Normalizing Flows pour surmonter le problème de pré-image en ML

Clément Glédel, Benoît Gaüzère, Paul Honeine

LITIS Lab, INSA et Université de Rouen, Rouen, France

Sommaire

Apprentissage machine

Apprentissage machine :

- Modèle de prédiction : $\mathcal{X} \rightarrow Y$ avec Y l'ensemble des valeurs de prédiction.
- Certains modèles prédictifs génèrent un espace latent $\mathcal{Z}$ dans lequel la prédiction est effectuée.

Le problème de la pré-image :

- Définition : Générer des données dans $\mathcal{X}$ à partir de points dans $\mathcal{Z}$.
- Motivations : Explicabilité (comprendre l'espace latent $\mathcal{Z}$), générer un *graphe moyen*, ...

Problème de pré-image

- Soit $\mathcal{Z}$ l'espace engendré par la fonction Φ.
- Trouver $x^* \in \mathcal{X}$ qui, après projection dans $\mathcal{Z}$, soit le plus proche possible de φ^* :

$$x^* = \underset{x \in \mathcal{X}}{\operatorname{argmin}} \|\Phi(x) - \varphi^*\|_{\mathcal{Z}}^2$$

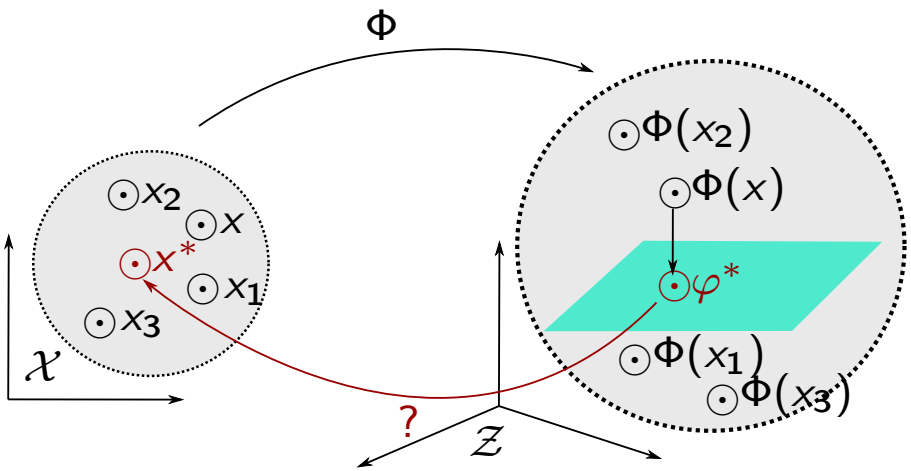


Figure – Illustration du problème de pré-image, qui consiste à trouver $x^* \in \mathcal{X}$ dont l'image dans $\mathcal{Z}$ est la plus proche possible de φ^*.

Solutions

Le problème de pré-image est un problème difficile et mal posé

Solutions :

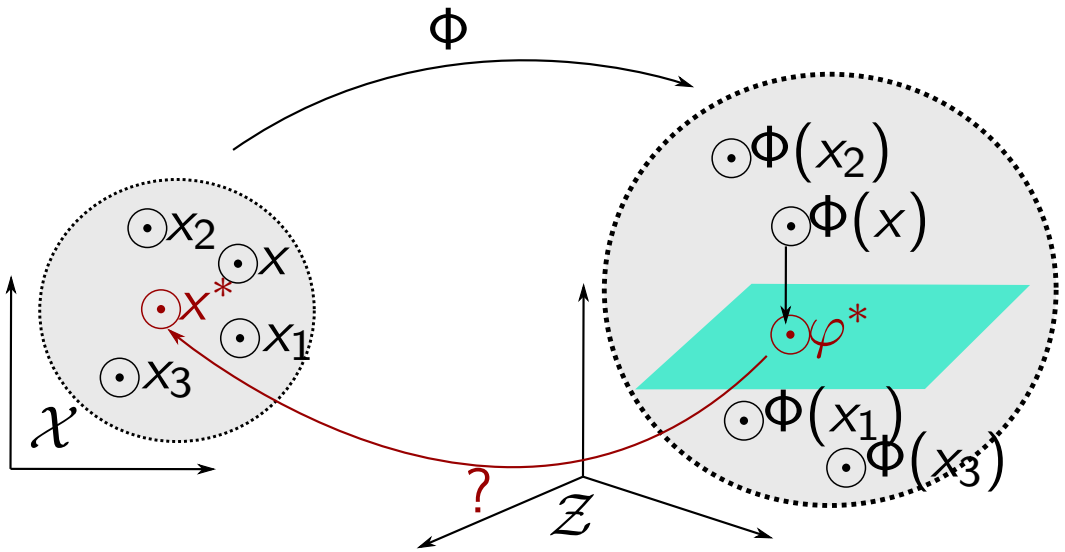
- Bakır, G. H., Weston, J., & Schölkopf, B. (2004). Learning to find pre-images. Advances in neural information processing systems
- Honeine, P., & Richard, C. (2009). Solving the pre-image problem in kernel machines : A direct method. In 2009 IEEE International Workshop on Machine Learning for Signal Processing
- Honeine, P., & Richard, C. (2011). Preimage problem in kernel-based machine learning. IEEE Signal Processing Magazine

Données structurées :

- Bakır, G. H., Zien, A., & Tsuda, K. (2004). Learning to find graph pre-images. In Proc. of 26th DAGM Symposium on Pattern Recognition
- Jia, L., Gaüzère, B., & Honeine, P. (2021). A graph pre-image method based on graph edit distances. In Proc. Joint IAPR International Workshops S+ SSPR (Structural, Syntactic, and Statistical Pattern Recognition)

Modèles génératifs

Modèles génératifs : à partir d'échantillons de l'espace latent $\mathcal{Z}$, permet la génération de données dans $\mathcal{X}$.



Notre idée : Apporter une solution au problème de pré-image avec l'utilisation de modèles génératifs définissant la transformation inverse Φ^{-1}.

Modèles génératifs

- Variational AutoEncoder (VAE)
  - Modèle composé de deux réseaux : un encodeur et un décodeur.
- Generative Adversarial Network (GAN)
  - Modèle constitué de deux réseaux en compétition : un générateur qui génère des données et un discriminateur qui évalue le realisme des données.
- **Normalizing Flows (NF)**
  - **Modèle définissant une transformation Φ inversible entre deux espaces.**

Normalizing Flows (NF)

- A partir d'une distribution complexe $\mathcal{X}$, permet la génération d'une simple distribution $\mathcal{Z}$
- Réversibilité naturelle

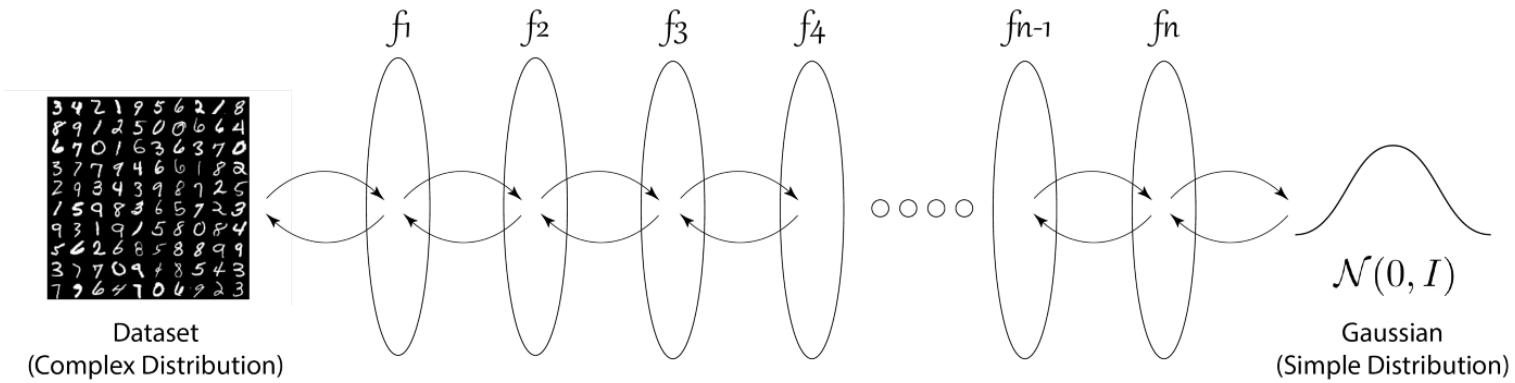
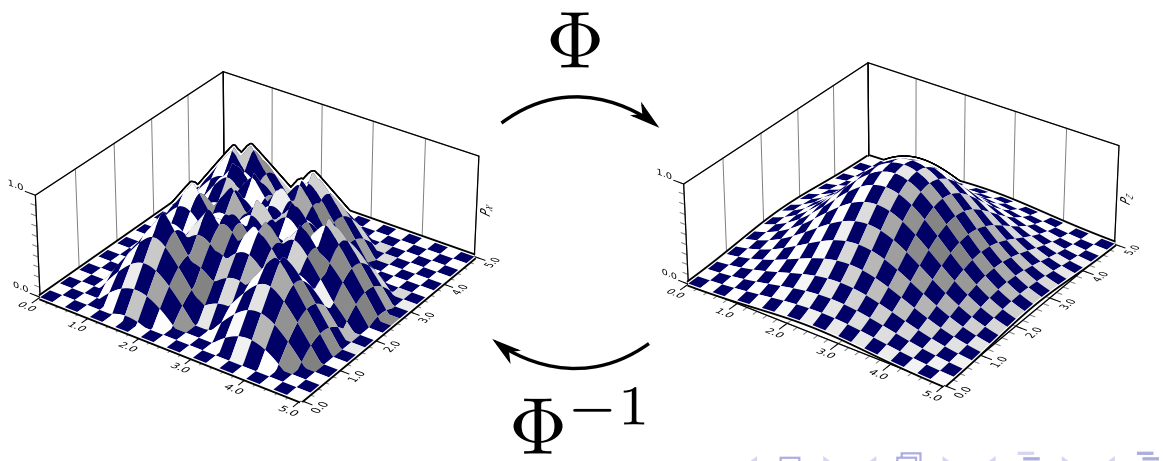


Figure – Normalizing Flows, *source : https://towardsdatascience.com/introduction-to-normalizing-flows*

Formule de changement de variable

$$P_{\mathcal{X}}(x) = P_{\mathcal{Z}}(z) \left| \det \left(\frac{\partial z}{\partial x} \right) \right|$$

$$\log P_{\mathcal{X}}(X) = \sum_{i=1}^{N} \log P_{\mathcal{Z}}(\Phi(\mathrm{x}_i)) + \log \left| \det \left(\frac{\partial \Phi(\mathrm{x}_i)}{\partial \mathrm{x}_i} \right) \right|$$



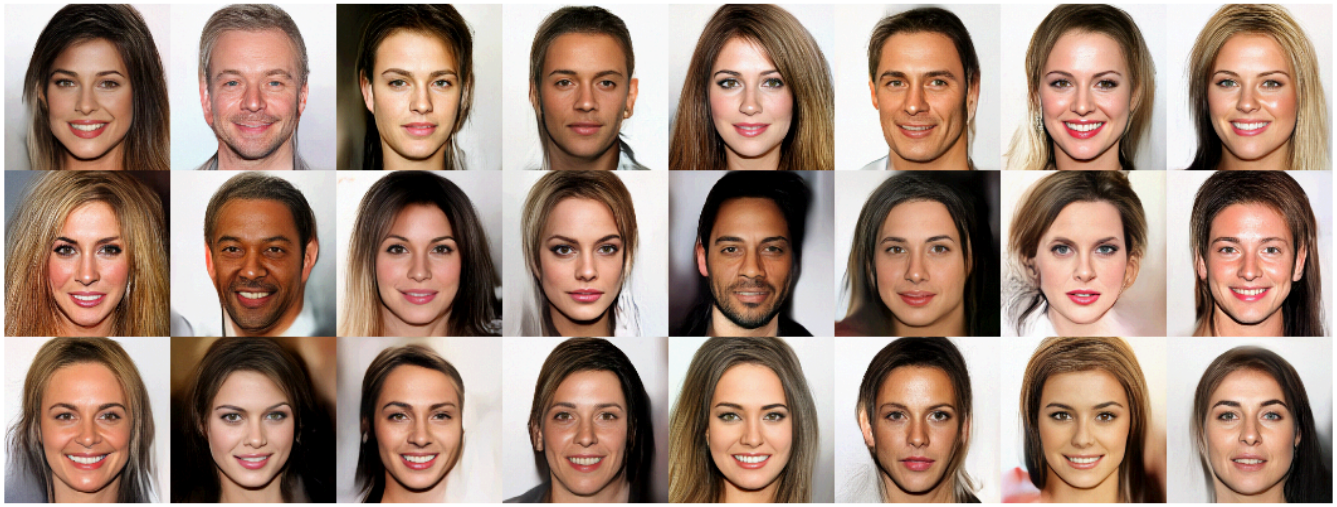
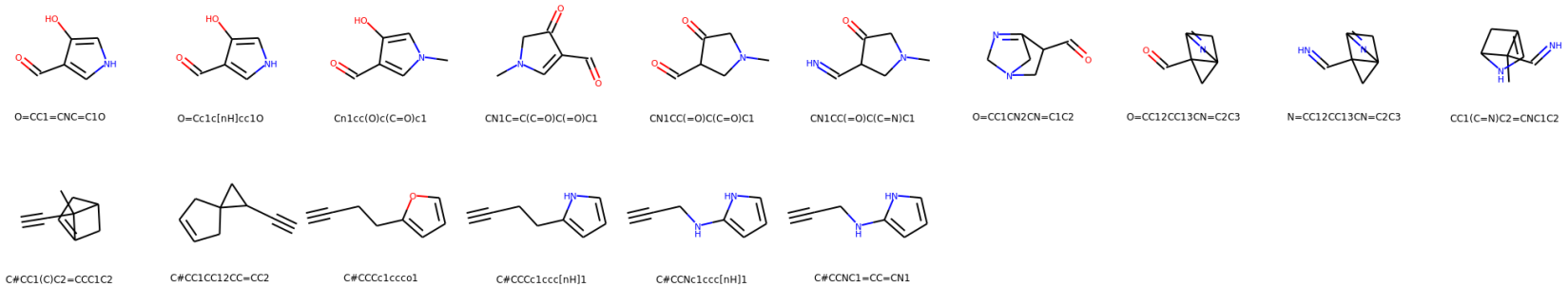


Figure – Images générées avec **Glow**. *source : Kingma, D. P., & Dhariwal, P. (2018). Glow : Generative flow with invertible 1x1 convolutions.*



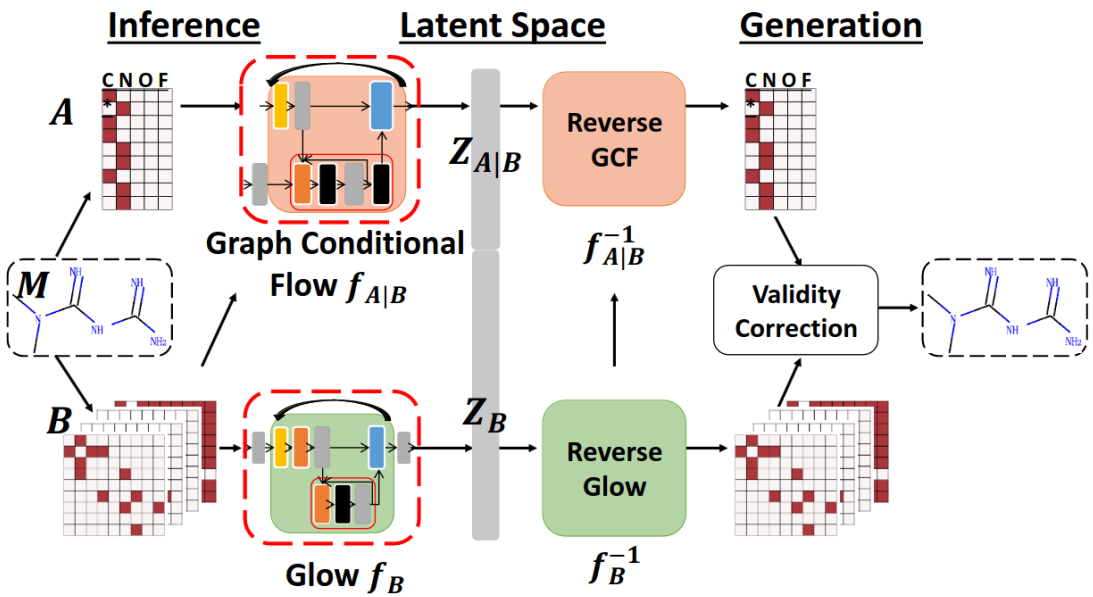


Figure – Architecture de **MoFlow**. *source : Zang, C., Wang, F. : Moflow : an invertible flow model for generating molecular graphs.*

Sommaire

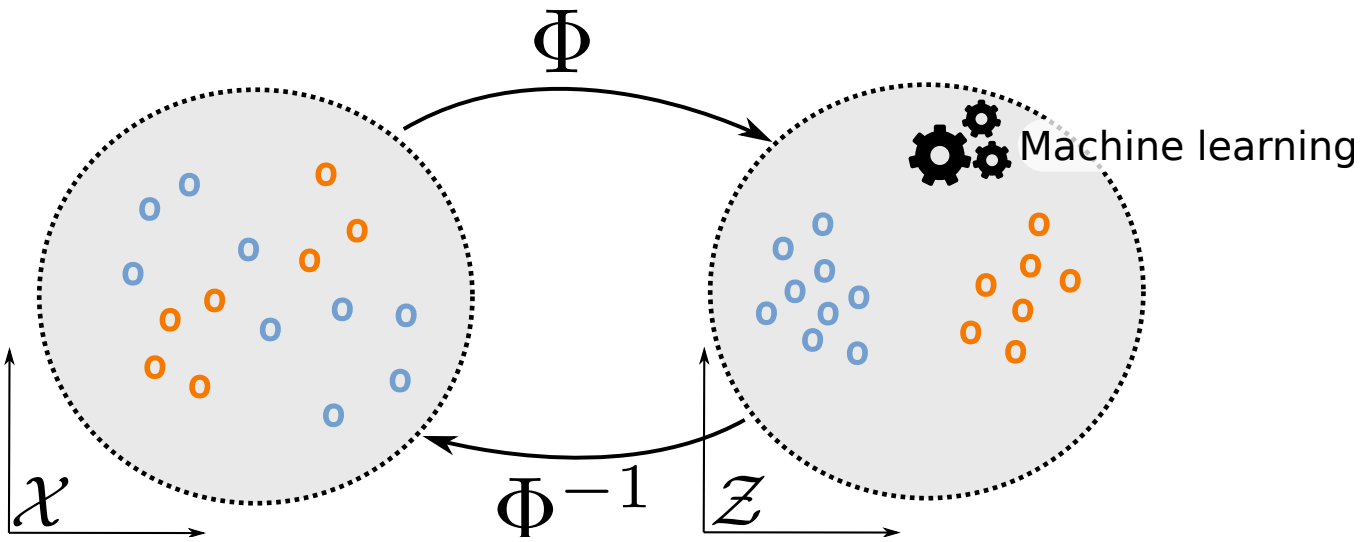


Figure – Illustration du cadre général de nos contributions.

- Modèle prédictif à base de NF.
- Application de méthode d'apprentissage machine dans l'espace du NF $\mathcal{Z}$.
- Contributions sur 2 tâches différentes : Classification et Régression.

NF pour la classification

Idée : Définition de distribution gaussienne pour chaque classe c paramétrées par $(\boldsymbol{\mu}_c, \Sigma_c)$ dans l'espace $\mathcal{Z}$.

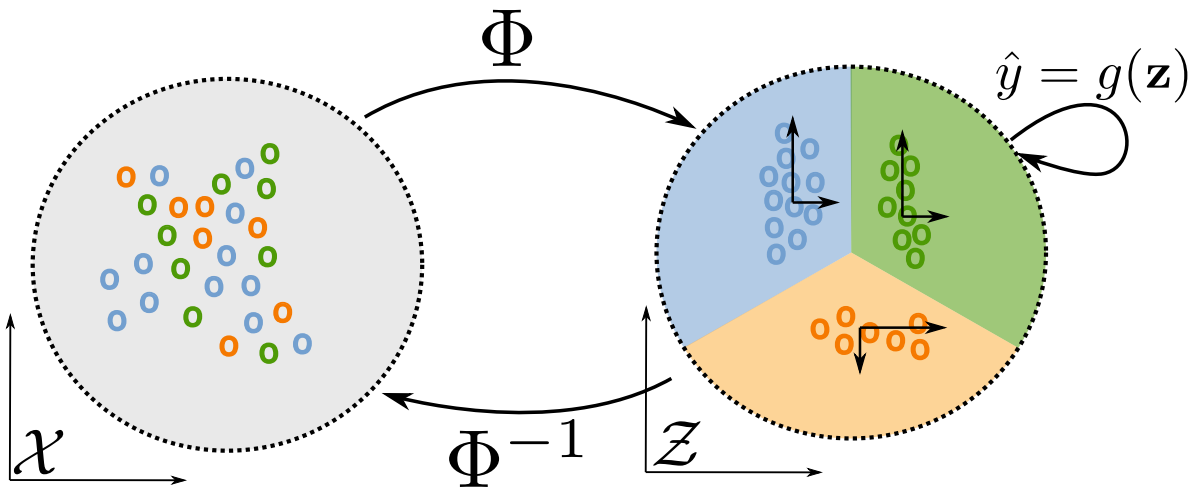


Figure – Illustration de l'approche de classification. Le processus d'apprentissage vise à séparer linéairement les données dans l'espace des caractéristiques $\mathcal{Z}$. Un classifieur linéaire $g : \mathcal{Z} \to \mathcal{Y}$ détermine la classe de chaque donnée. La pré-image de tout point $z \in \mathcal{Z}$ peut être retrouvée grâce à Φ^{-1}.

NF pour la classification

- Construction d'une distribution gaussienne par classe $c \in \{1 \dots C\}$.

$$\log P_{\mathcal{Z}}(z, c) = -\frac{1}{2}\left(d \log(2\pi) + (z - \mu_c)^\top \Sigma_c^{-1}(z - \mu_c)\right) - \log(\det(\Sigma_c))$$

- Apprentissage de la séparation des gaussiennes par leurs positions μ_c.

Optimisation d'un NF pour la classification

- Perte de vraisemblance du NF :
$\mathcal{L}_{\text{nf}}(x, c) = -\log P_{\mathcal{Z}}(\Phi(x), c) - \log \left|\det\left(\frac{\partial \Phi(x)}{\partial x}\right)\right|$
- Perte de séparation des gaussiennes :
$\mathcal{L}_{\mu} = -\log\left(1 + \frac{1}{C^2}\sum_{i=1}^{C}\sum_{j=1}^{C} \| \mu_i - \mu_j \|_2^2\right)$
- Perte totale : $\mathcal{L}(x, c) = \mathcal{L}_{\text{nf}}(x, c) + \beta \mathcal{L}_{\mu}$

NF pour la régression

Idée : Interpolations entre deux distributions gaussiennes paramétrées par (μ_1, Σ_1) et (μ_2, Σ_2), associées aux valeurs min et max de Y.

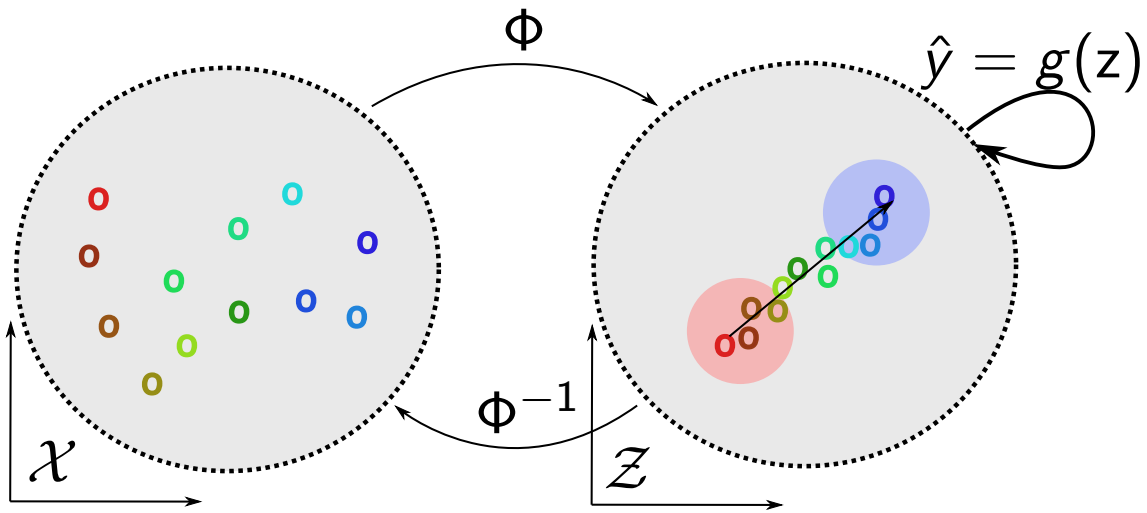


Figure – Illustration de l'approche pour une tâche de régression, où la couleur de chaque échantillon correspond à sa valeur quantitative. Un modèle linéaire $g : \mathcal{Z} \to \mathcal{Y}$ prédit la valeur quantitative de chaque donnée. La pré-image de tout point $z \in \mathcal{Z}$ peut être retrouvée grâce à Φ^{-1}.

NF pour la régression

- Construction de distribution gaussienne en fonction de $y \in \mathbb{R}$.

$$\log P_{\mathcal{Z}}(z, \boldsymbol{\mu}_y, \Sigma_y) = -\frac{1}{2}\left(d \log(2\pi) + (z - \boldsymbol{\mu})^\top \Sigma_y^{-1}(z - \boldsymbol{\mu}_y)\right) - \log(\det(\Sigma_y)).$$

- Paramètres calculés pour un y donné par :
  $\Sigma_y = \sigma^2 \mathbb{I}_d$,
  $\boldsymbol{\mu}_y = \tau_y \boldsymbol{\mu}_1 + (1 - \tau_y)\boldsymbol{\mu}_2$ avec $\tau_y = \frac{y - \min(Y)}{\max(Y) - \min(Y)}$
- Apprentissage de la séparation des gaussiennes par leurs positions $\boldsymbol{\mu}_1$ et $\boldsymbol{\mu}_2$.

Optimisation d'un NF pour la régression

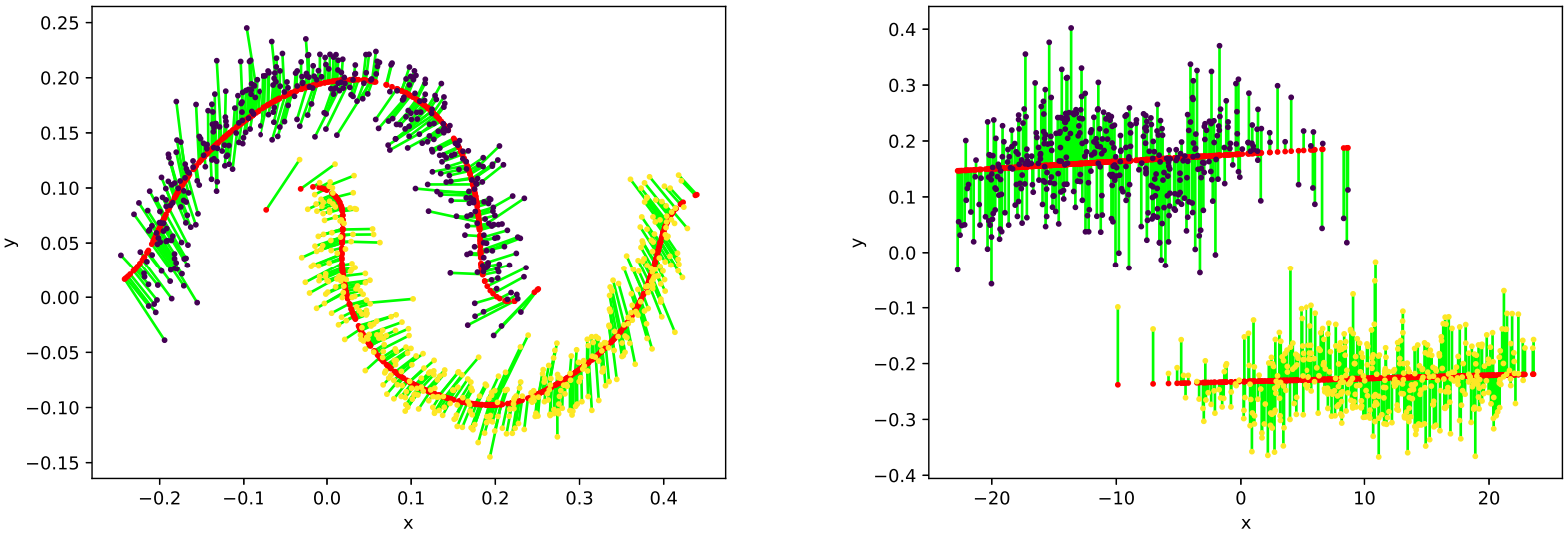
- Perte de vraisemblance du NF :
  $\mathcal{L}_{\text{nf}}(x, \boldsymbol{\mu}_y, \Sigma_y) = -\log P_{\mathcal{Z}}(\Phi(x), \boldsymbol{\mu}_y, \Sigma_y) - \log \left|\det\left(\frac{\partial \Phi(x)}{\partial x}\right)\right|$
- Perte de séparation des gaussiennes : $\mathcal{L}_\mu = -\log\left(1 + \| \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 \|_2^2\right)$
- Perte totale : $\mathcal{L}(x, y) = \mathcal{L}_{\text{nf}}(x, \boldsymbol{\mu}_y) + \beta \mathcal{L}_\mu$,

Performances de classification

Table – Performances de classification obtenues par nos approches basées sur les NF (RealNVP, FFJORD, et Glow) en comparaison aux méthodes SVM utilisant les noyaux linéaires, RBF, Polynomial et Sigmoid. Les meilleurs résultats sont mis en évidence en gras.

Datasets	SVM Linear	SVM RBF	SVM Poly	SVM Sigmoid	Méthode proposée RealNVP	FFJORD	Glow
double moon	93, 00%	**100,00%**	93, 00%	90, 00%	**100,00%**	**100,00%**	-
Iris	93, 34%	93, 34%	93, 34%	80, 00%	100, 00%	**100,00%**	-
Breast Cancer	94, 64%	94, 64%	94, 64%	96, 43%	**100,00%**	98, 21%	-
MNIST	94.63%	97.31%	97.68%	90.98%	-	-	**99.47%**

Capacité de génération



(a) Input space $\mathcal{X}$

(b) Feature space $\mathcal{Z}$

Figure – (a) Ensemble de données dans l'espace d'entrée $\mathcal{X}$. (b) Leurs représentations dans l'espace des caractéristiques $\mathcal{Z}$ obtenu par la transformation apprise Φ. Les points rouges correspondent aux projections linéaires dans l'espace de caractéristiques $\mathcal{Z}$ sur une composante principale.

Capacité de génération

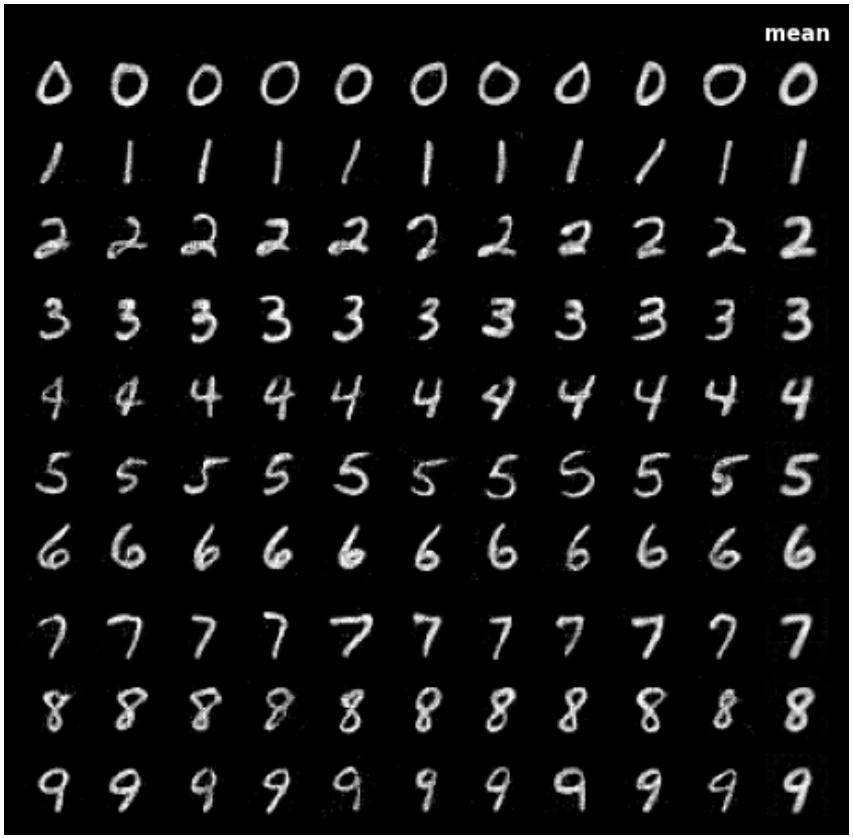


Figure – Pré-images obtenues en échantillonnant des vecteurs dans l'espace $\mathcal{Z}$ selon chaque distribution gaussienne associée à chaque classe.

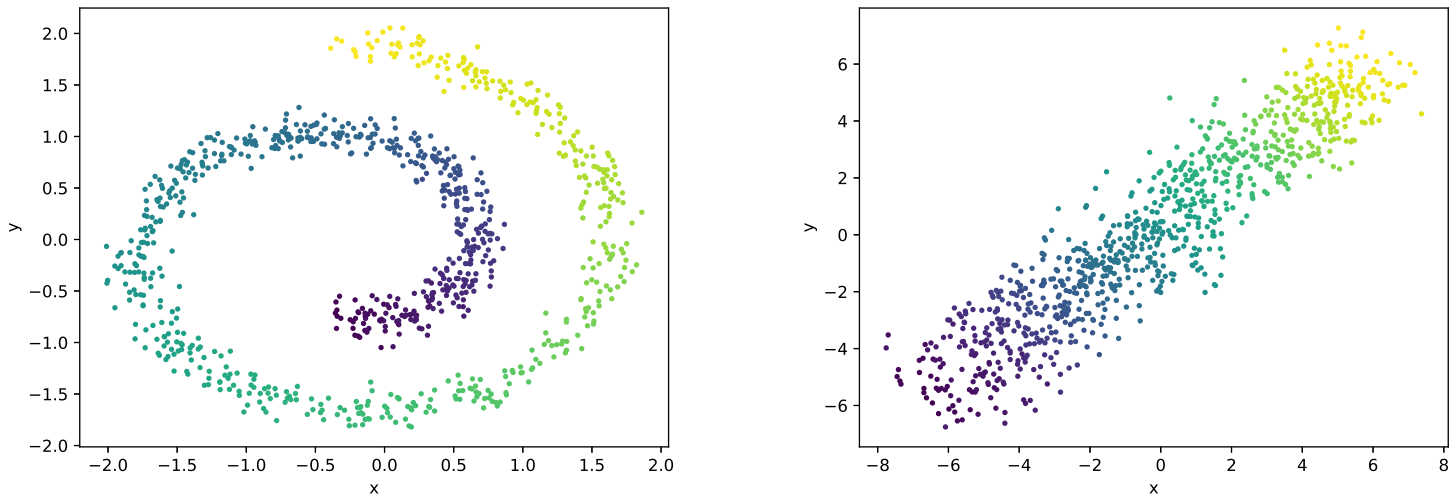
Figure – Illustration des pré-images d'interpolation dans $\mathcal{Z}$. Chaque ligne illustre deux visages sélectionnés (le visage le plus à gauche et le plus à droite) et 10 pré-images de 10 interpolations entre les représentations de ces deux visages dans l'espace latent.

Performances de régression

Table – Évaluation de notre approche dans une configuration de régression en appliquant la régression Ridge dans $\mathcal{Z}$ et en calculant le coefficient de détermination R^2 sur les résultats prédits en comparaison avec les méthodes kernel-Ridge utilisant les kernels suivants : Linear, RBF, Polynomial et Sigmoid.

					Méthode proposée	
	Linear	RBF	Polynomial	Sigmoid	RealNVP	FFJORD
swiss roll	0.012	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
diabetes	0.452	0.458	0.461	0.454	**0.548**	0.463
QSAR aquatic toxicity	0.392	0.411	0.381	0.403	**0.565**	0.508
QSAR fish toxicity	0.567	0.600	0.603	0.567	0.635	**0.661**

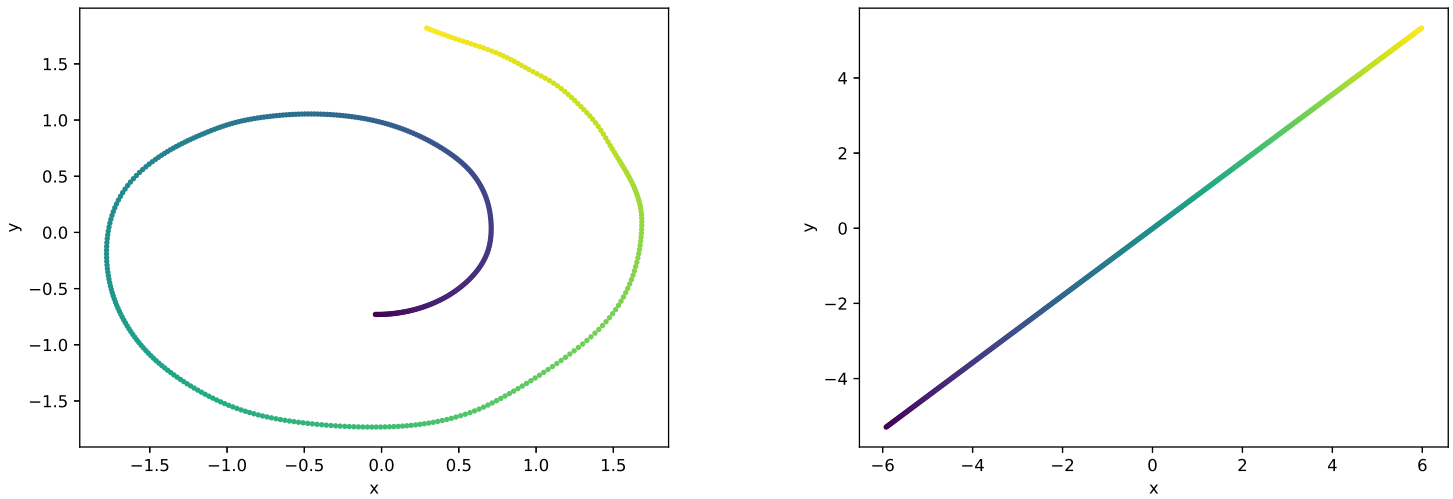
Capacité de génération



(a) $\mathcal{X}$ (b) $\mathcal{Z}$

Figure – (a) Ensemble de données dans l'espace d'entrée $\mathcal{X}$. (b) Leurs représentations dans l'espace des caractéristiques $\mathcal{Z}$ obtenu par la transformation apprise Φ.

Capacité de génération



(a) $\mathcal{X}$ (b) $\mathcal{Z}$

Figure – (a) Pré-images générées à partir de points de $\mathcal{Z}$ obtenus par la transformation inverse apprise Φ^{-1} sur l'ensemble de données. (b) 500 points dans $\mathcal{Z}$ échantillonnés uniformément entre les valeurs min et max de Y.

Sommaire

- Utilisation de représentation de graphe sous forme de matrice de caractéristiques et d'adjacence :
  $G = (X, A) \in \mathbb{R}^{n \times d} \times \mathbb{R}^{n \times n \times e}$.

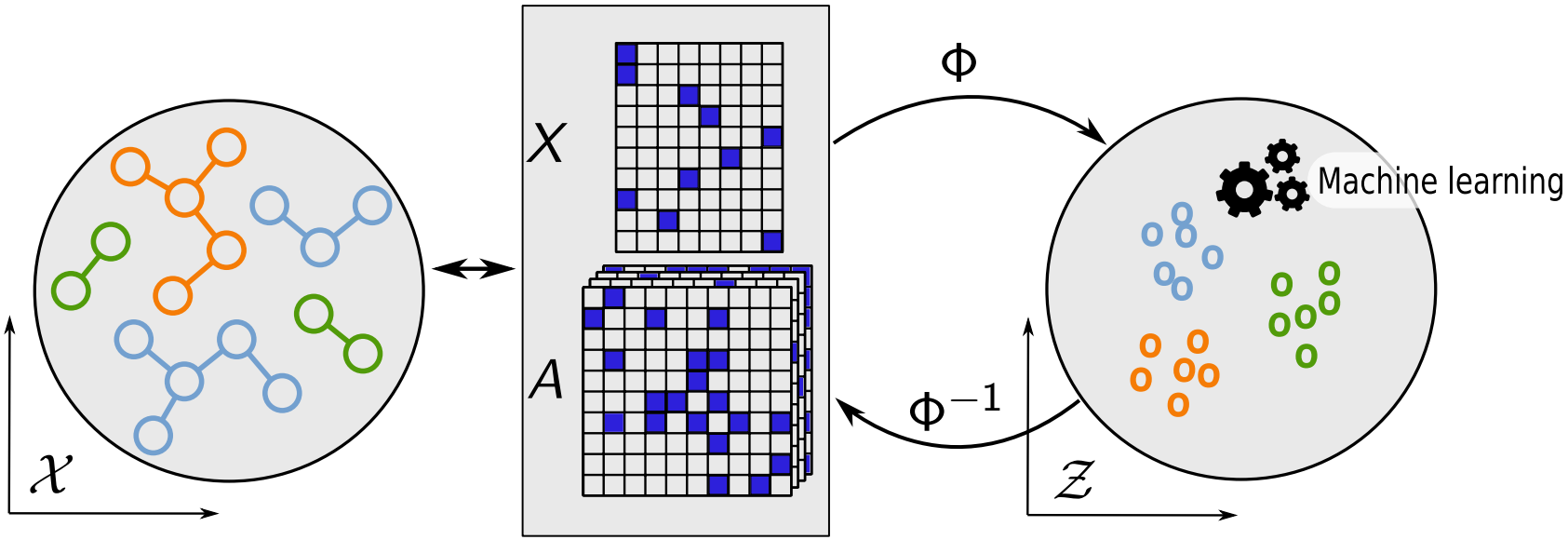


Figure – Illustration du cadre général de nos contributions sur des données structurées de type graphe.

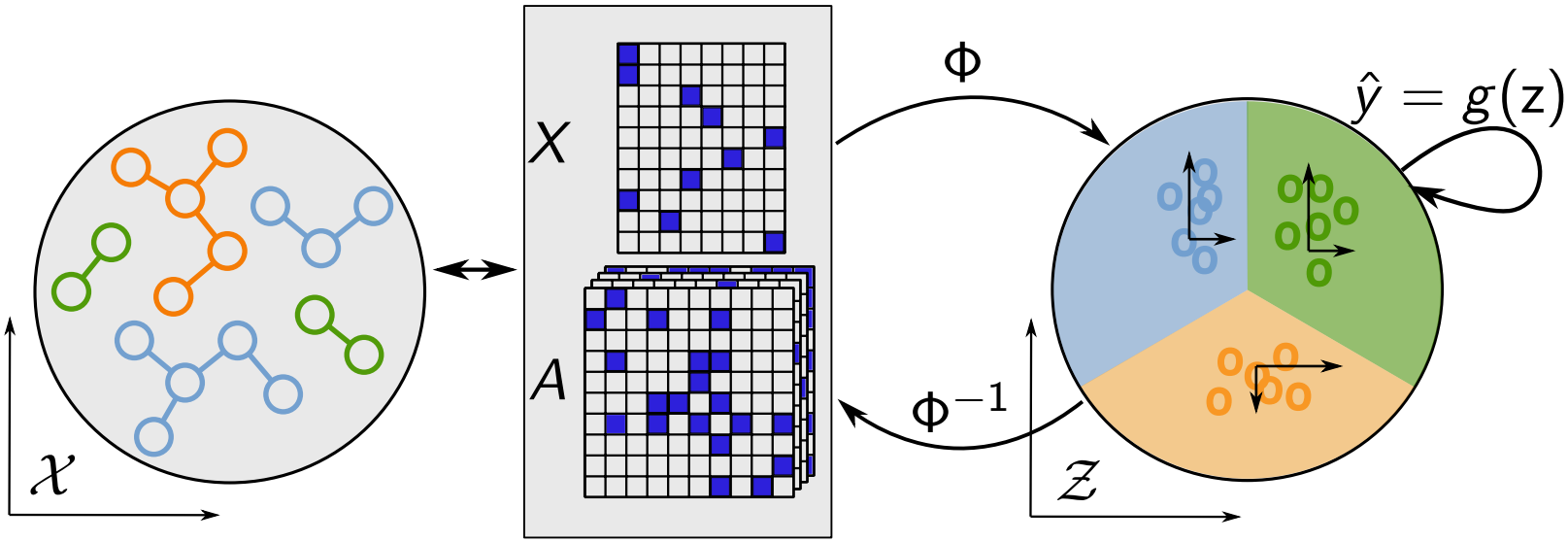


Figure – Illustration de notre approche pour une tâche de classification. Le processus d'apprentissage vise à séparer linéairement les données dans l'espace des caractéristiques $\mathcal{Z}$. Un classifieur linéaire $g : \mathcal{Z} \to \mathcal{Y}$ détermine la classe de chaque donnée. La pré-image de tout point $z \in \mathcal{Z}$ peut être retrouvée grâce à Φ^{-1}.

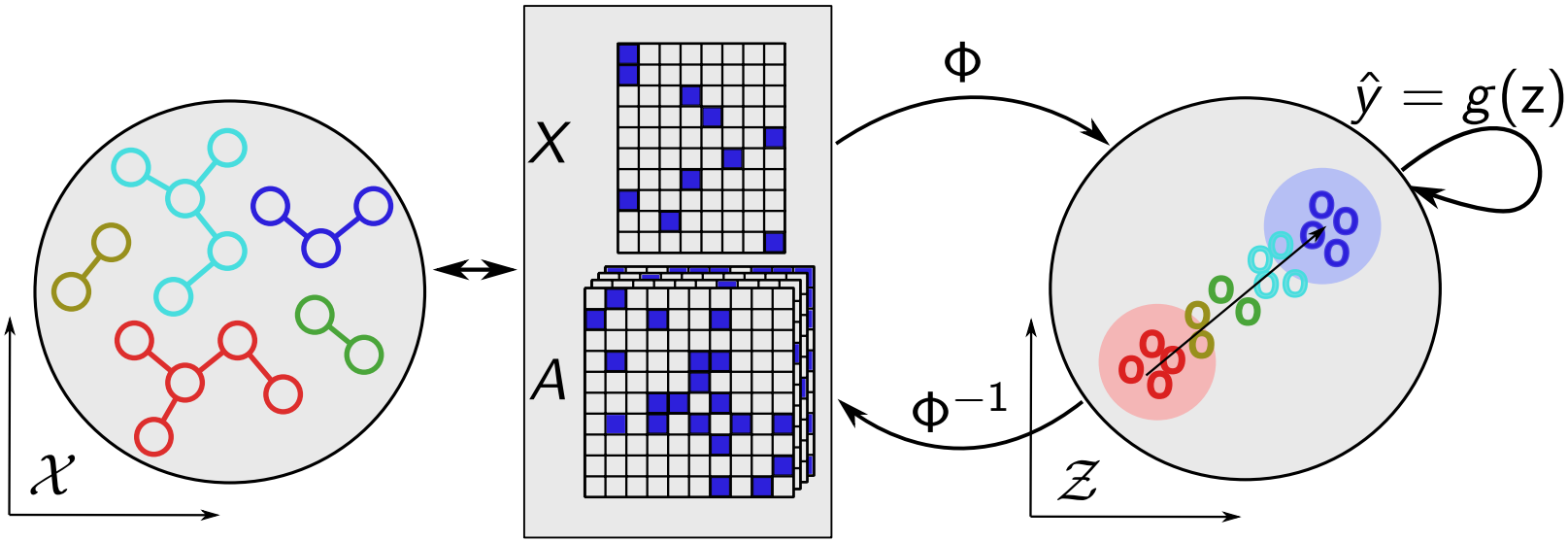


Figure – Illustration de l'approche pour une tâche de régression, où la couleur de chaque échantillon correspond à sa valeur quantitative. Un modèle linéaire $g : \mathcal{Z} \to \mathcal{Y}$ prédit la valeur quantitative de chaque donnée. La pré-image de tout point $z \in \mathcal{Z}$ peut être retrouvée grâce à Φ^{-1}.

Performances de classification de graphe

Table – Score de classification sur les ensembles de données de graphes de noeuds étiquetés par rapport aux méthodes kernel-SVC.

	Kernel								GraphKernel				Our approach	
	Linear		RBF		Polynomial		Sigmoid		WL		SP		Hadcode		MoFlow	
Datasets	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
MUTAG	0.761	0.070	0.772	0.039	0.717	0.046	0.683	0.056	0.778	0.0	0.778	0.0	0.778	0.0	**0.939**	0.016

Table – Score de classification sur les ensembles de données composés de noeud attribué par rapport aux méthodes kernel-SVC.

	Kernel								GraphKernel				Our approach	
	Linear		RBF		Polynomial		Sigmoid		Prop		MSLap		MoFlow	
Datasets	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Letter-med	0.414	0.022	0.419	0.022	0.427	0.028	0.281	0.024	0.298	0.135	0.410	0.039	0.752	0.012
Letter-med (+10)	0.414	0.022	0.419	0.022	0.427	0.028	0.281	0.024	0.298	0.135	0.410	0.039	**0.867**	0.012

Performances de régression de graphe

Table – **Coefficient de détermination** R^2 comparé aux méthodes kernel-Ridge.

	Kernel								GraphKernel						Our approach	
	Linear		RBF		Polynomial		Sigmoid		WL		SP		Hadcode		MoFlow	
Datasets	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
QM7	0.681	0.001	0.680	0.002	0.681	0.001	0.673	0.002	0.49	0.0	0.721	0.0	0.491	0.0	**0.730**	0.008
QM9	0.002	0.001	0.0	0.0	0.0	0.002	0.0	0.0	0.065	0.0	0.036	0.0	0.041	0.0	**0.218**	0.007
ESOL	0.555	0.032	0.558	0.032	0.566	0.034	0.563	0.037	0.602	0.0	0.531	0.0	0.573	0.0	**0.685**	0.039
FREESOLV	0.254	0.114	0.262	0.113	0.264	0.102	0.255	0.074	**0.895**	0.0	0.543	0.0	**0.901**	0.0	0.754	0.042

Capacité de génération de graphes

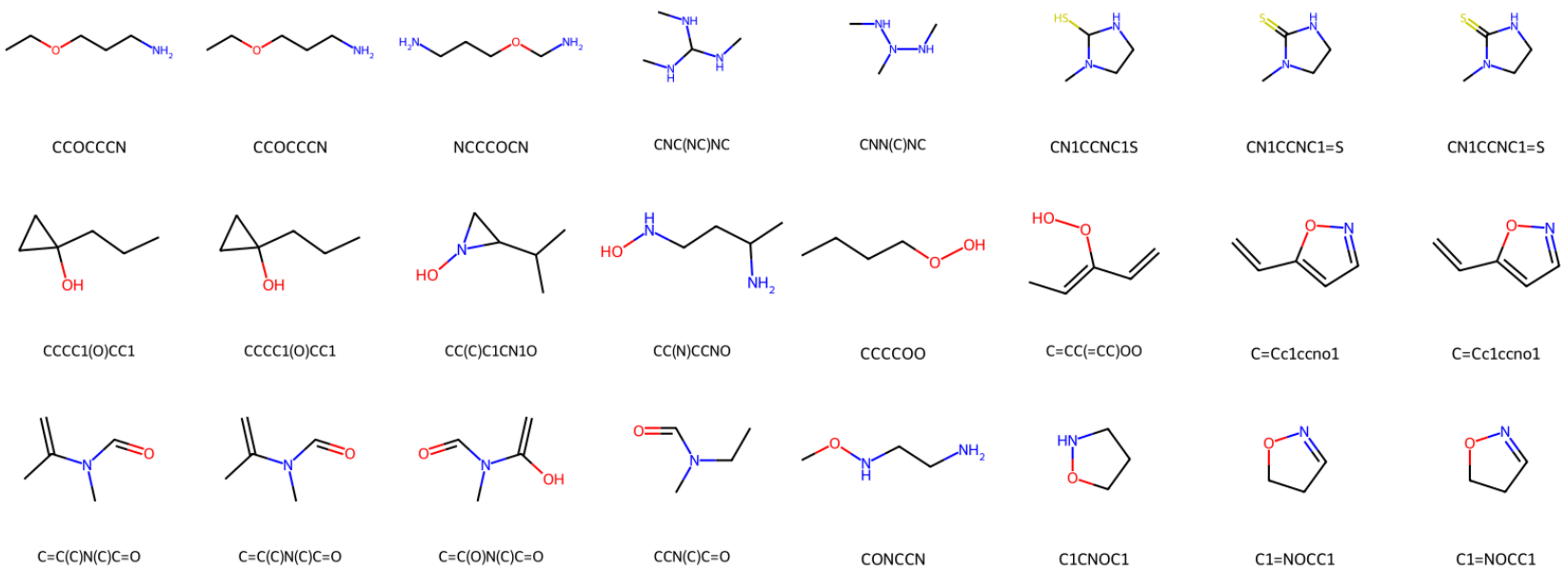


Figure – Pré-images d'interpolation dans $\mathcal{Z}$. Chaque ligne illustre deux molécules sélectionnés (le plus à gauche et le plus à droite) et les pré-images d'interpolations entre les représentations de ces deux graphes dans l'espace latent.

Sommaire

Conclusion

- Contributions utilisant les architectures NF pour l'apprentissage machine avec pré-image.
- Bonnes performances sur des données vectorielles et images dans les tâches de classification et de régression en utilisant des méthodes simples dans l'espace des caractéristiques.
- Bons premiers résultats sur les graphes en régression et en classification.