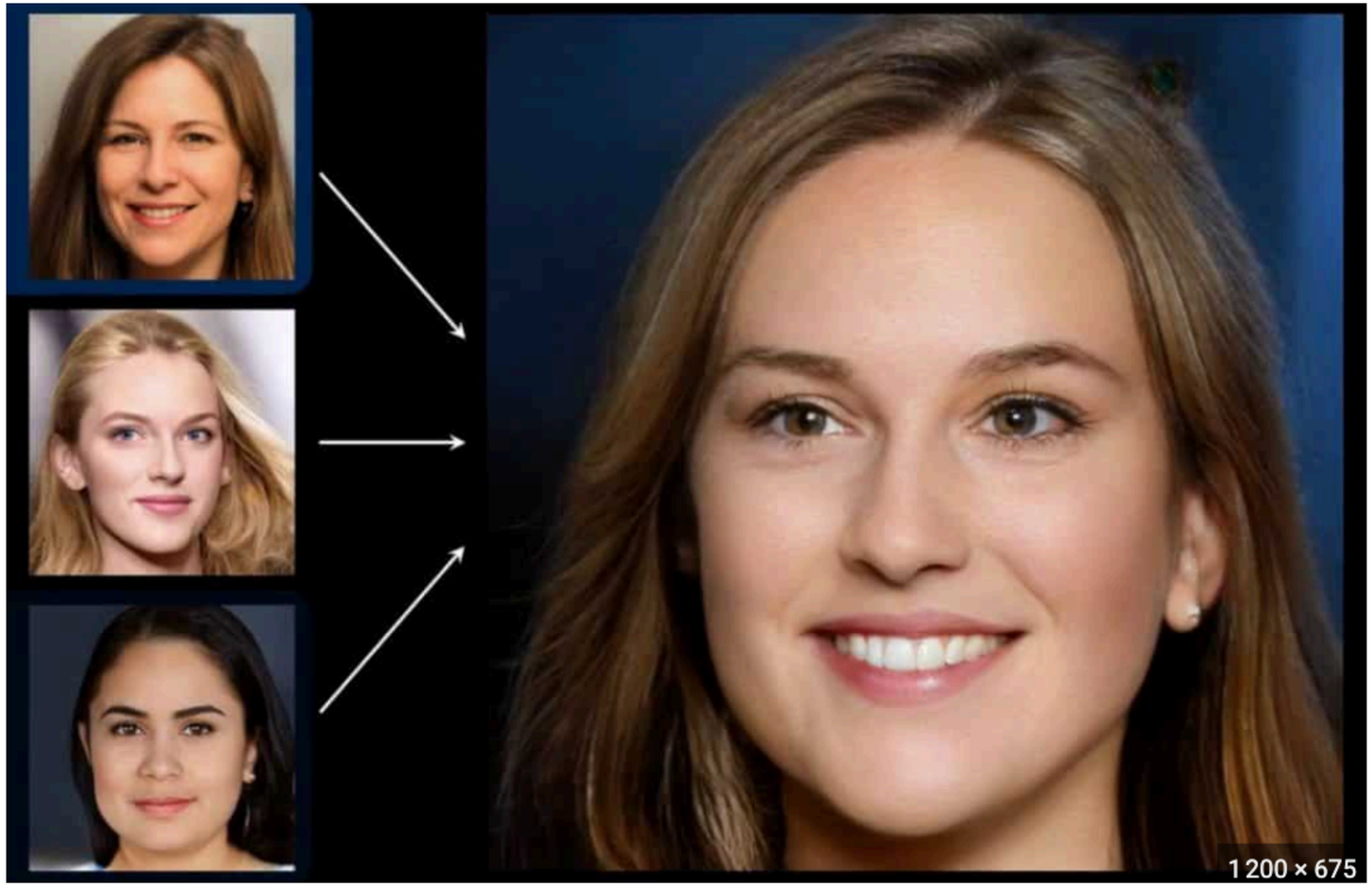
Normale multivariée ($\mathbb{R}^{10}$)

Normale multivariée ($\mathbb{R}^{20}$)

Réseaux antagonistes génératifs (GAN)

GANs - Introduction

Bien connus pour la génération d’images



Note historique

- Façon d’entraîner un modèle génératif (par exemple un CNN “inversé”)
- Venant de chercheurs à l’Université de Montréal (2014)
- Les premiers modèles à générer des images complexes de grande qualité
- Sont petits à petits remplacés notamment par les modèles de diffusion
  - L’étude des GANs nous a appris sur l’optimisation des “modèles à deux joueurs”

GANs — Intuition

GANs — Intuition

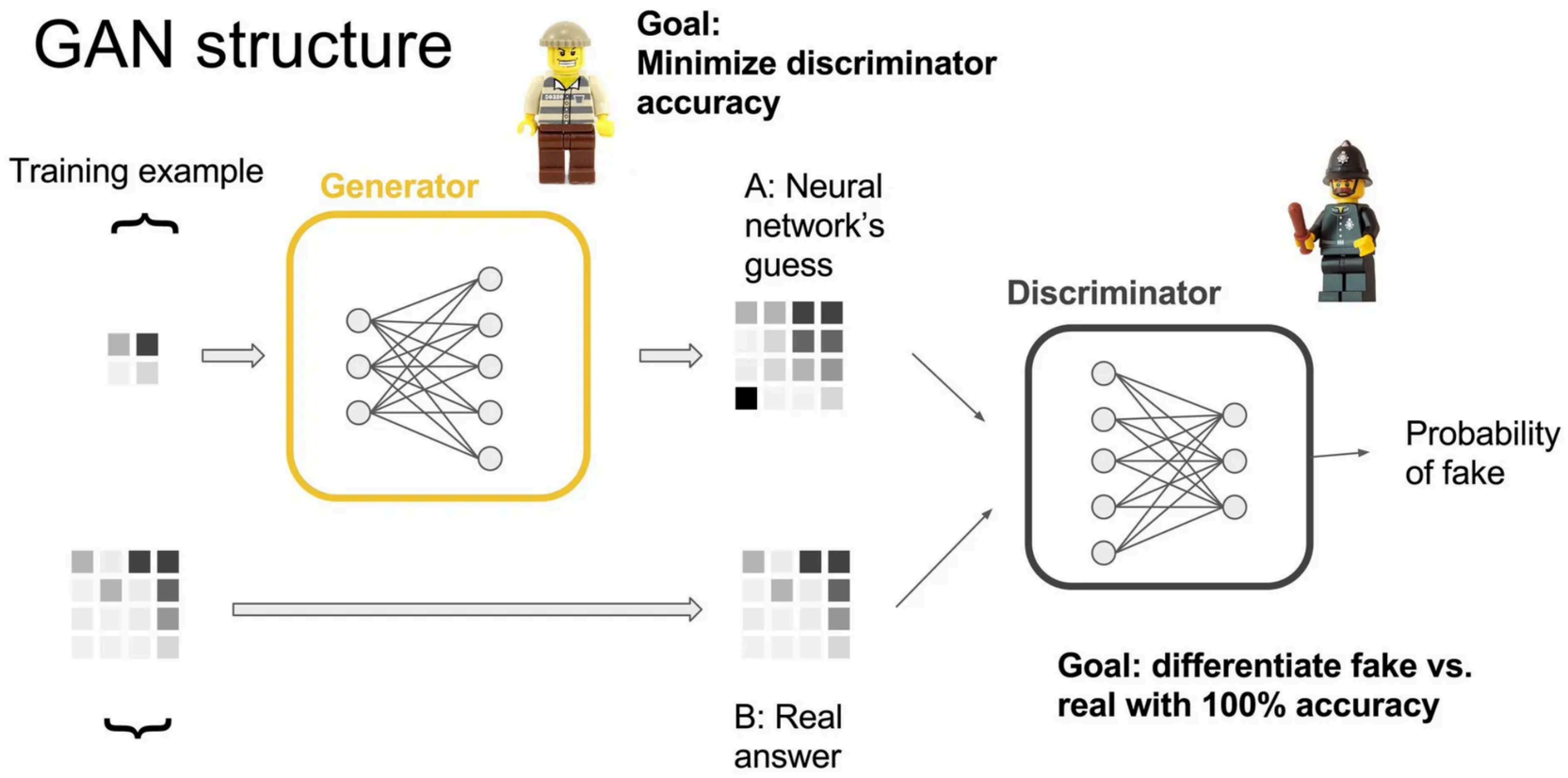
- 2 joueurs (réseaux)
  - Générateur: Le premier apprend à générer une image
  - Discriminator: Le second apprend à déterminer si une image est réelle ou pas

GANs — Intuition

- 2 joueurs (réseaux)
  - Générateur: Le premier apprend à générer une image
  - Discriminator: Le second apprend à déterminer si une image est réelle ou pas
- Le jeu (à chaque ronde):
  - Le second joueur reçoit une image. Le joueur doit déterminer si l’image a été générée ou vient du jeu de données d’entraînement.
    - En fonction de la réponse, les joueurs se mettent à jour (leurs poids)

GANs - Introduction

Comment ça marche?



GANs - Formalisme

$$\boldsymbol{x} \sim \mathbb{P}_r$$

Data

GANs - Formalisme

$$\boldsymbol{x} \sim \mathbb{P}_r$$

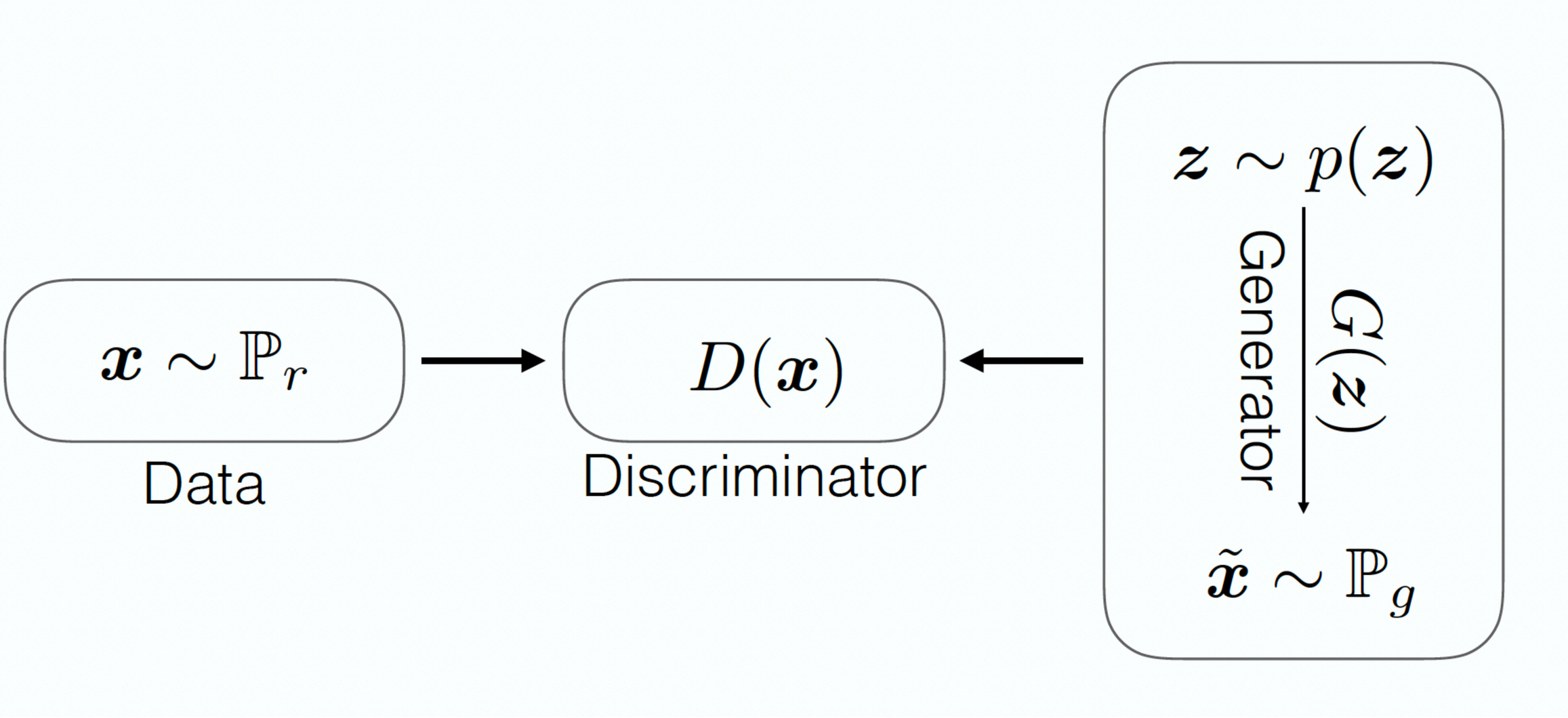
Data

$$\boldsymbol{z} \sim p(\boldsymbol{z})$$

$$\xrightarrow[\text{Generator}]{G(\boldsymbol{z})}$$

$$\tilde{\boldsymbol{x}} \sim \mathbb{P}_g$$

GANs - Formalisme



GANs - Objectifs

Le problème a deux objectifs distincts:

Objectif Discriminateur:

$$\max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\boldsymbol{x}}))].$$

Objectif Générateur:

$$\max_G \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(D(\tilde{\boldsymbol{x}}))].$$

GANs - Formalisme

Formellement, on peut exprimer ce jeu entre le discriminateur *D* et le générateur *G* via un problème min max:

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\boldsymbol{x}}))].$$

GANs - Formalisme

Formellement, on peut exprimer ce jeu entre le discriminateur *D* et le générateur *G* via un problème min max:

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\boldsymbol{x}}))].$$

où:

$\mathbb{P}_r$ est la distribution générant les données observées

GANs - Formalisme

Formellement, on peut exprimer ce jeu entre le discriminateur *D* et le générateur *G* via un problème min max:

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1 - D(\tilde{\boldsymbol{x}}))].$$

où:

$\mathbb{P}_r$ est la distribution générant les données observées

$\mathbb{P}_g$ est la distribution générant les données trafiquées telles que:

$$\tilde{\boldsymbol{x}} = G(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z})$$

où *z* suit une loi normale (par exemple).

En pratique, pour l'entrainement, on alterne entre l'apprentissage de D et G

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

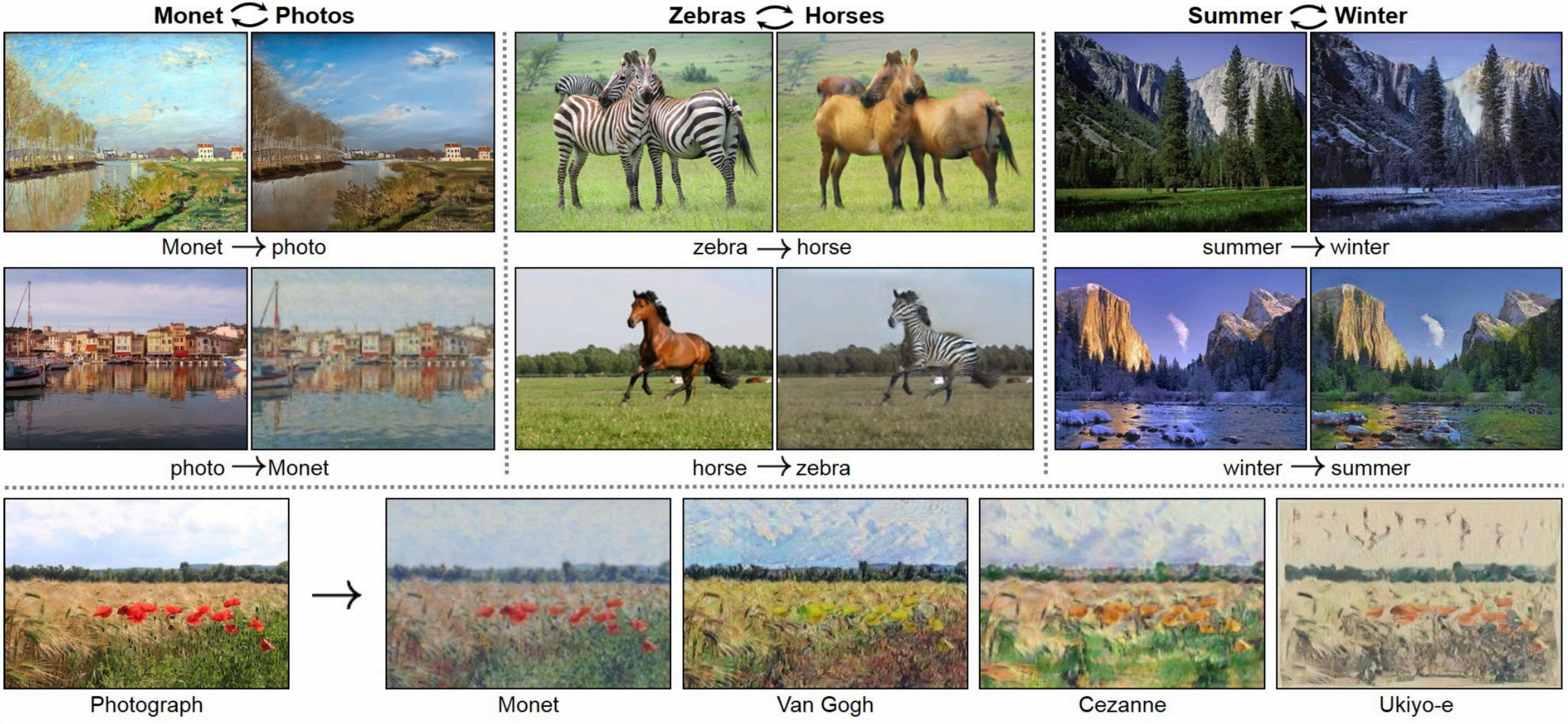
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

GANs - Varia


Monet ⇄ Photos
Zebras ⇄ Horses
Summer ⇄ Winter
Monet → photo
zebra → horse
summer → winter
photo → Monet
horse → zebra
winter → summer
Photograph
Monet
Van Gogh
Cezanne
Ukiyo-e

DALL.E (2)
(Pour le modèle de diffusion)

DALL.E

an espresso machine that makes coffee from human souls, artstation

panda mad scientist mixing sparkling chemicals, artstation

a corgi's head depicted as an explosion of a nebula

a dolphin in an astronaut suit on saturn, artstation

a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese

a teddy bear on a skateboard in times square

DALL.E - Survol

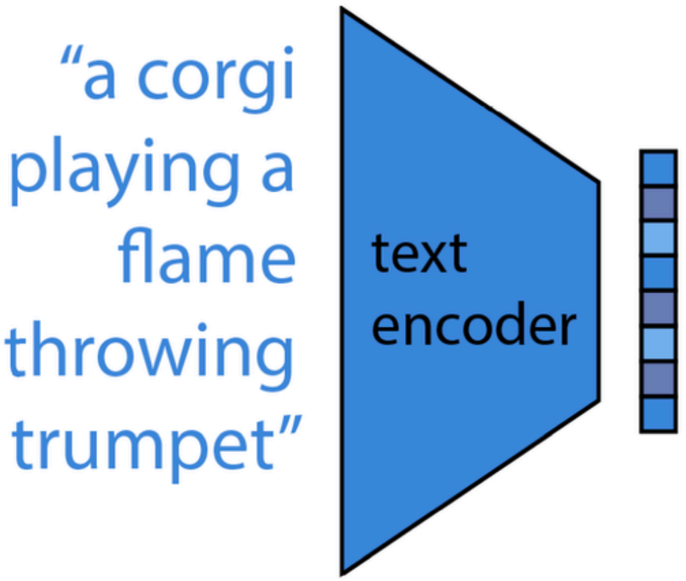
"a corgi playing a flame throwing trumpet"

DALL.E - Survol

“a corgi playing a flame throwing trumpet”

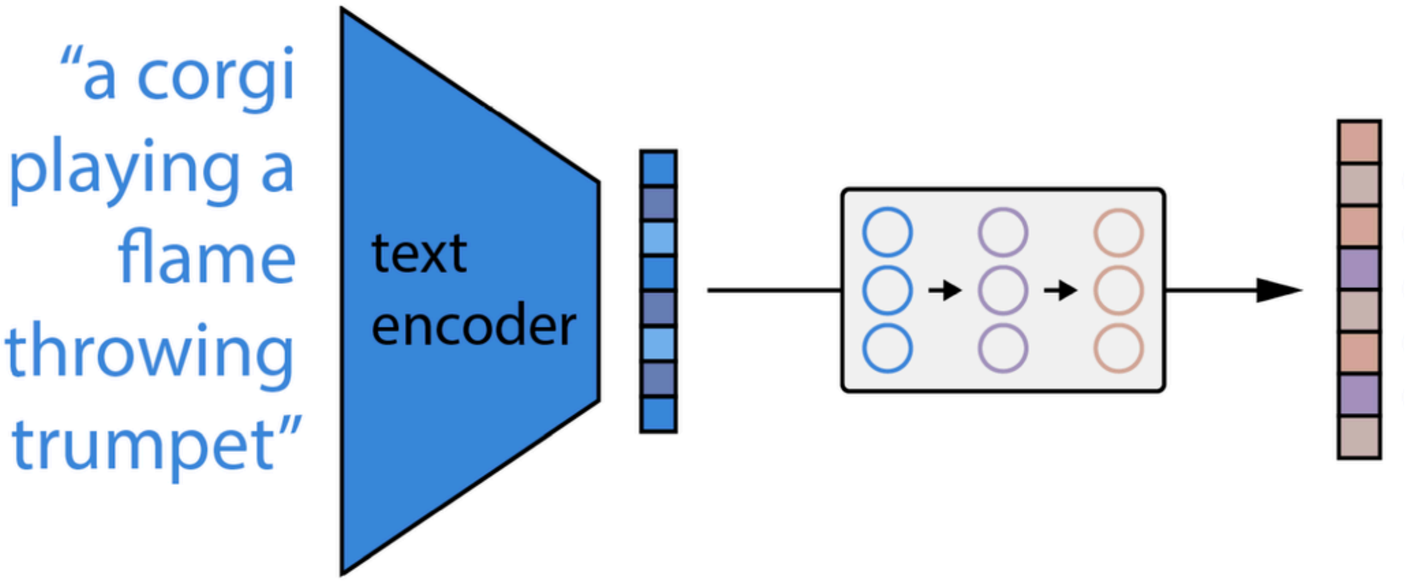
1. En entrée, une phrase (ou *prompt*) de l’image que l’on veut créer.

DALL.E - Survol



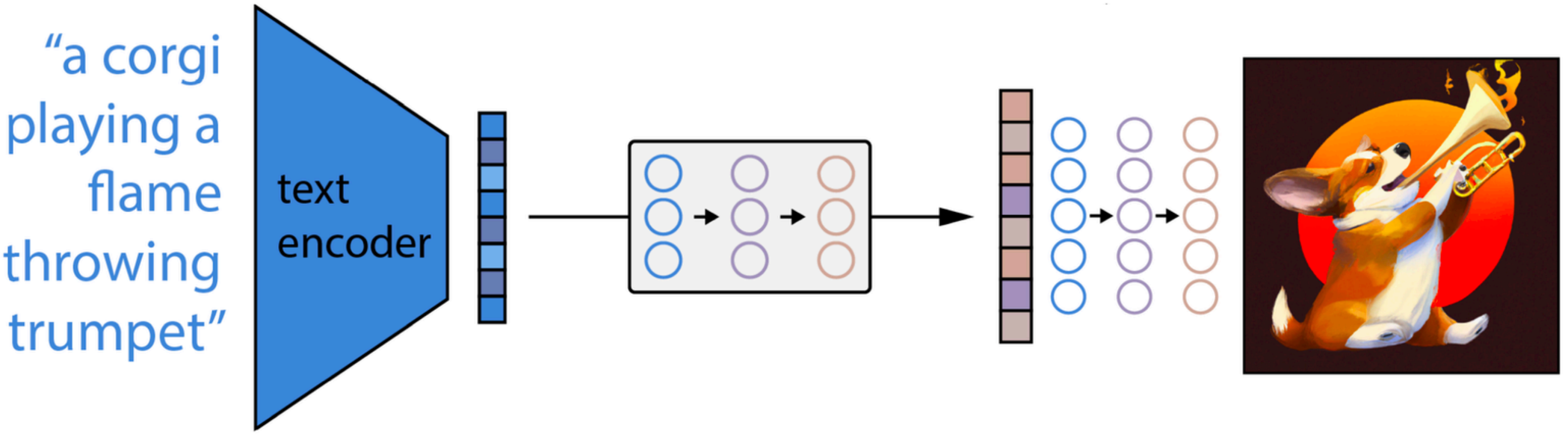
1. En entrée, une phrase (ou *prompt*) de l’image que l’on veut créer.
2. Ce prompt est encodé une représentation latente.

DALL.E - Survol



1. En entrée, une phrase (ou *prompt*) de l’image que l’on veut créer.
2. Ce prompt est encodé une représentation latente.
3. Transformation de la représentation latente du *prompt* dans un espace latent des images.

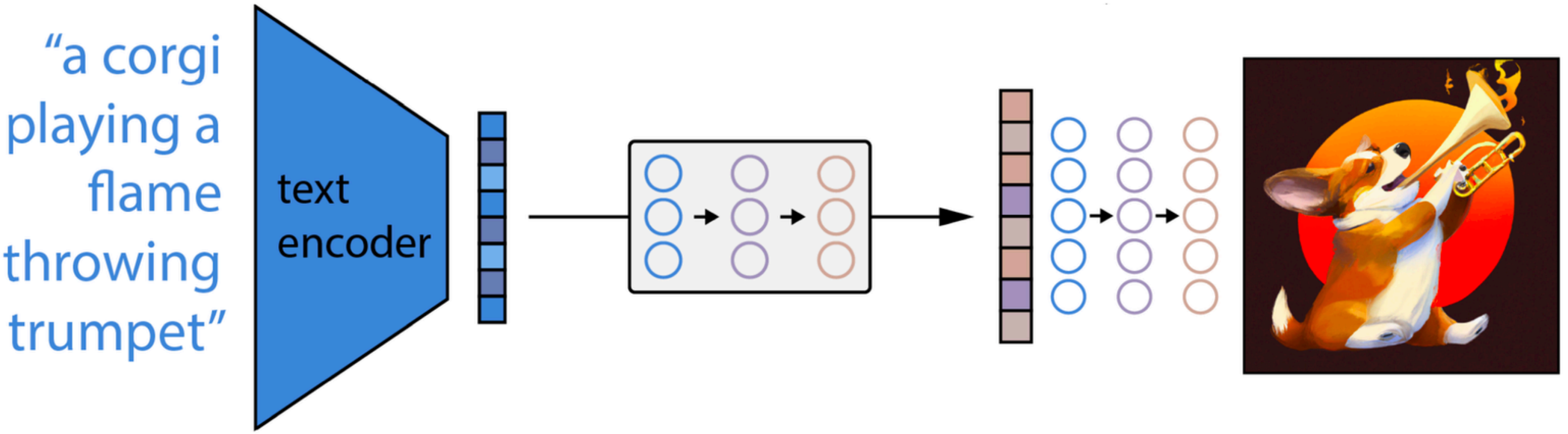
DALL.E - Survol



1. En entrée, une phrase (ou *prompt*) de l’image que l’on veut créer.
2. Ce prompt est encodé une représentation latente.
3. Transformation de la représentation latente du *prompt* dans un espace latent des images.
4. Un décodeur crée une image à partir de cette dernière représentation.

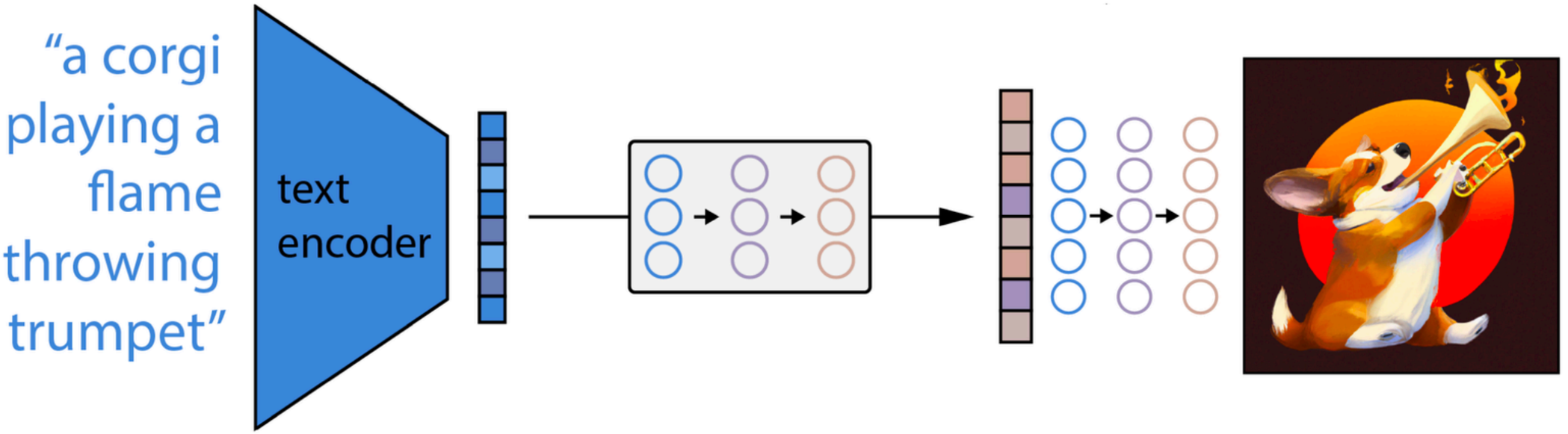
DALL.E - Survol

Chacune de ces étapes fait appel à des techniques bien particulières:



DALL.E - Survol

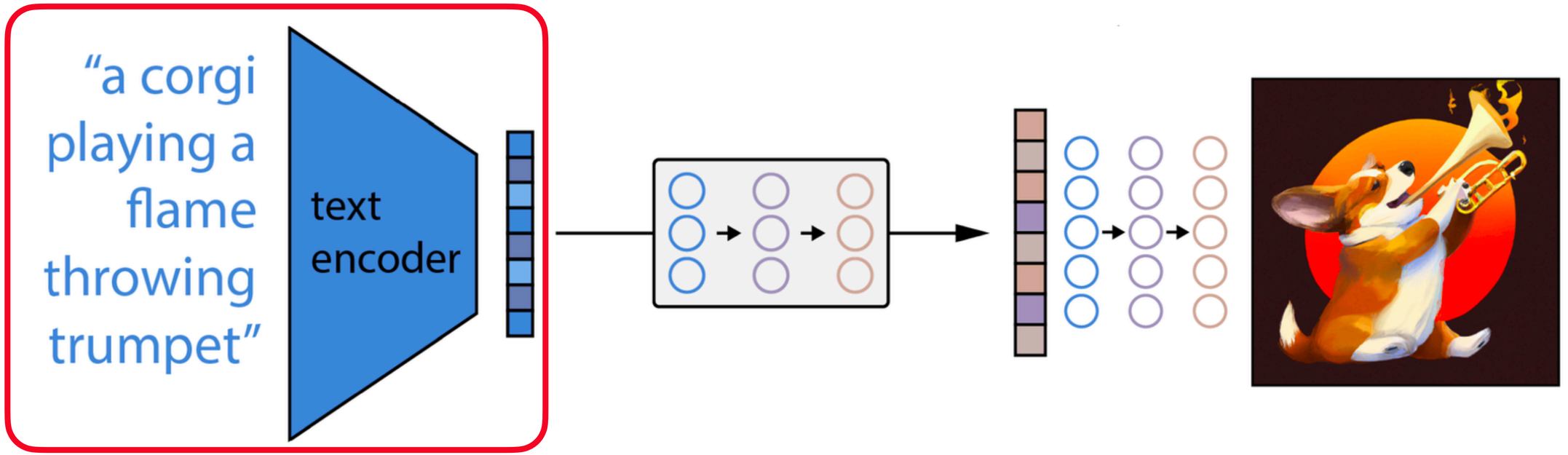
Chacune de ces étapes fait appel à des techniques bien particulières:



1. Contrastive Language-Image Pre-training (CLIP)

DALL.E - Survol

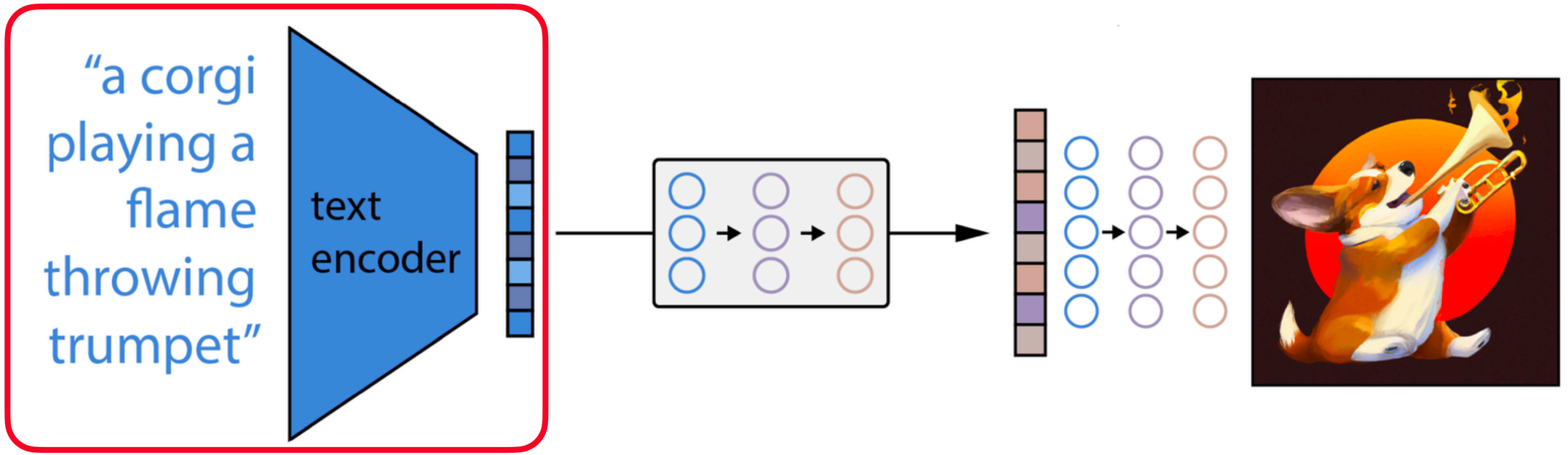
Chacune de ces étapes fait appel à des techniques bien particulières:



1. Contrastive Language-Image Pre-training (CLIP)

DALL.E - Survol

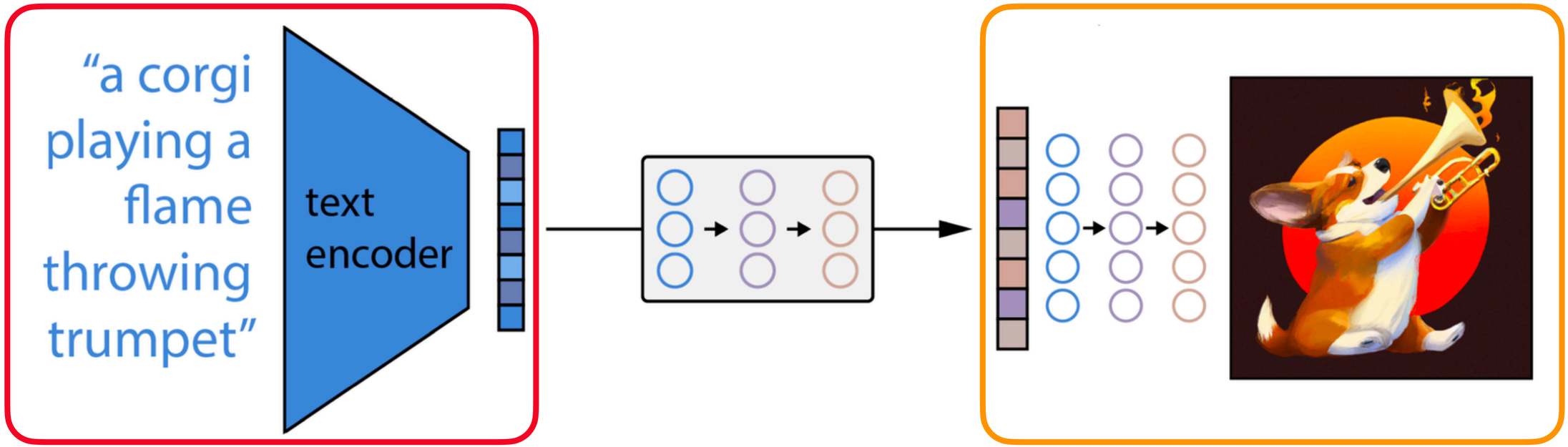
Chacune de ces étapes fait appel à des techniques bien particulières:



1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion

DALL.E - Survol

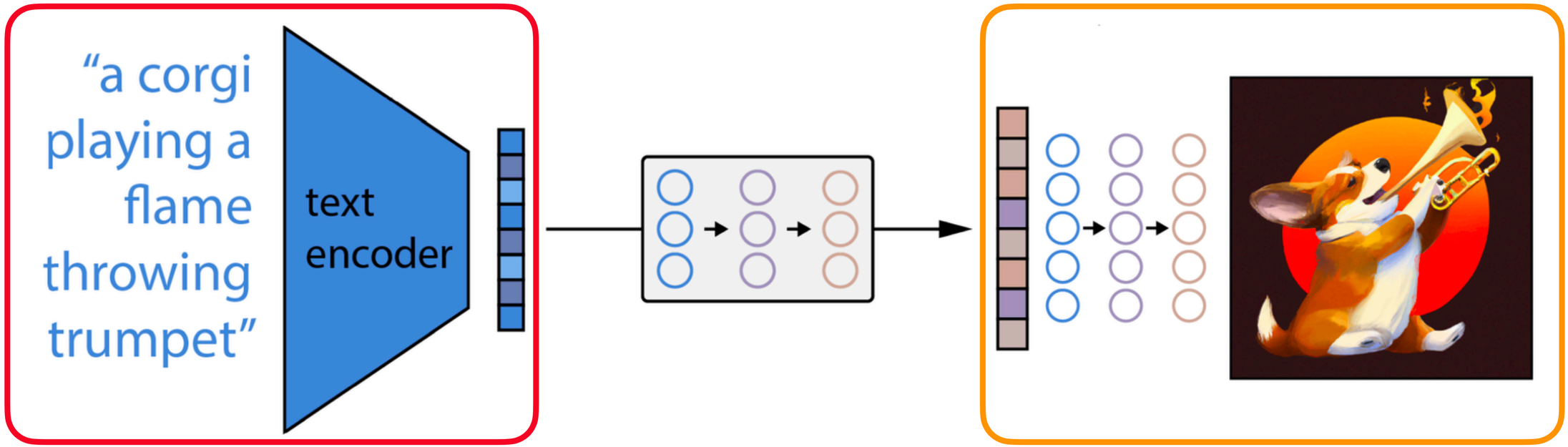
Chacune de ces étapes fait appel à des techniques bien particulières:



1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion

DALL.E - Survol

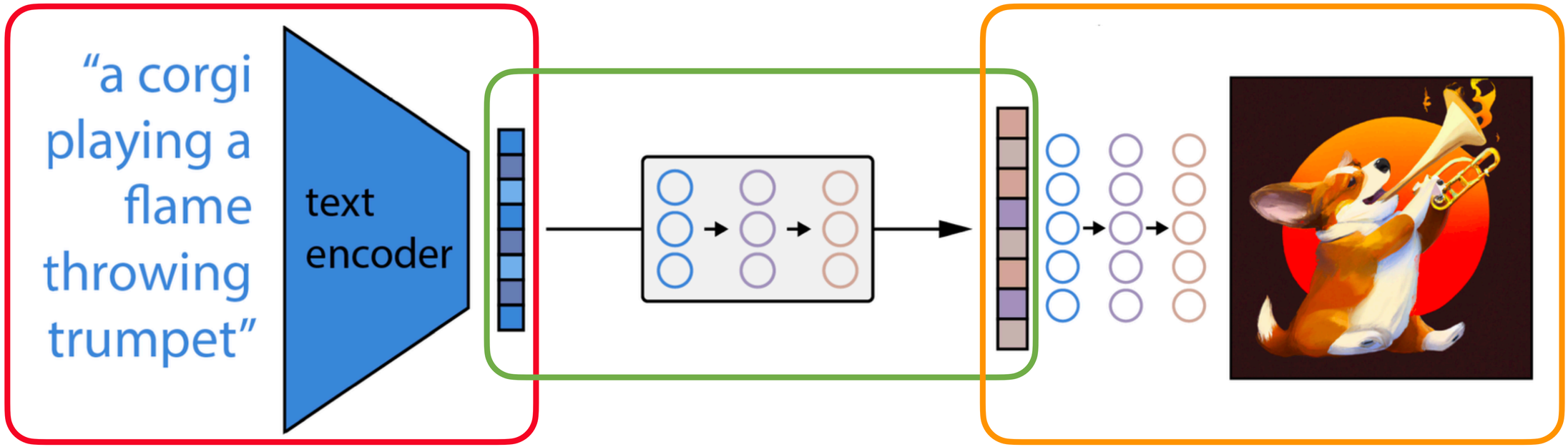
Chacune de ces étapes fait appel à des techniques bien particulières:



1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion
3. Apprentissage de représentations latentes de textes et d’images

DALL.E - Survol

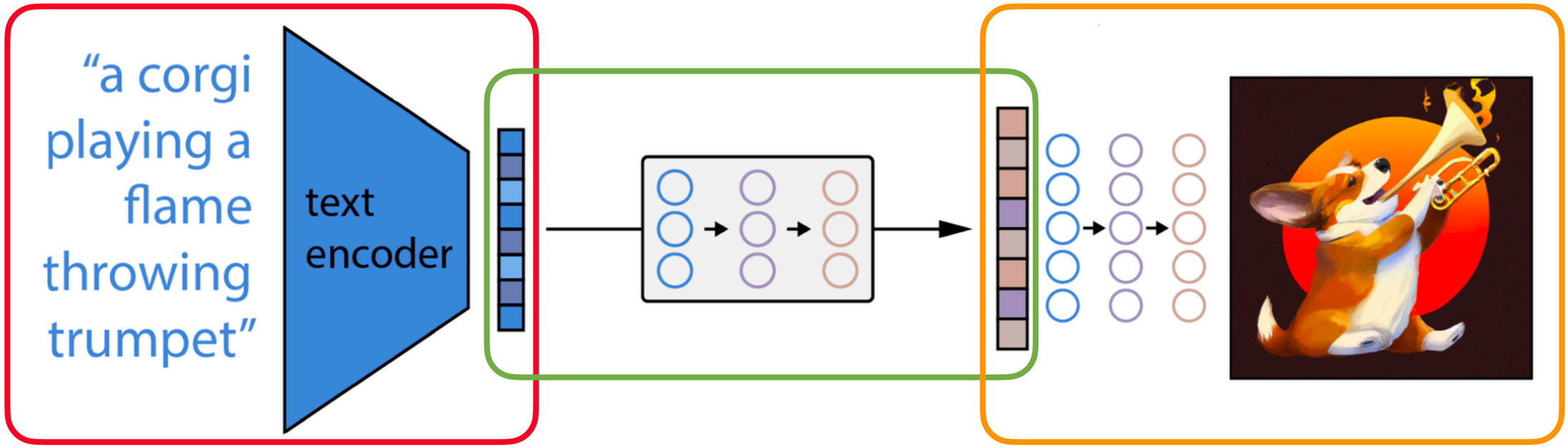
Chacune de ces étapes fait appel à des techniques bien particulières:



1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion
3. Apprentissage de représentations latentes de textes et d’images

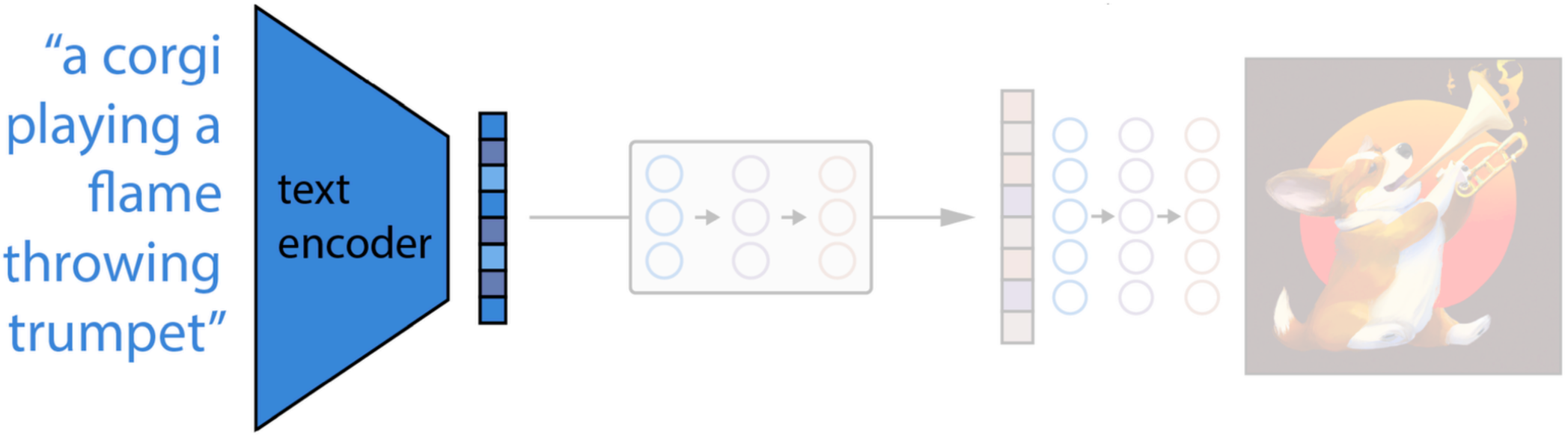
DALL.E - Survol

Chacune de ces étapes fait appel à des techniques bien particulières:

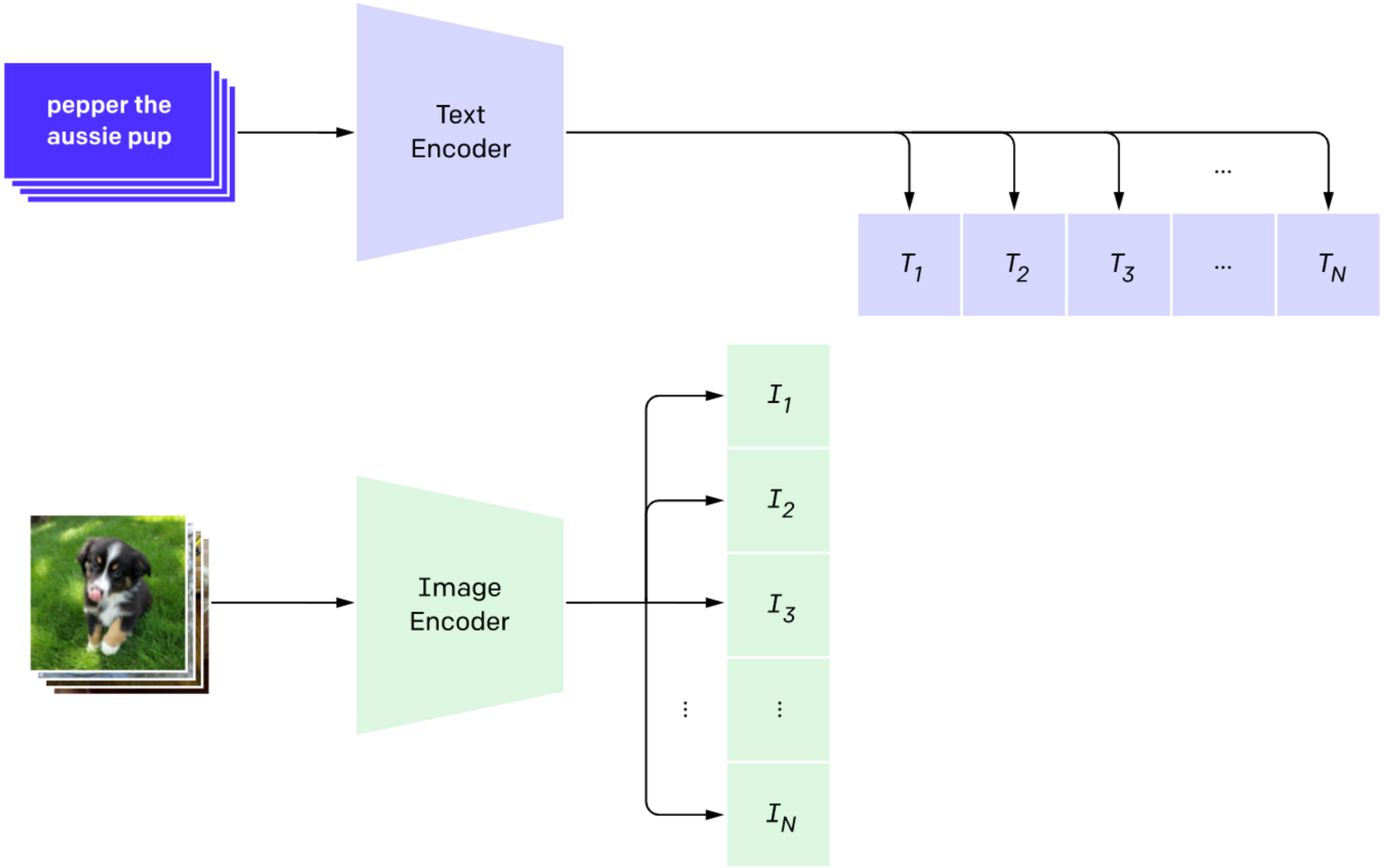


1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion
3. Apprentissage de représentations latentes de textes et d’images
4. Wrap-it up!

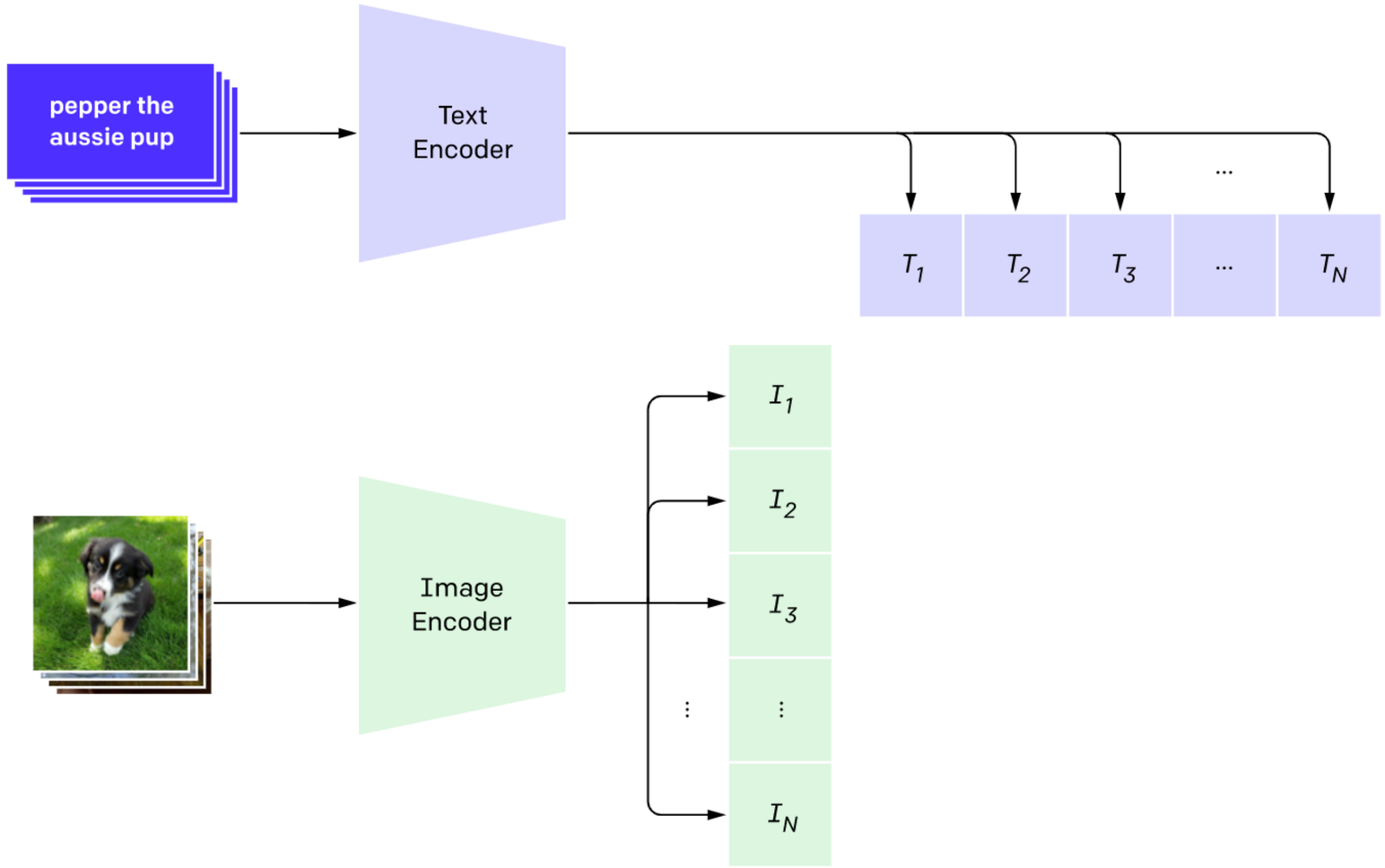
DALL.E - Survol



DALL.E - CLIP

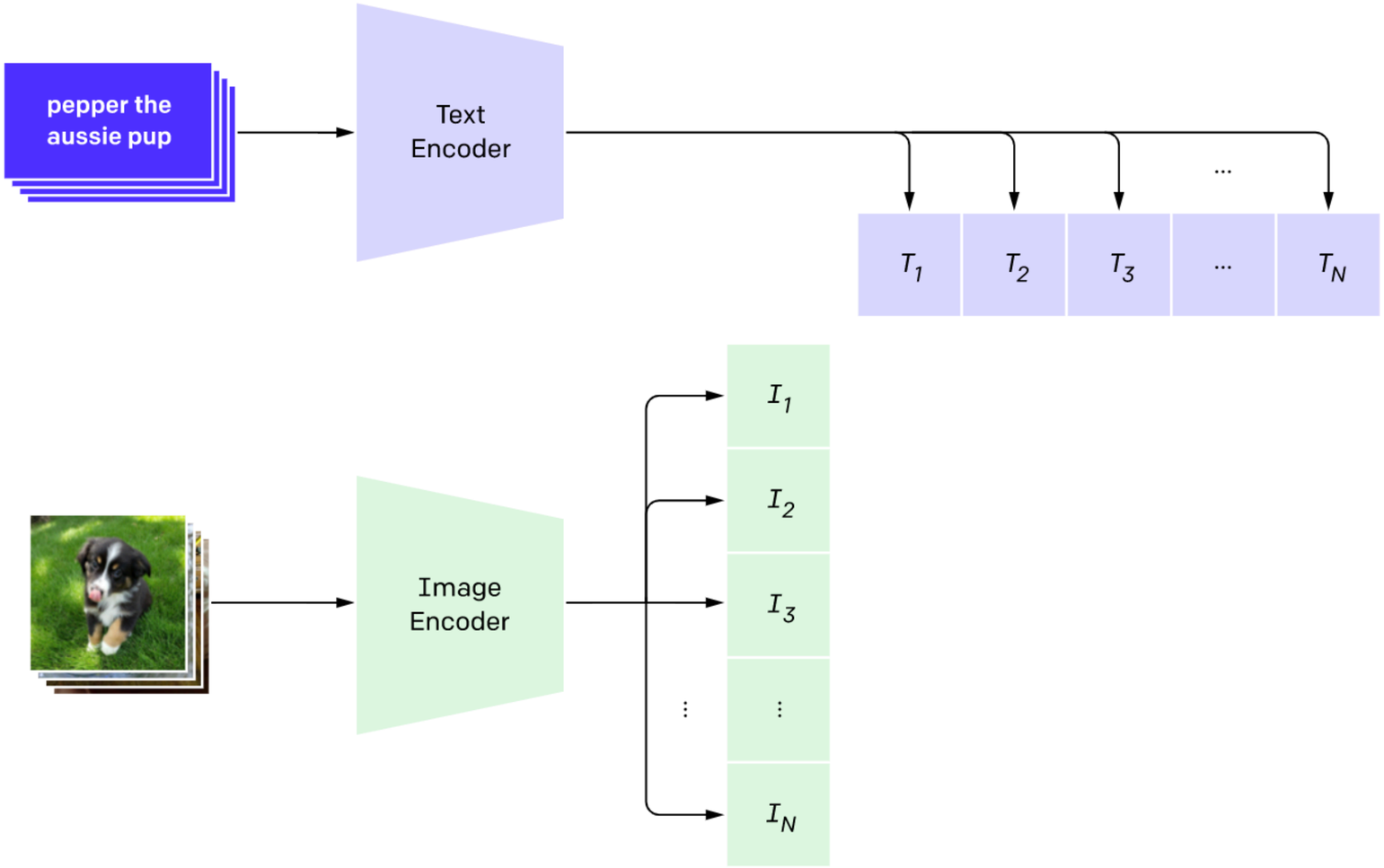

pepper the aussie pup
Text Encoder
...
T_1
T_2
T_3
...
T_N
Image Encoder
I_1
I_2
I_3
I_N

DALL.E - CLIP



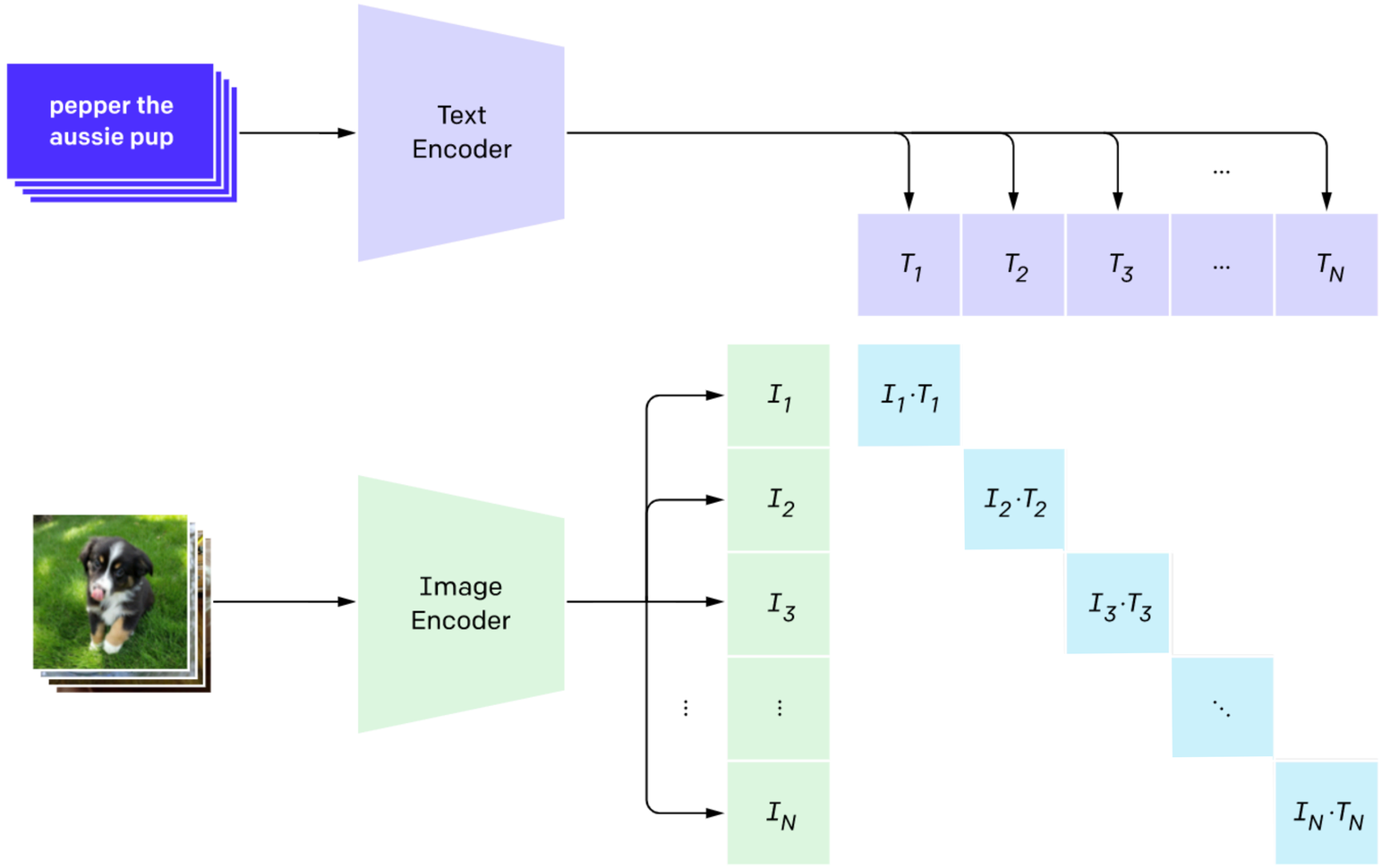
1. Apprentissage représentations latentes pour le texte et les images (d'une batch de taille N)

DALL.E - CLIP



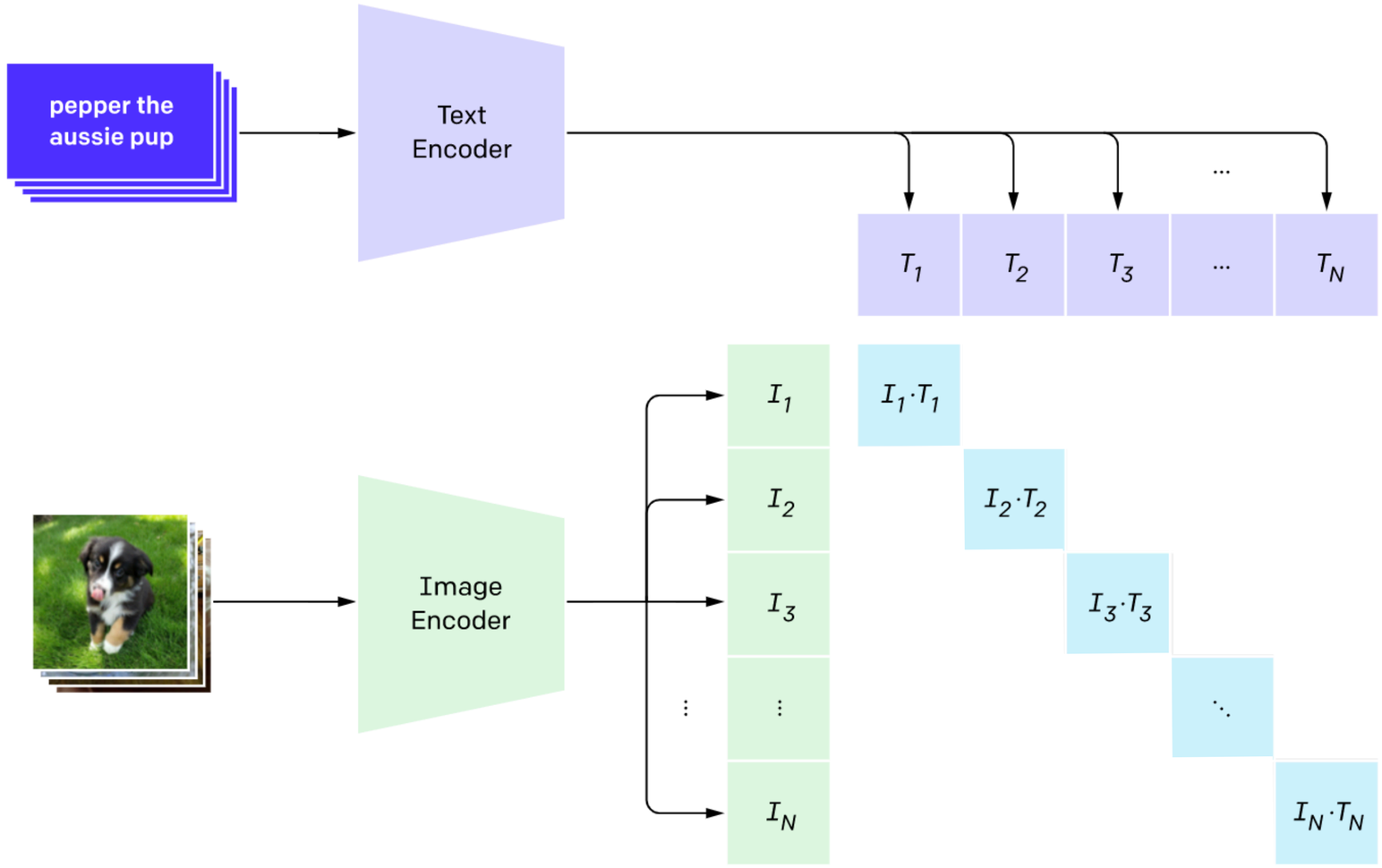
1. Apprentissage représentations latentes pour le texte et les images (d'une batch de taille N)
2. Mesure de similarité

DALL.E - CLIP



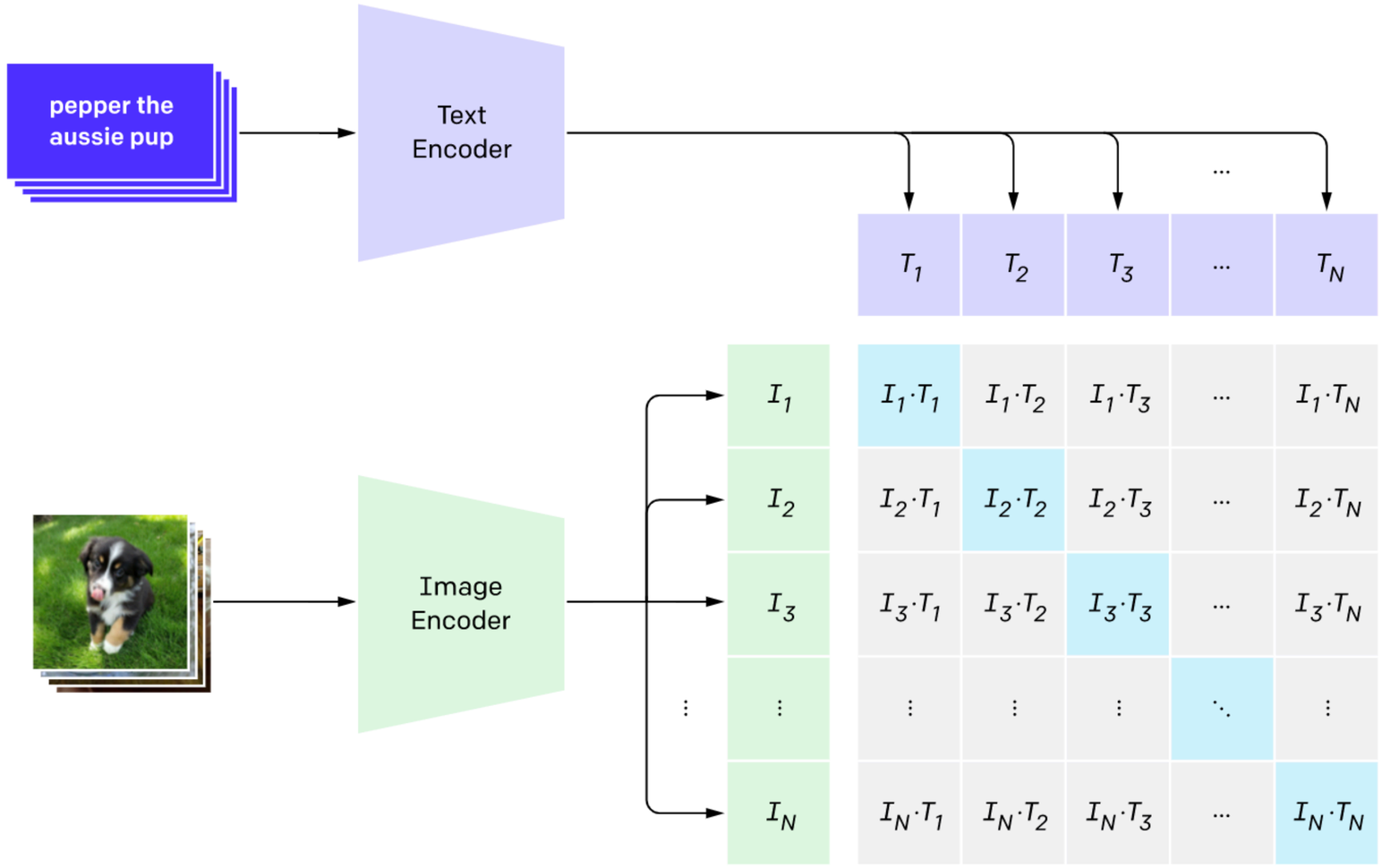
1. Apprentissage représentations latentes pour le texte et les images (d'une batch de taille N)
2. Mesure de similarité

DALL.E - CLIP



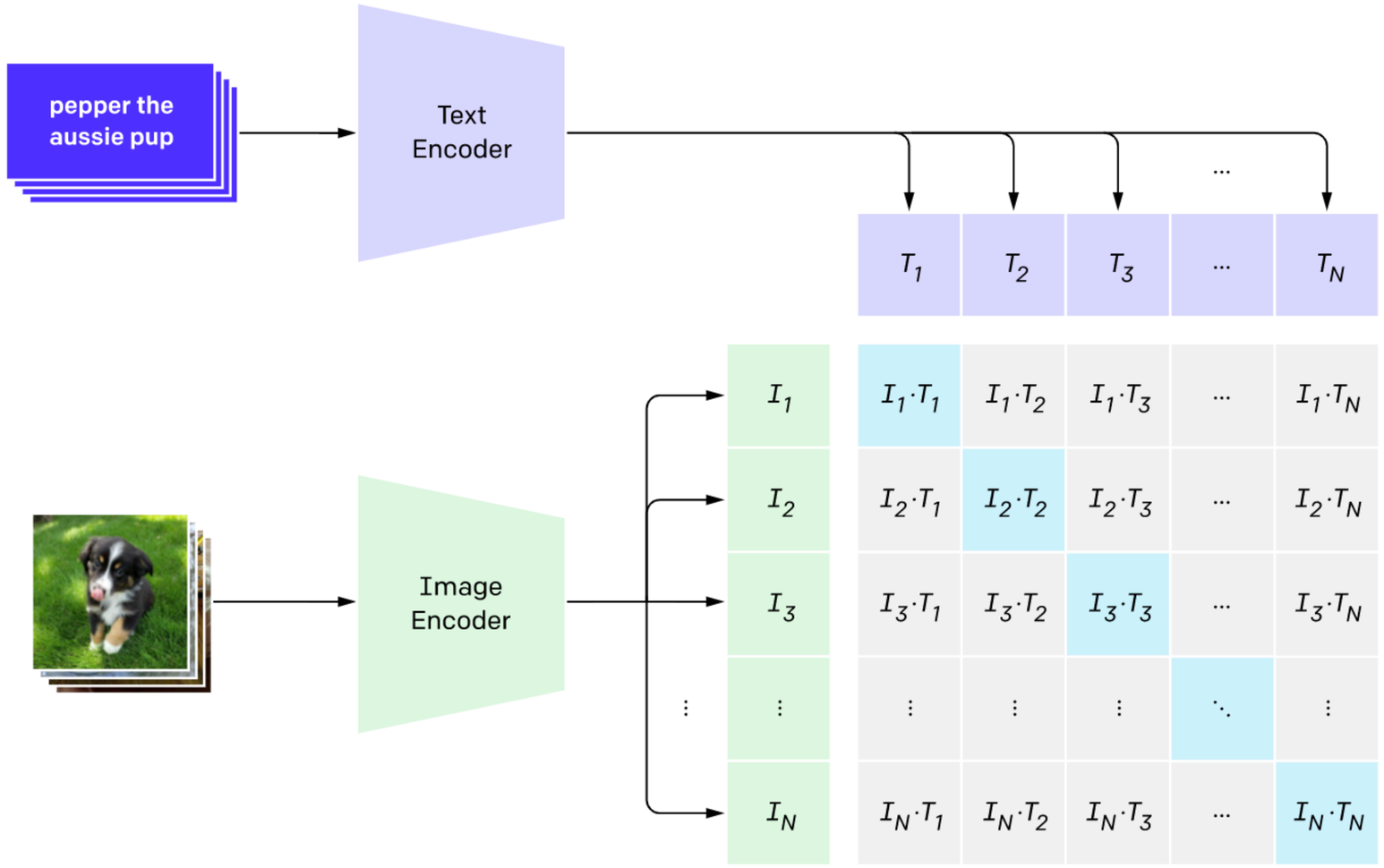
1. Apprentissage représentations latentes pour le texte et les images (d'une batch de taille N)
2. Mesure de similarité
3. Mesure de dissimilarité

DALL.E - CLIP



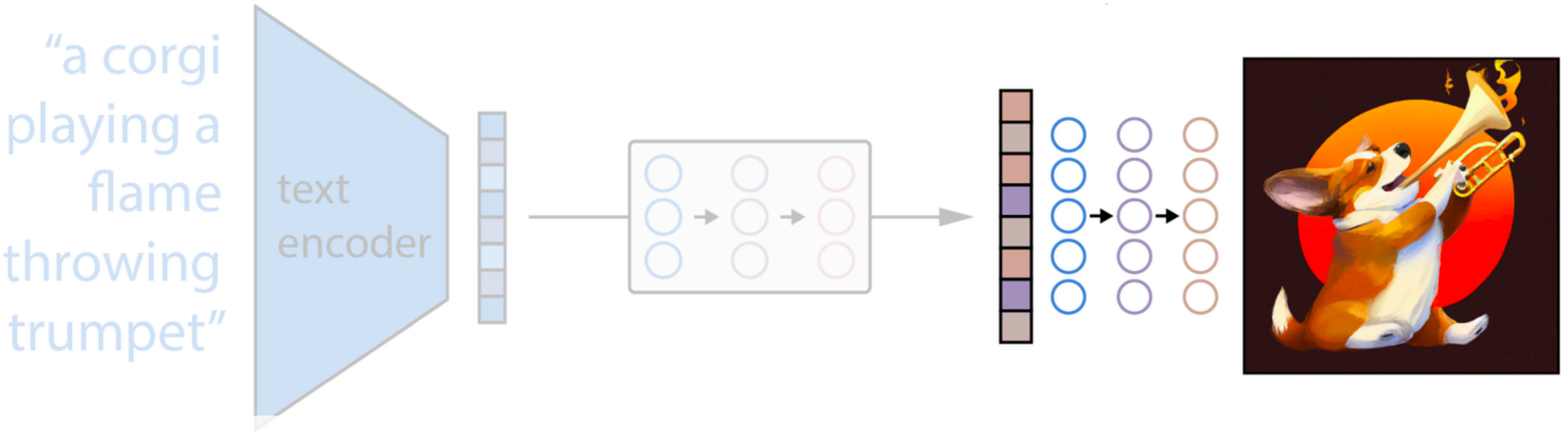
1. Apprentissage représentations latentes pour le texte et les images (d'une batch de taille N)
2. Mesure de similarité
3. Mesure de dissimilarité

DALL.E - CLIP



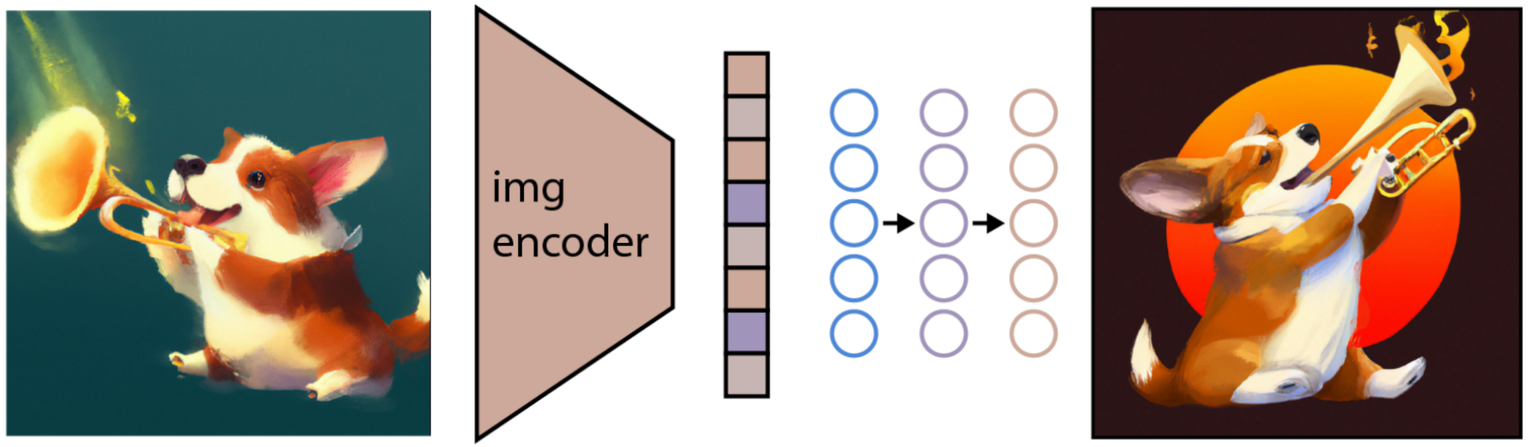
1. Apprentissage représentations latentes pour le texte et les images (d'une batch de taille N)
2. Mesure de similarité
3. Mesure de dissimilarité
4. Maximiser les similarités et minimiser les dissimilarités

DALL.E - GLIDE et modèles de diffusion



1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion
3. Apprentissage de représentations latentes de textes et d’images
4. Wrap-it up!

DALL.E - GLIDE et modèles de diffusion

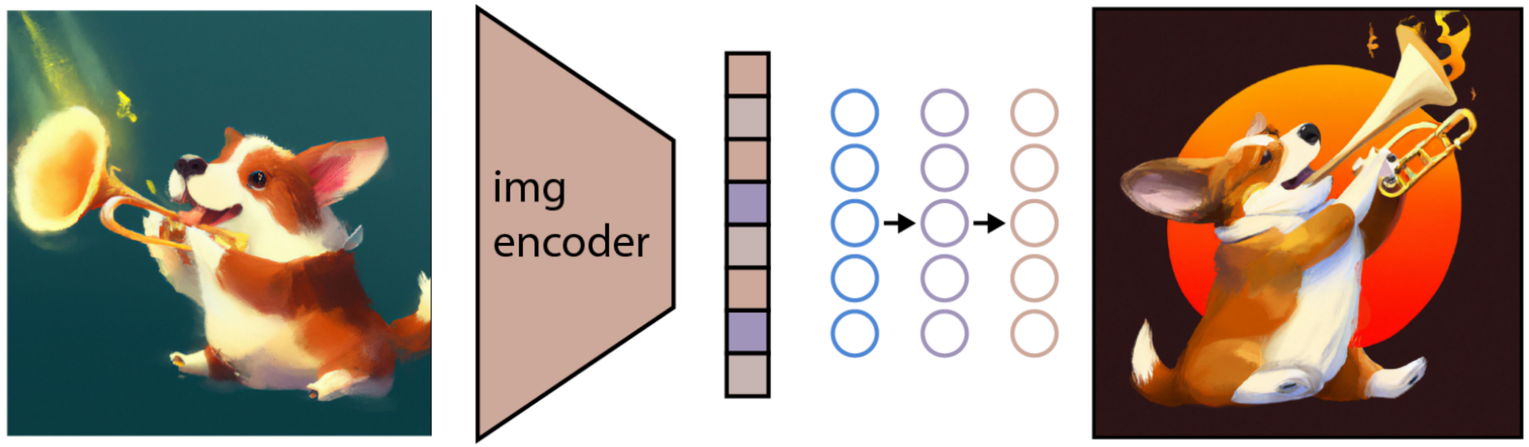


1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion
3. Apprentissage de représentations latentes de textes et d’images
4. Wrap-it up!

DALL.E - GLIDE et modèles de diffusion

Idée:

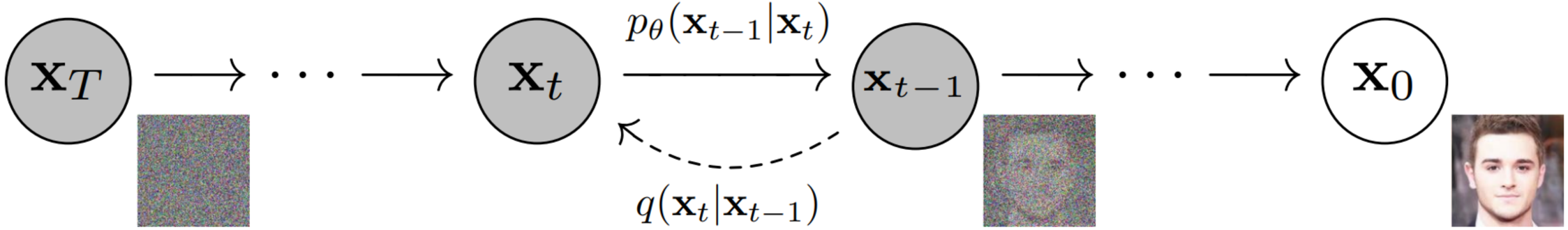
- À partir d'une image, en générer plusieurs lui ressemblant
- Pour ce faire, les modèles de diffusion sont utilisés



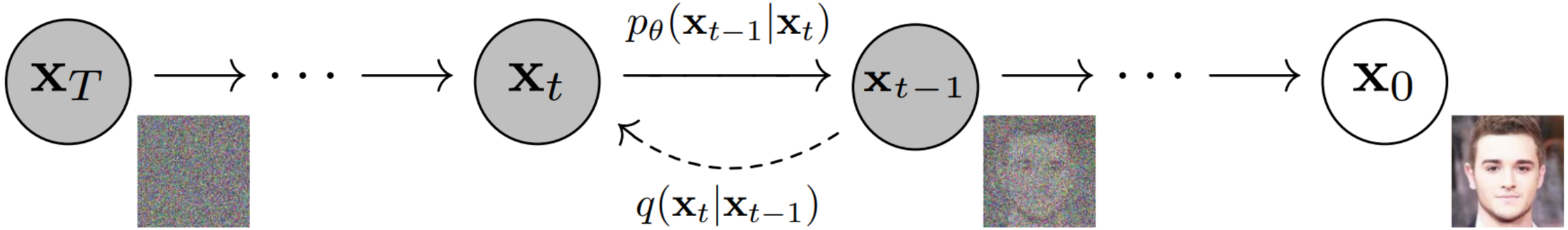
Modèles de diffusion

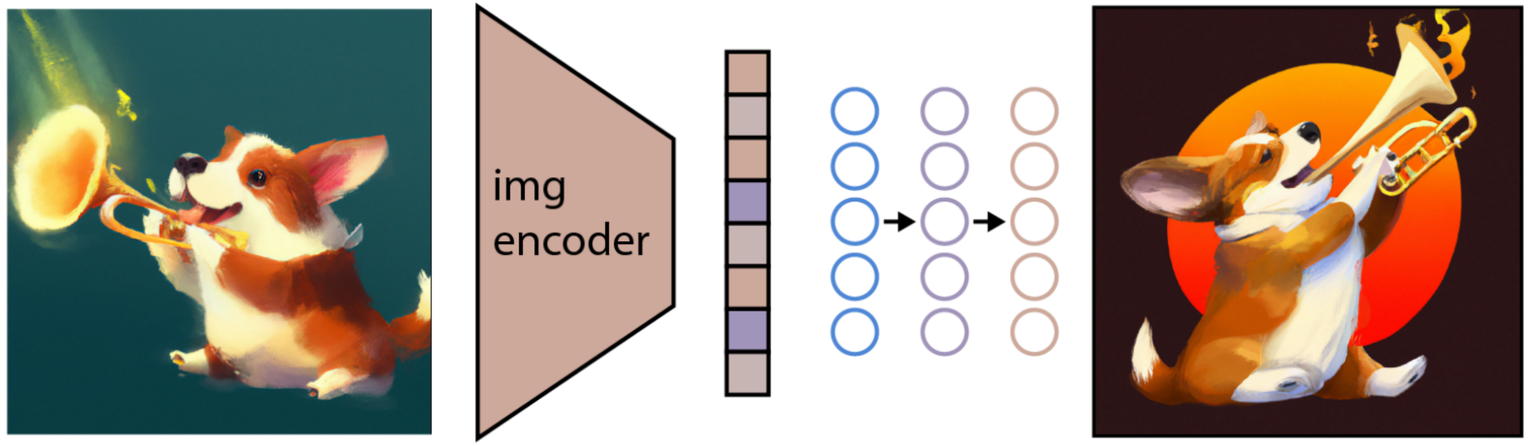
Idée:

- Bruiter de façon incrémentale une image jusqu'à l'obtention d'un bruit blanc
- Débruiter ce bruit pour obtenir l'image originale
- Si on connait le mécanisme de bruitage, on peut à partir d'un bruit blanc, générer des images



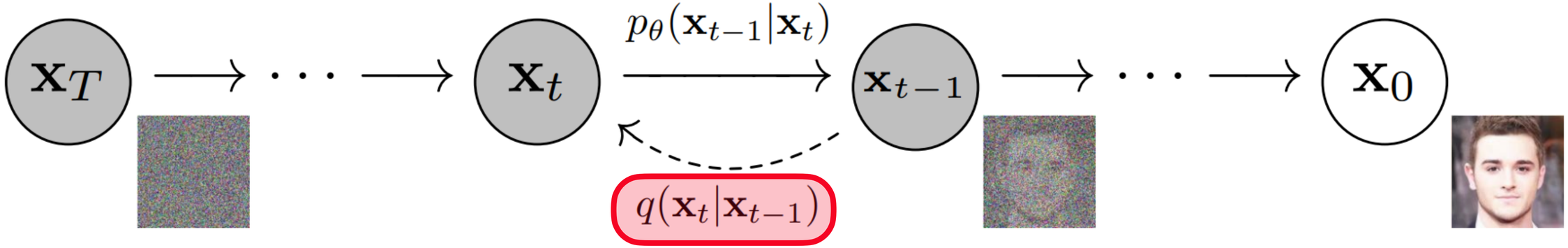
DALL.E - modèles de diffusion

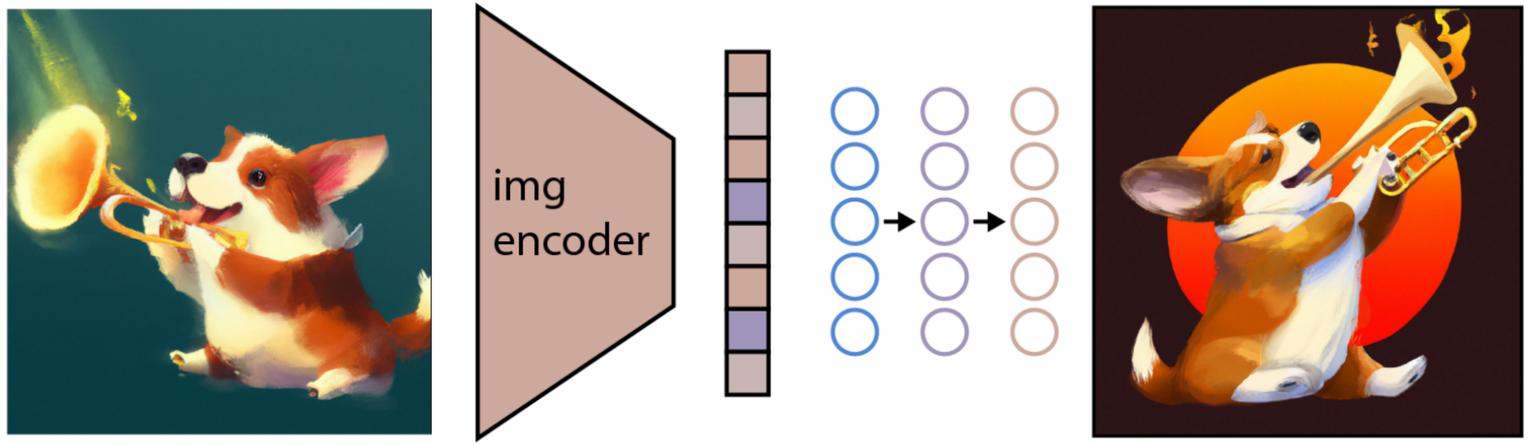




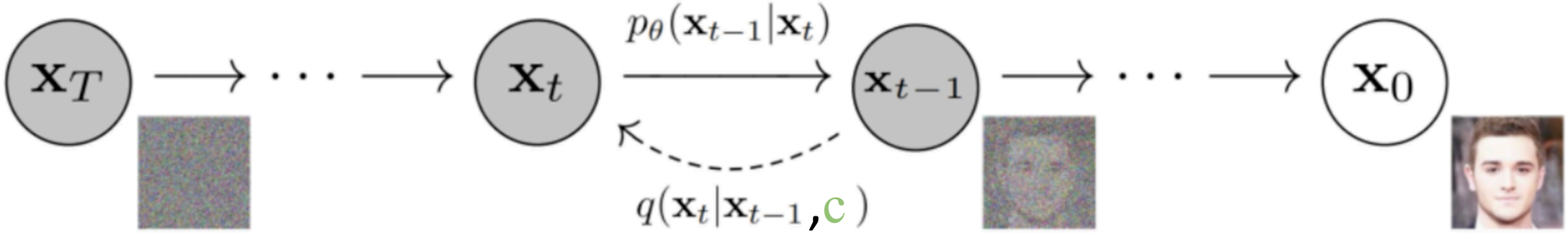
DALL.E - modèles de diffusion

Par contre, dans le processus de diffusion, de l'information supplémentaire sera ajoutée



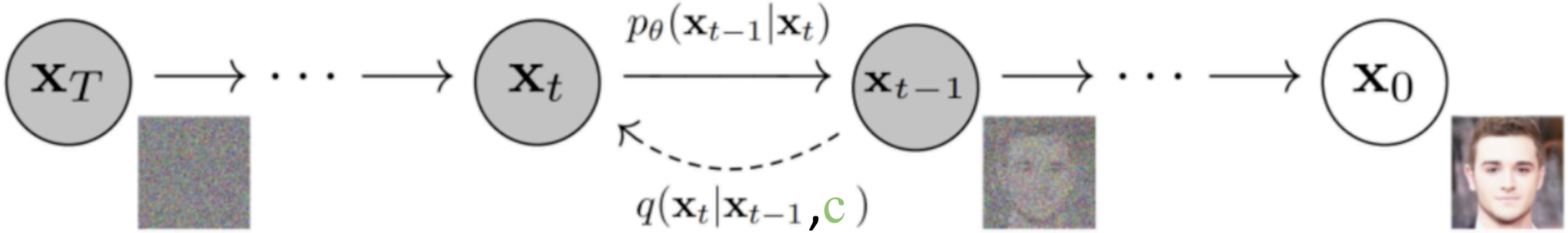


DALL.E - Survol



Utilisation de cette représentation dans le modèle de diffusion

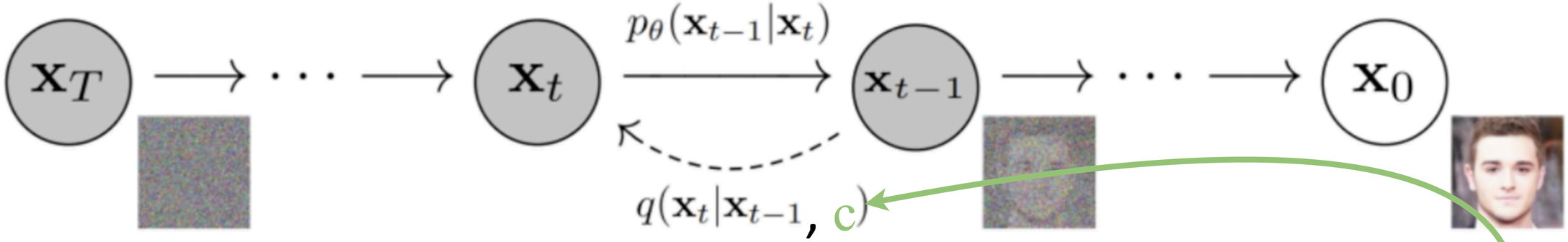
DALL.E - Survol



Utilisation de cette représentation dans le modèle de diffusion

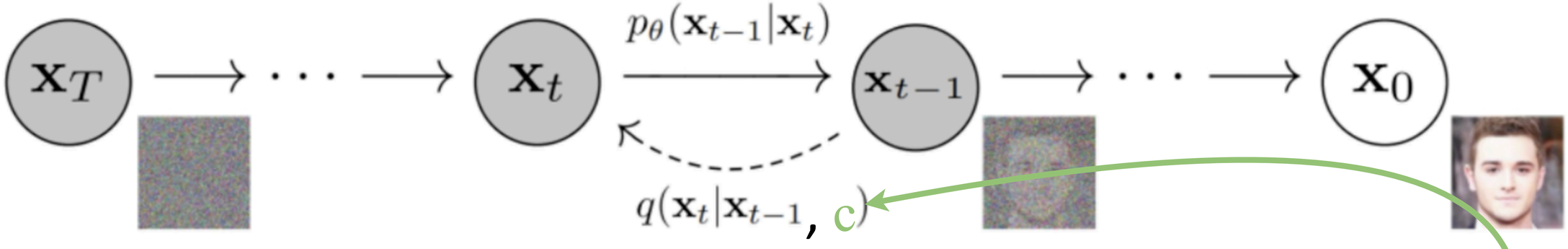
« L'image d'une visage d'un homme un peu roux »

DALL.E - GLIDE



Utilisation de cette représentation dans le modèle de diffusion

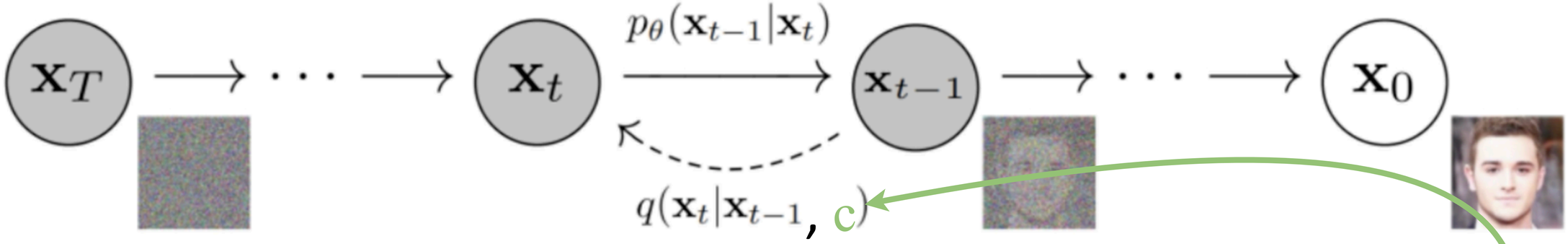
DALL.E - GLIDE



Nous pouvons guider l'apprentissage de la structure latente en fonction de la représentation latente du texte!

Utilisation de cette représentation dans le modèle de diffusion

DALL.E - GLIDE

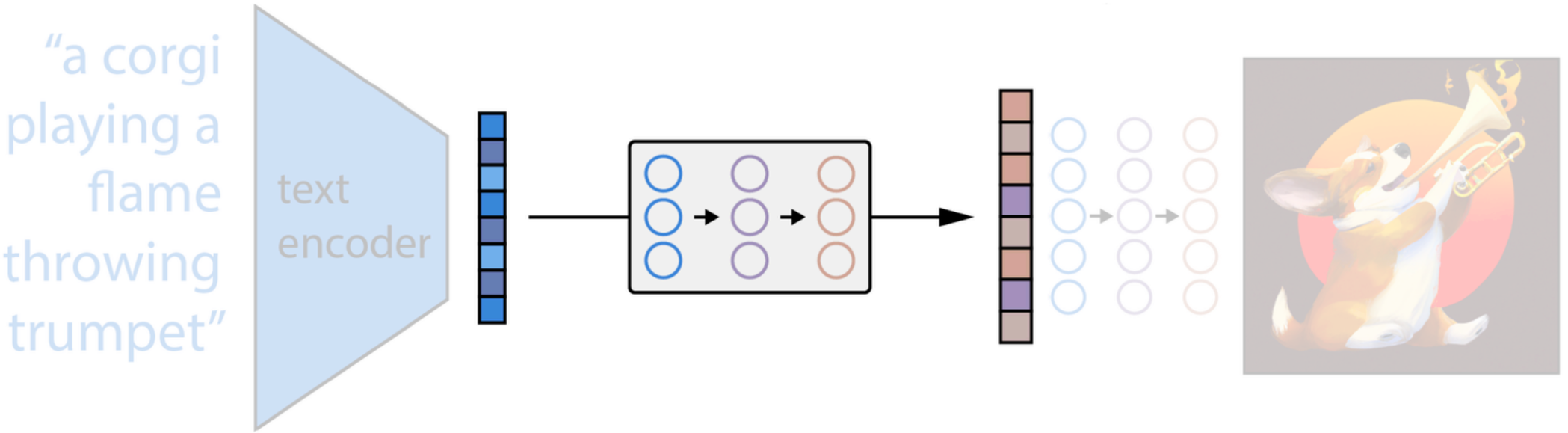


Nous pouvons guider l'apprentissage de la structure latente en fonction de la représentation latente du texte!

Utilisation de cette représentation dans le modèle de diffusion

Mais comment générer une image simplement à partir d'un texte?!?

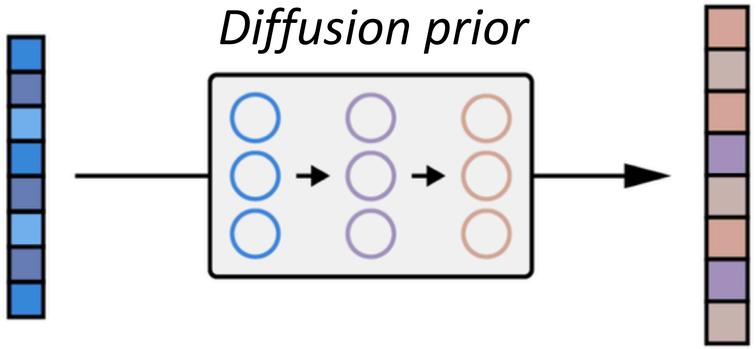
DALL.E - Cartographie sémantique



1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion
3. Apprentissage de représentations latentes de textes et d’images
4. Wrap-it up!

DALL.E - Cartographie sémantique

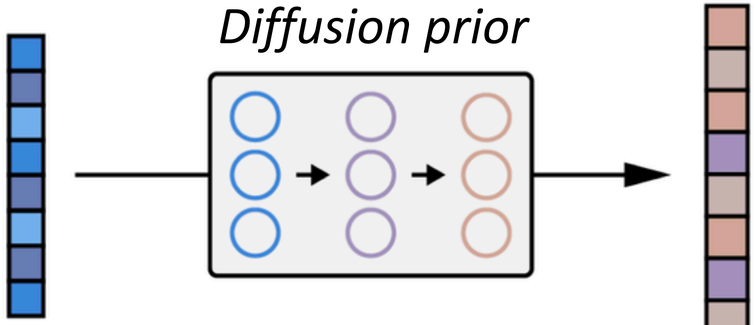
Représentation de
l’image



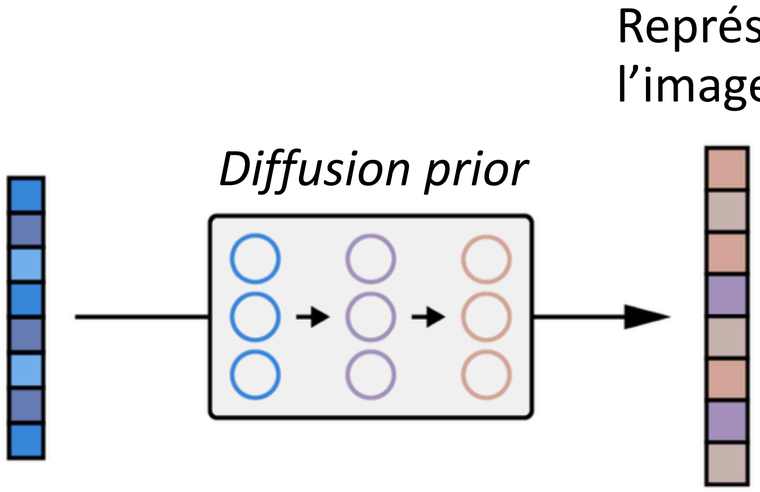
Représentation de
la phrase

DALL.E - Cartographie sémantique

Entraînement (par diffusion)



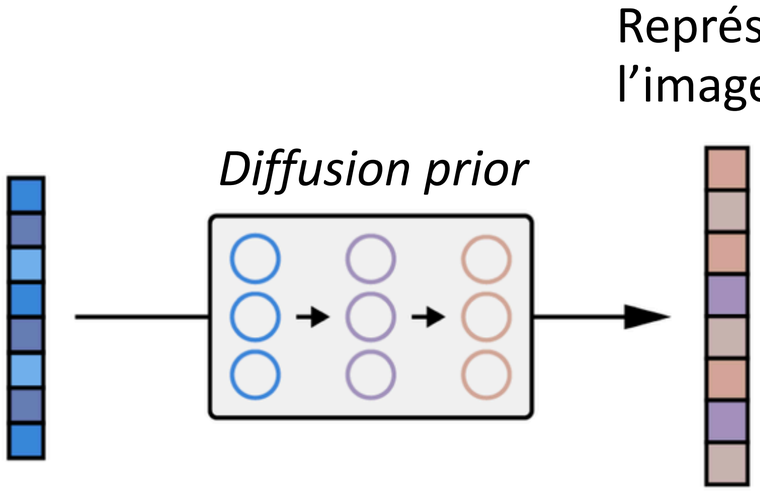
DALL.E - Cartographie sémantique



Entraînement (par diffusion)

1. Plongement du texte à l’aide de CLIP

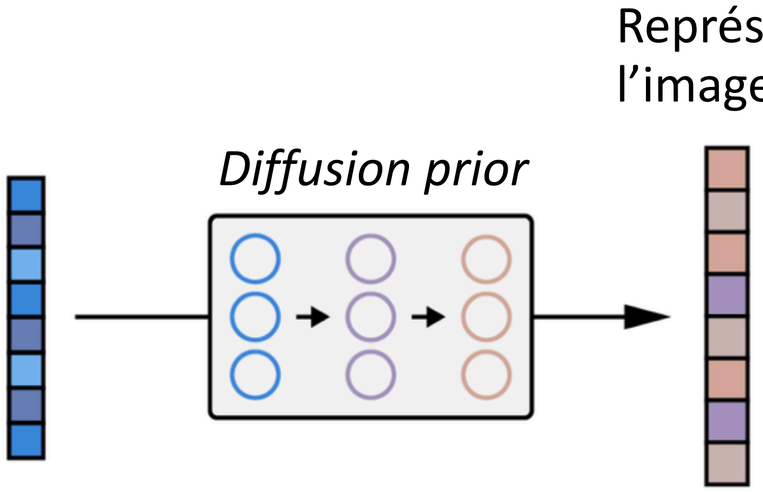
DALL.E - Cartographie sémantique



Entraînement (par diffusion)

1. Plongement du texte à l’aide de CLIP
2. Diffusion vers le plongement CLIPÉ et BRUITÉ de l’image

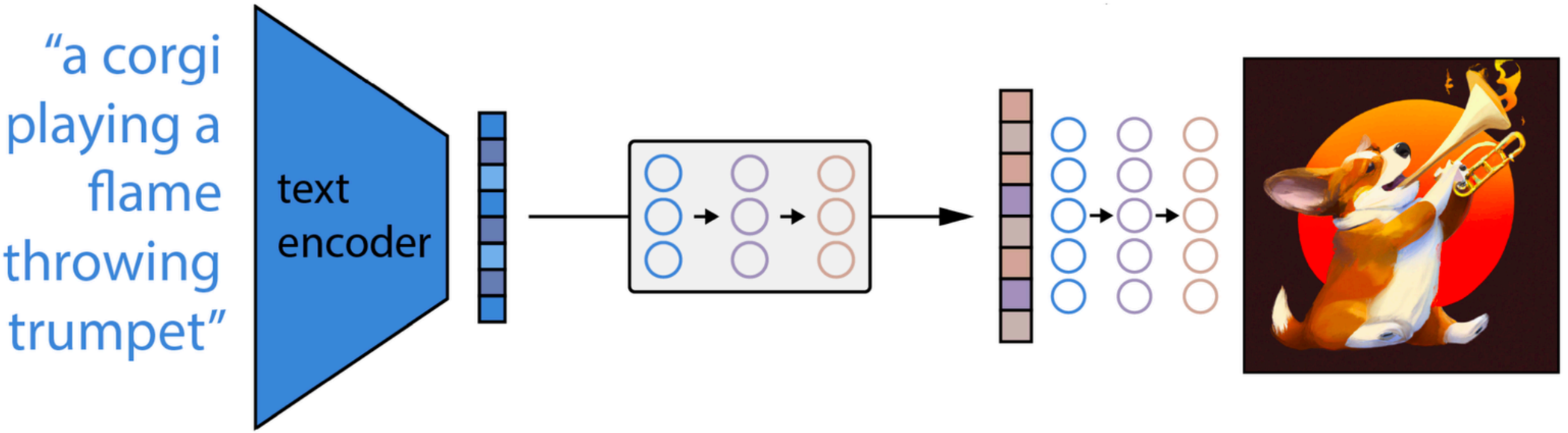
DALL.E - Cartographie sémantique



Entraînement (par diffusion)

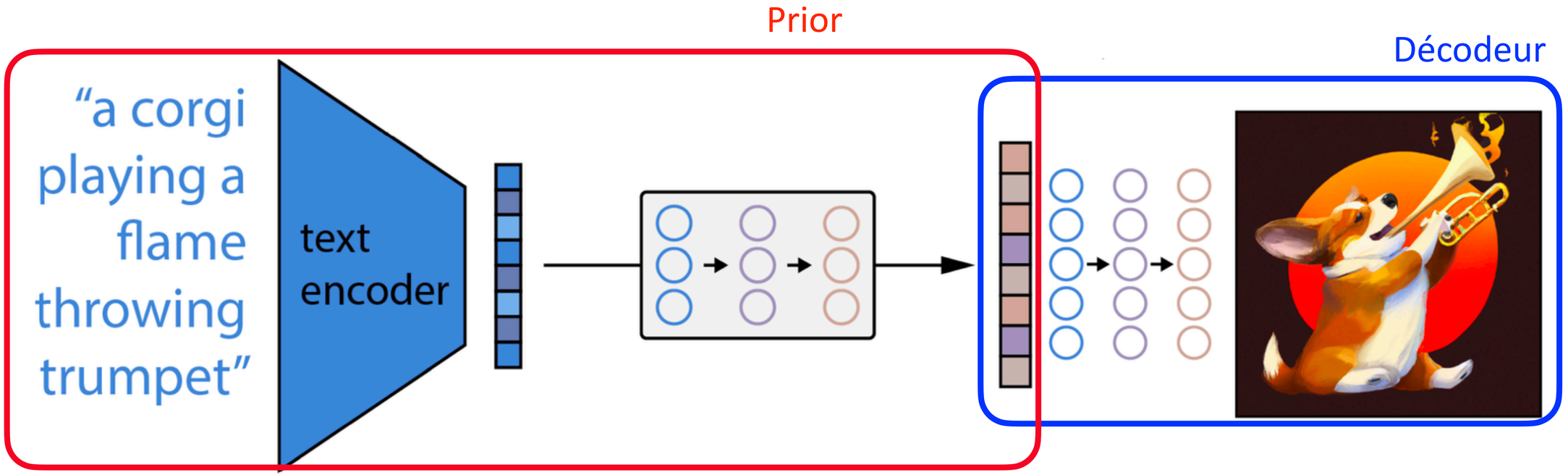
1. Plongement du texte à l’aide de CLIP
2. Diffusion vers le plongement CLIPÉ et BRUITÉ de l’image
3. Prédiction par transformer du plongement CLIPÉ de l’image

DALL.E - Wrap-it up!



1. Contrastive Language-Image Pre-training (CLIP)
2. Génération d’images à partir d’une image à l’aide des modèles de diffusion
3. Apprentissage de représentations latentes de textes et d’images
4. Wrap-it up!

DALL.E - Wrap-it up!



Idée:

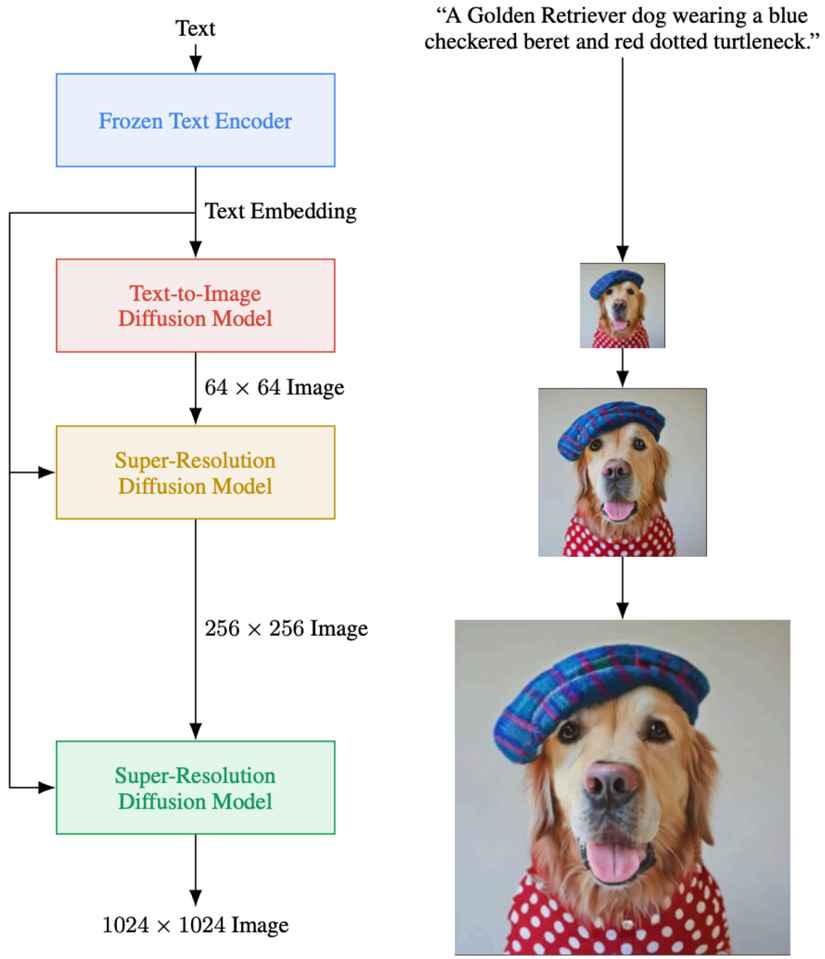
- Soit (*x, y*) un couple composé d’une image *x* et de texte *y*.
- Soit z la représentation latente de l’image.
- On peut exprimer la distribution de l’image en fonction du texte par:

$$P(x|y) = P(x, z_i|y) = P(x|z_i, y)P(z_i|y)$$

Imagen

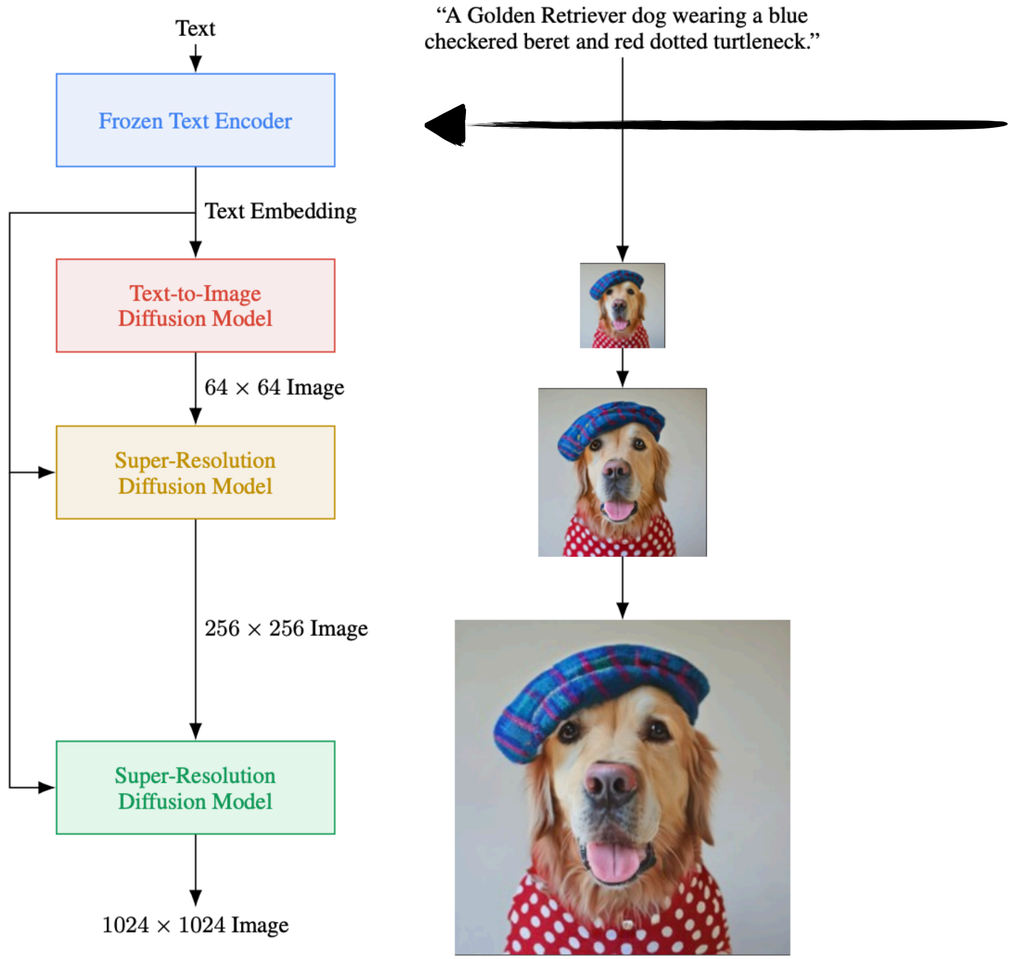
https://imagen.research.google/

Architecture



https://arxiv.org/pdf/2205.11487

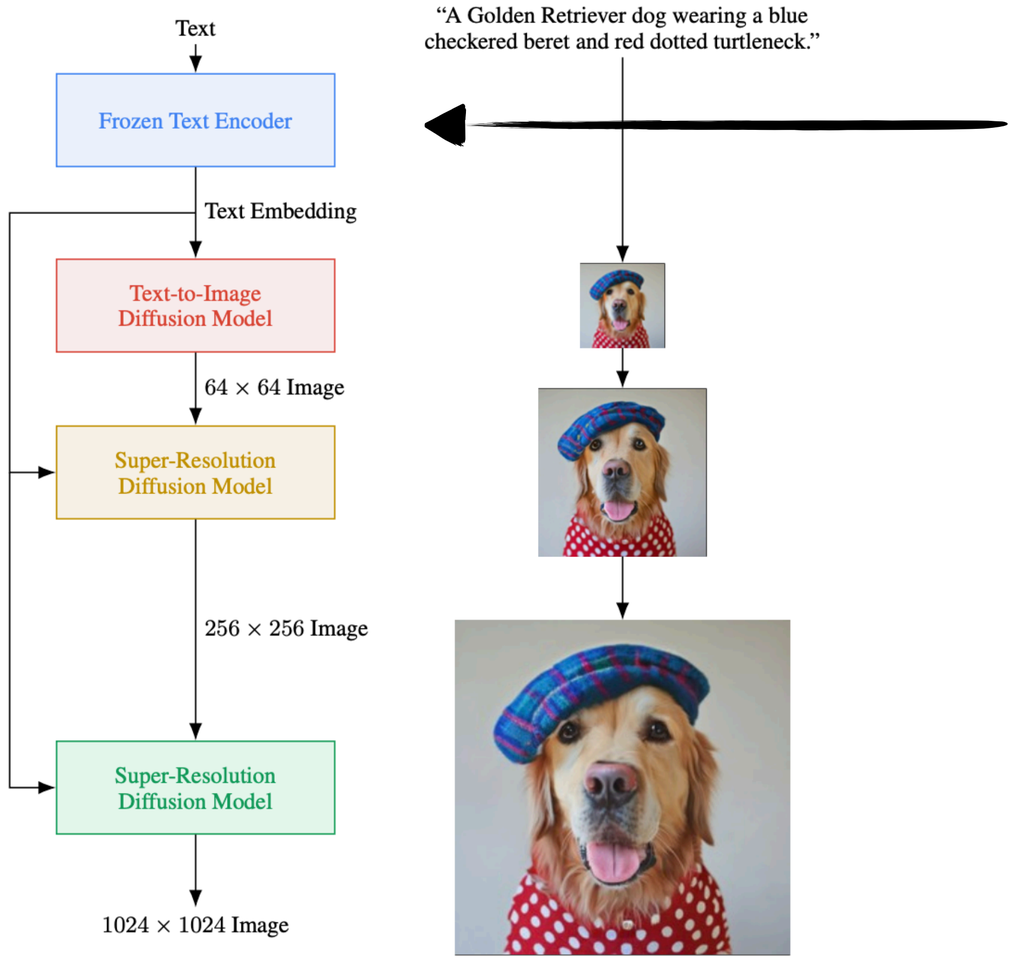
Architecture



- Utilise un encodeur *préentraîné et fixé*

https://arxiv.org/pdf/2205.11487

Architecture

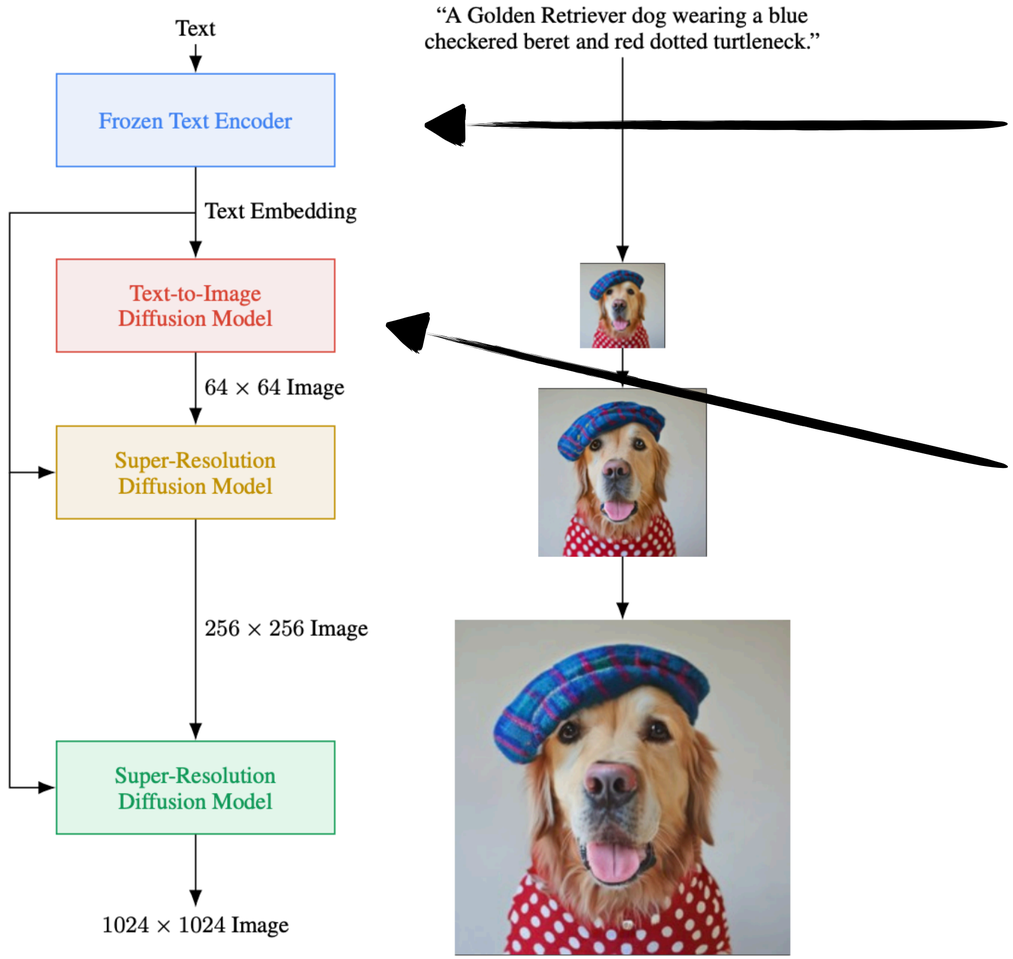


- Utilise un encodeur *préentraîné et fixé*

P. ex., un transformeur de type encodeur entraîné à partir d'un grand jeu de données de texte à prédire le prochain mot d'une phrase. **Aucune image!**

https://arxiv.org/pdf/2205.11487

Architecture



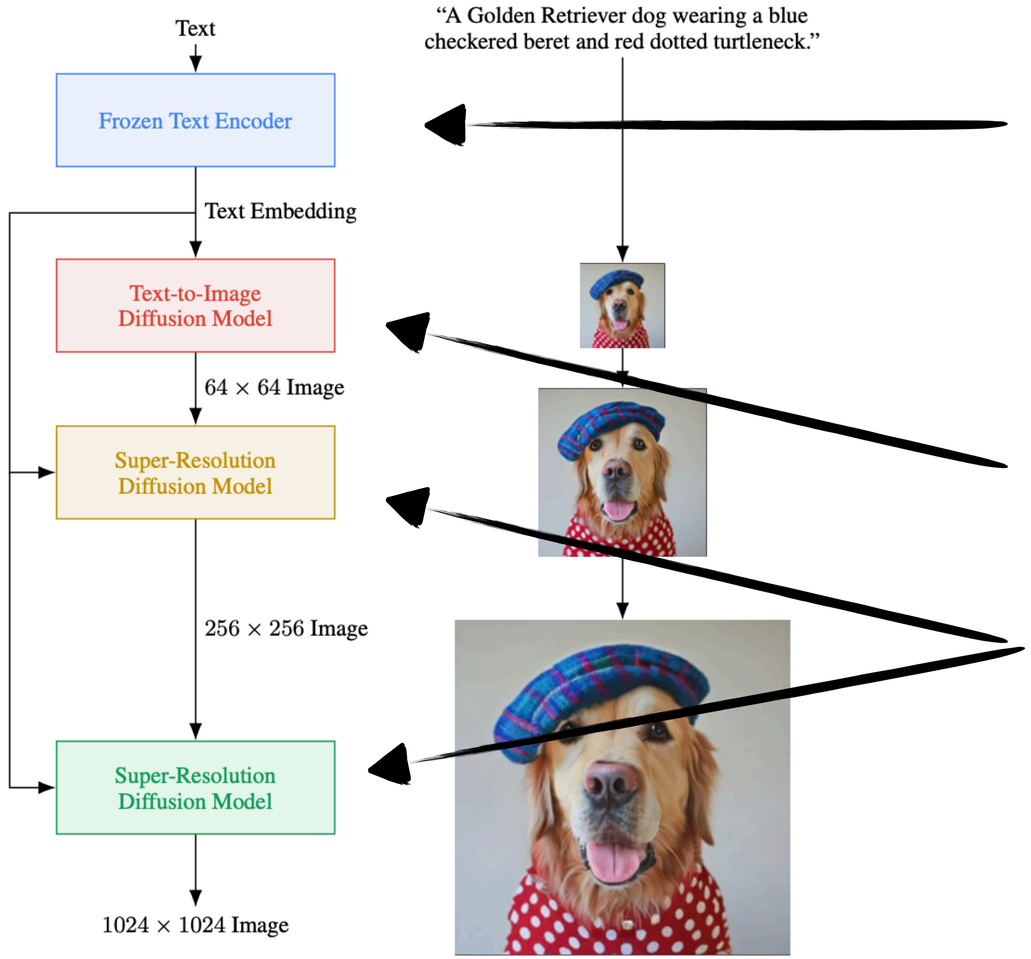
- Utilise un encodeur *préentraîné et fixé*

P. ex., un transformeur de type encodeur entraîné à partir d'un grand jeu de données de texte à prédire le prochain mot d'une phrase. **Aucune image!**

- *Suivi d'un modèle de diffusion pour obtenir une première image*

https://arxiv.org/pdf/2205.11487

Architecture



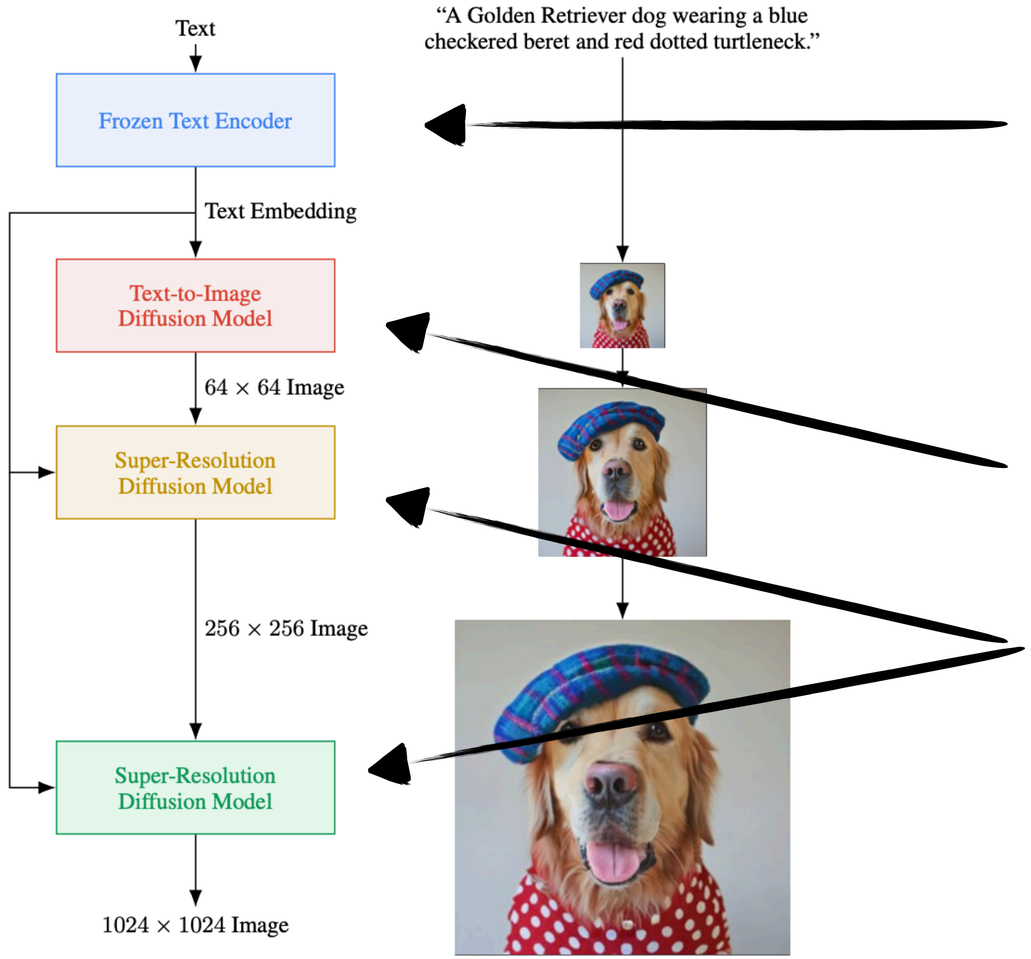
- Utilise un encodeur *préentraîné et fixé*

P. ex., un transformeur de type encodeur entraîné à partir d'un grand jeu de données de texte à prédire le prochain mot d'une phrase. **Aucune image!**

- *Suivi d'un modèle de diffusion pour obtenir une première image*
- Puis de quelques autres modèles de diffusion pour obtenir des images en plus haute résolution

https://arxiv.org/pdf/2205.11487

Architecture



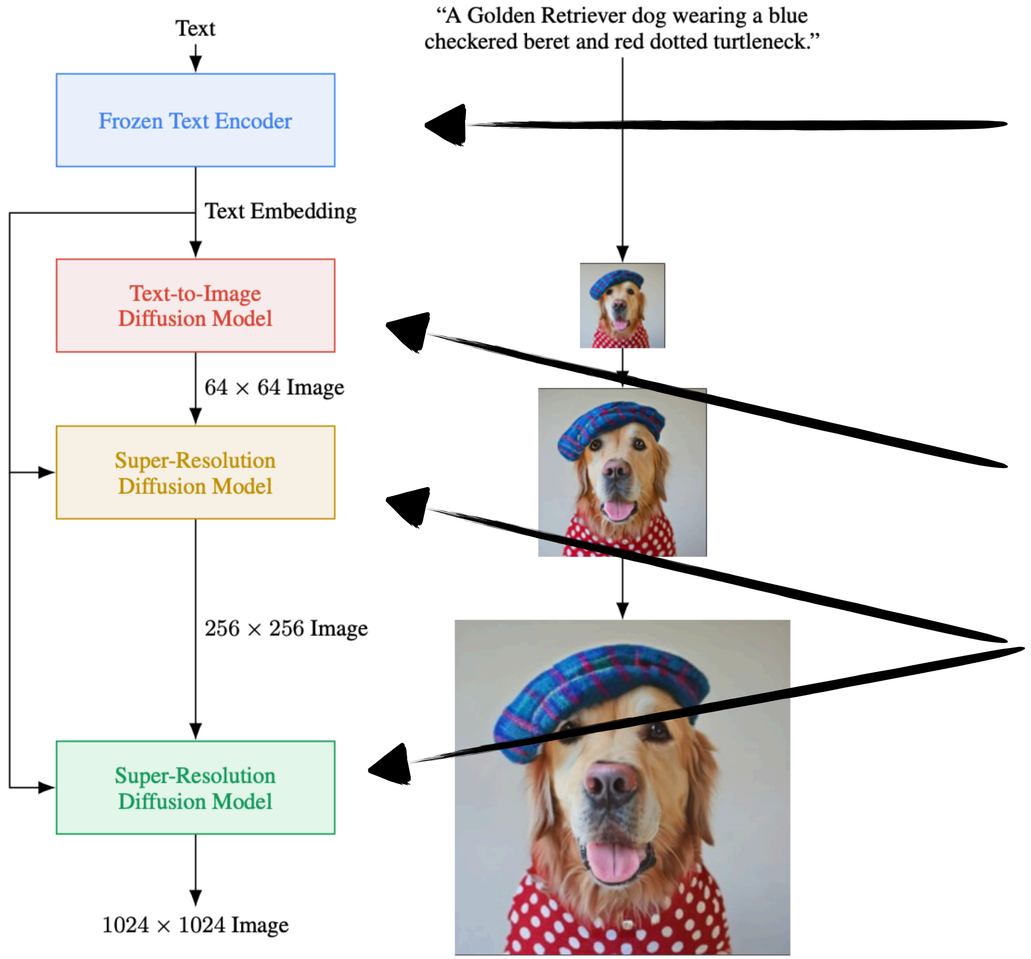
- Utilise un encodeur *préentraîné et fixé*

P. ex., un transformeur de type encodeur entraîné à partir d'un grand jeu de données de texte à prédire le prochain mot d'une phrase. **Aucune image!**

- *Suivi d'un modèle de diffusion pour obtenir une première image*
- Puis de quelques autres modèles de diffusion pour obtenir des images en plus haute résolution
  - Les modèles utilisent un mécanisme d'attention sur la représentation du texte

https://arxiv.org/pdf/2205.11487

Architecture



- Utilise un encodeur *préentraîné et fixé*

P. ex., un transformeur de type encodeur entraîné à partir d'un grand jeu de données de texte à prédire le prochain mot d'une phrase. **Aucune image!**

- *Suivi d'un modèle de diffusion pour obtenir une première image*
- Puis de quelques autres modèles de diffusion pour obtenir des images en plus haute résolution
  - Les modèles utilisent un mécanisme d'attention sur la représentation du texte
- Les modèles de diffusion utilisent de U-Net

https://arxiv.org/pdf/2205.11487

« Classifier-free » guidance

- Le modèle de diffusion est entraîné avec deux objectifs:

  1. Générer des images à partir de texte

  2. (Mais aussi) Générer des images

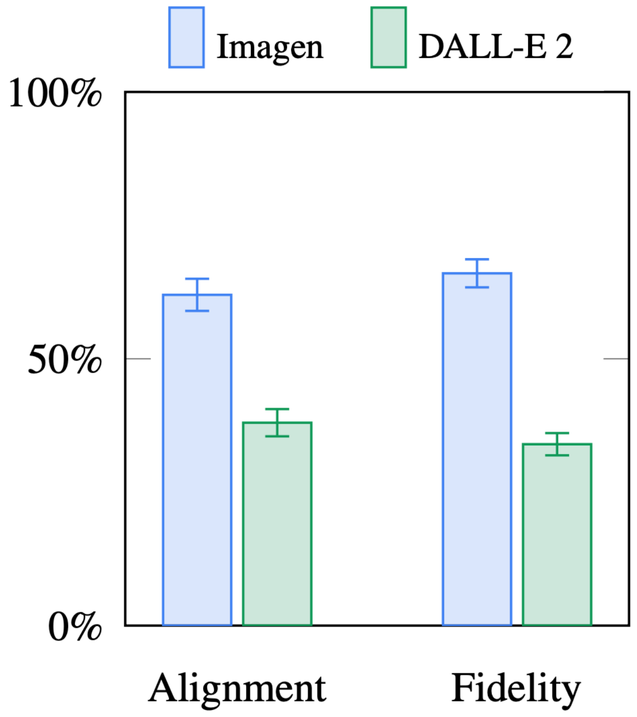
     - Obtenir des images de grande qualité (1) et diversifiées (2)

- Imagen propose une façon pour éviter que certains pixels « saturent » pendant la diffusion (analogie au clipping dans les RNN)

Avant Imagen

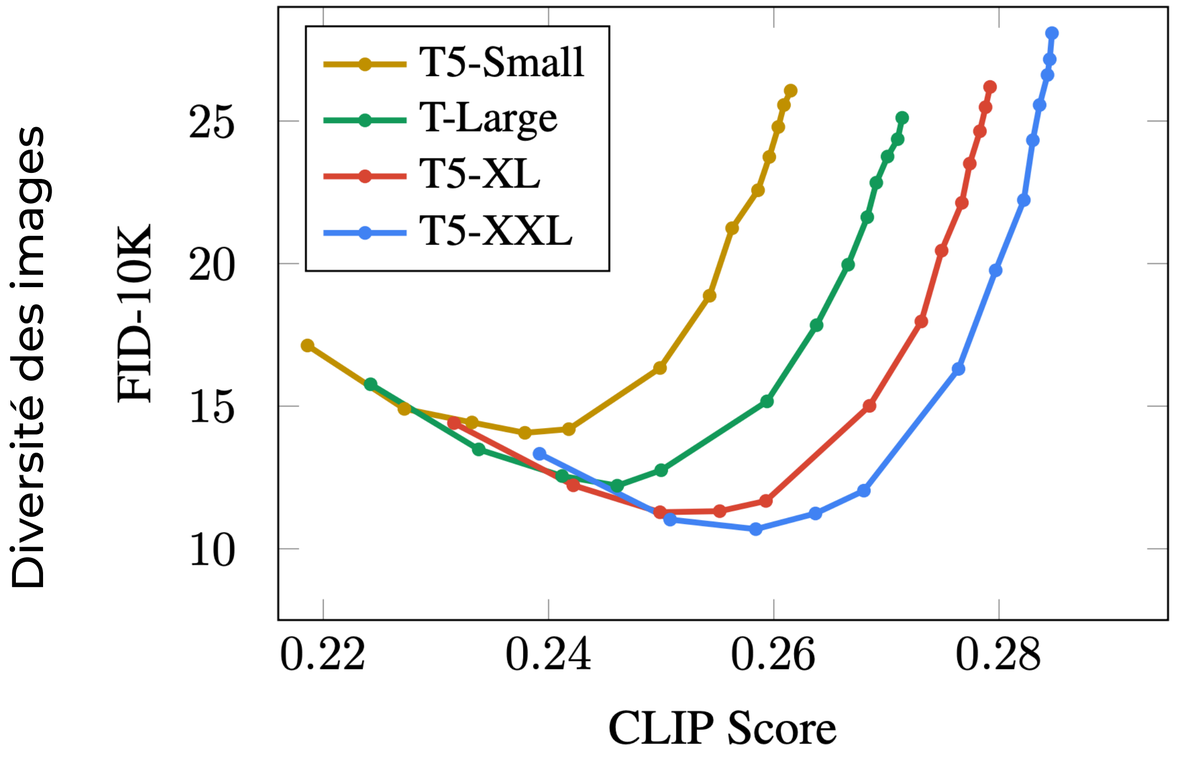
Imagen

Par rapport à Dall-E 2



- Meilleure performance empirique (selon une comparaison par des humains)
  - “Alignement” -> “La description décrit-elle précisément l’image?”
  - “Fidélité” -> “Quelle image est la plus photoréaliste (a l’air plus réelle)?”
- Architecture plus simple (sans CLIP)

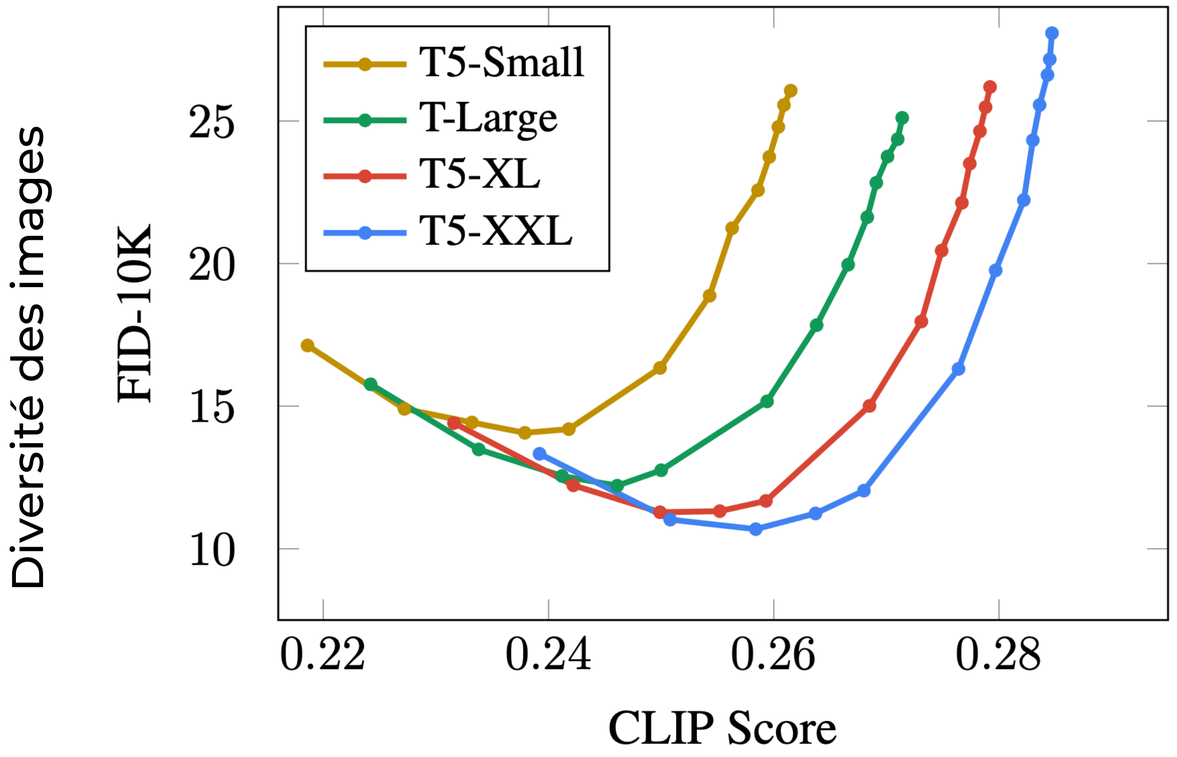
La taille de l’encodeur est un hyperparamètre important



T5 XXL (2.6B)

- Trans. Encodeur-Decodeur
  - Utilise que l’encodeur

La taille de l’encodeur est un hyperparamètre important

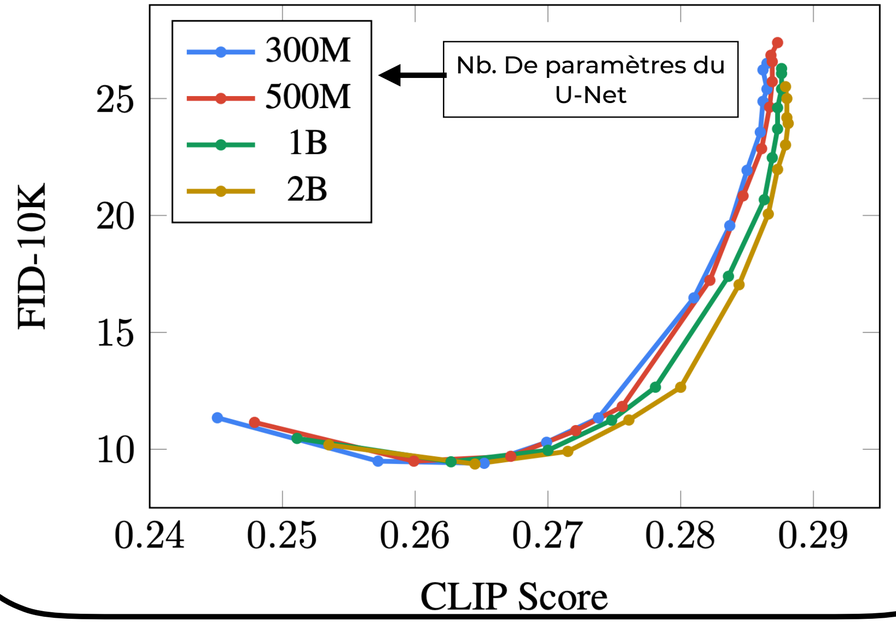


Alignement entre les images et le texte

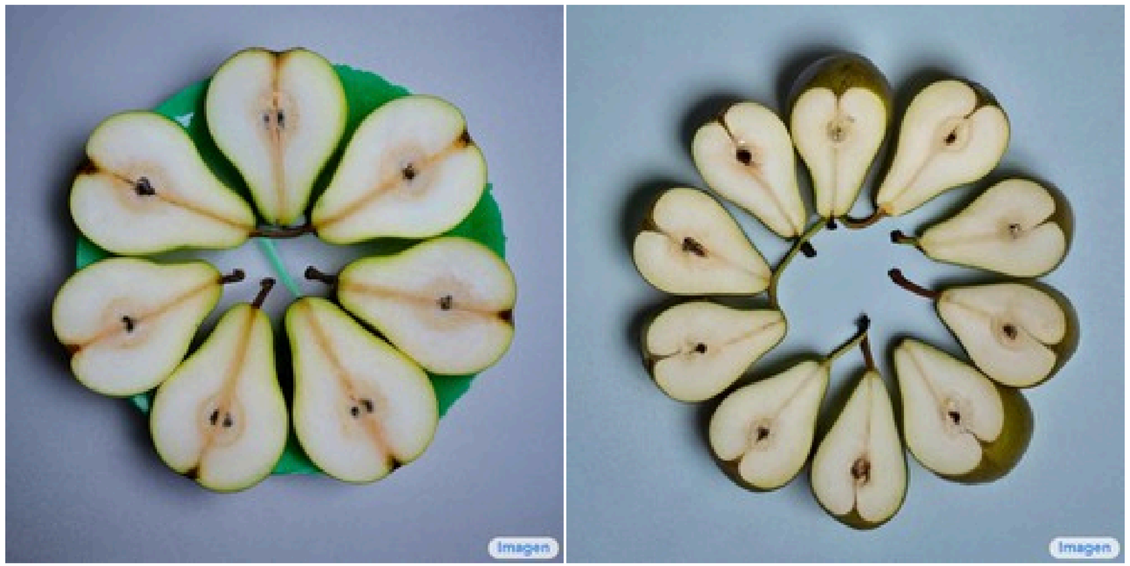
T5 XXL (2.6B)

- Trans. Encodeur-Decodeur
  - Utilise que l’encodeur

En comparaison la taille du modèle de génération d’images est moins importante



Encore loin d’être parfait …



A pear cut into seven pieces arranged in a ring.

One cat and two dogs sitting on the grass.

- **Opérations demandant de compter (ou d’écrire) restent difficiles**