Détection des Données Hors Distribution : Une Approche Basée sur un Auto-Encodeur Variationnel Structuré

Maxime OSSONCE [1,3] Pierre DUHAMEL[2] Florence ALBERGE[3]

[1]ESME, 94200 Ivry-sur-Seine, France

[2]Laboratoire des signaux et systèmes (L2S), Université Paris-Saclay, CNRS, CentraleSupelec, 91190 Gif-sur-Yvette, France

[3]SATIE, Université Paris-Saclay, ENS Paris-Saclay, CNRS, 91190 Gif-sur-Yvette, France

Résumé – Les modèles d'intelligence artificielle (IA) sont généralement entraînés sur des ensembles de données représentatifs de leur tâche. Cependant, leur performance peut être gravement affectée lorsqu'ils rencontrent des données hors distribution (OoD). Cet article explore une approche fondée sur un auto-encodeur variationnel structuré, permettant de mieux distinguer les données OoD des données légitimes sans nécessiter un ensemble d'entraînement OoD spécifique mais en tirant partie des échantillons OoD disponibles lors du déploiement. En particulier, nous montrons l'utilité d'un travail en deux étapes, et d'un "padding" à l'aide de données OoD connues et de types divers, ce qui permet une amélioration sensible des performances.

Abstract – Artificial intelligence (AI) models are typically trained on datasets representative of their task. However, their performance can be severely impacted when encountering out-of-distribution (OoD) data. This paper explores an approach based on a structured variational autoencoder, allowing for better differentiation between OoD data and legitimate data without requiring a specific OoD training set, but instead leveraging available OoD samples during deployment. In particular, we demonstrate the usefulness of a two-step approach and "padding process" with known OoD data of various types, which leads to a significant improvement in performance.

1 Introduction

Les modèles d'intelligence artificielle (IA) sont généralement entraînés sur des données In-Distribution (InD), représentatives de leur tâche. Lorsqu'ils rencontrent des données hors distribution (Out-of-Distribution, OoD), leurs performances peuvent chuter. La détection des données OoD permet ainsi d'éviter des erreurs critiques, d'améliorer les performances des modèles et de renforcer la confiance des utilisateurs.

De nombreuses méthodes de détection des données OoD ont été proposées dans la littérature, notamment basées sur l'établissement d'un score dérivé des sorties du réseau de neurones et/ou des couches intermédiaires [7, 6, 3]. En général, ces méthodes n'utilisent pas de données OoD (ou seulement un faible nombre pour le réglage des hyperparamètres). Une autre catégorie de méthodes repose sur l'utilisation d'un ensemble d'échantillons OoD, collectés et étiquetés avant l'entraînement, pour apprendre à différencier les données InD des données OoD [4, 12, 3]. Ces méthodes s'avèrent plus efficaces que les premières lorsque les données OoD rencontrées en déploiement ressemblent à celles vues lors de l'entraînement. Toutefois, elles peuvent échouer lorsque les données OoD diffèrent substantiellement. De plus, ces approches supposent une disponibilité préalable de données OoD pour l'entraînement, une hypothèse qui peut être difficilement réalisable en pratique [12]. La détection de données OoD est une tâche difficile pour deux raisons : la connaissance partielle ou inexistante des distributions $\mathbb{P}_{\text{in}}$ et $\mathbb{P}_{\text{out}}$ dont sont issues les données InD et OoD, la difficulté à disposer d'un ensemble $\mathcal{S}_{\text{out}}$ d'échantillons représentatifs de la distribution $\mathbb{P}_{\text{out}}$.

Notre méthode surmonte ces limitations en utilisant un auto-encodeur variationnel qui structure l'espace latent afin de séparer les données InD et OoD selon des distributions distinctes. Elle repose sur un ensemble d'échantillons collectés librement lors du déploiement, constituant un mélange non étiqueté de données InD et OoD, évitant ainsi les problèmes liés à la constitution de l'ensemble $\mathcal{S}_{\text{out}}$.

2 Problème et hypothèses

Nous supposons que la tâche principale du modèle est la classification. Soit $\mathcal{X}$ l'ensemble des entrées et $\mathcal{Y} = \{1, \dots, K\}$ celui des étiquettes. L'ensemble d'apprentissage, noté $S_{\text{in}}^{\text{train}} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, est issu de $\mathbb{P}_{\mathcal{X}\mathcal{Y}}$, dont la distribution marginale sur $\mathcal{X}$ est $\mathbb{P}_{\text{in}}$.

Nous considérons le cas classique où l'algorithme est entraîné sur des données étiquetées, puis déployé dans un environnement contenant d'éventuelles données OoD de classe inconnue, que le modèle ne doit pas prédire. L'objectif est donc de déterminer, au déploiement, si une entrée est InD ou OoD. Pour cela, nous utilisons le modèle de contamination de Huber pour caractériser la distribution marginale des données mixtes observées :

$$\mathbb{P}_{\text{mix}} = (1 - \pi)\mathbb{P}_{\text{in}} + \pi\mathbb{P}_{\text{out}} \tag{1}$$

avec $\pi \in [0, 1[$. Nous notons $\mathcal{S}_{\text{mix}}$ un ensemble de m échantillons issus de $\mathbb{P}_{\text{mix}}$. Nous supposons que le modèle peut supporter un traitement différé dans lequel les m échantillons de $\mathcal{S}_{\text{mix}}$ ne seront pas traités au fil de l'eau mais en bloc. Le modèle (1) est similaire à celui utilisé dans [5, 1, 2]. Dans notre méthode, l'ensemble $\mathcal{S}_{\text{mix}}$ est l'ensemble sur lequel doivent être prises les décisions alors que, dans [5, 1], $\mathcal{S}_{\text{mix}}$ est utilisé à l'entraînement ce qui peut conduire à une baisse de performance lorsque $\mathbb{P}_{\text{out}}^{\text{train}} \neq \mathbb{P}_{\text{out}}^{\text{test}}$. Inspirés de [2], nous avons

proposé dans [9] une méthode originale de classification et de détection conjointe d'OoD. Cet article approfondit sa mise en œuvre dans un environnement réaliste.

3 Méthode proposée

3.1 Classification

Nous présentons tout d'abord le modèle de classification. Nous utilisons un auto-encodeur variationnel (VAE) dont l'espace latent sera structuré pour faciliter la tâche de classification (CVAE). Un VAE est constitué d'un encodeur et d'un décodeur. L'encodeur génère, pour une entrée $x \in \mathcal{X}$, les paramètres d'une distribution variationnelle $q_\phi(z|x)$ selon laquelle on peut générer une variable latente $z \in \mathbb{R}^\kappa$ qui constitue une représentation de x. Pour une réalisation z de la variable latente, le décodeur renvoie les paramètres de $p_\theta(x|z)$ permettant d'échantillonner une estimation de x. L'approche variationnelle tient du fait que l'encodeur $q_\phi(z|x)$ sera une approximation variationnelle de l'a posteriori $p_\theta(z|x)$, lui-même intractable : $q_\phi(z|x)$ sera la distribution qui minimisera la divergence de Kullback-Leibler (KL) entre les deux densités $q_\phi(z|x)$ et $p_\theta(z|x)$. L'objectif d'un classifieur génératif est d'apprendre $p(x|y)$. Une borne inférieure de l'évidence (ELBO) sur $p(x|y)$ est utilisée comme critère afin d'entraîner le modèle :

$$\begin{aligned} \text{elbo}_{\text{CVAE}}(x|y) &= \log p_{\phi,\theta}(x|y) - \text{KL}[q_\phi(z|x)\|p_\theta(z|x)] \quad (2)\\ &= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(z|x)] \\ &\quad - \text{KL}[q_\phi(z|x)\|p_\phi(z|y)] \quad (3)\\ &= -\mathcal{L}_{\text{CVAE}} \end{aligned}$$

Notre modèle diffère du CVAE de [11] dans le conditionnement de l'encodeur : ici on suppose une chaîne de Markov $Y \to X \to Z$. Afin de structurer l'espace latent et faciliter ainsi la classification, nous imposons $p_\theta(z|y) = \mathcal{N}(z|m^y, I_\kappa)$ pour $y \in \mathcal{Y}$. L'*a priori* $p_\theta(z)$ est alors un mélange de Gaussiennes (GMM). Les centroïdes $\{m^y\}_{y\in\mathcal{Y}}$ peuvent être vus comme les représentants des classes dans l'espace latent, fixés avant l'entraînement du CVAE. En choisissant $p_\theta(x|z) = \mathcal{N}(x|f_\theta(z), \sigma_\theta^2 I_d)$ (σ_θ appris), le premier terme de (3) fait apparaître l'erreur de reconstruction entre l'entrée x et la sortie $\hat{x} = f_\theta(z)$:

$$\log p_\theta(z|x) = -\frac{d}{2}\log(2\pi\sigma_\theta^2) - \frac{1}{2\sigma_\theta^2}\|x - f_\theta(z)\|^2 \quad (4)$$

Enfin, $q_\phi(z|x)$ est choisi tel que $q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \Lambda_\phi(x))$ où $\Lambda_\phi(x))$ est une matrice diagonale. La KL prend alors la forme suivante :

$$\begin{aligned} \text{KL}[q_\phi(z|x)\|p_\theta(z|y)] &= \frac{1}{2}\|m^y - \mu_\phi(x)\|^2 - \frac{\kappa}{2} \quad (5)\\ &+ \frac{1}{2}(tr\Lambda_\phi(x) - \log\det\Lambda_\phi(x)) \end{aligned}$$

La minimisation de la KL permet ainsi de structurer l'espace latent autour de chacun des centroïdes. Au moment du test, l'estimation de la classe pourra alors se faire de manière très simple par une recherche de plus proche voisin selon :

$$\hat{y} = \arg\max_y \text{elbo}_{\text{CVAE}}(x|y) = \arg\min_y \|m^y - \mu_\phi(x)\|^2 \quad (6)$$

La position des centroïdes $\{m^y\}_{y\in\mathcal{Y}}$ peut être fixée comme dans cet article ou apprise au cours de l'entraînement (voir [9] pour le deuxième cas).

3.2 Détection d'OoD

La détection des échantillons hors distribution (OoD) est une tâche difficile en raison de notre manque de connaissance sur les distributions $\mathbb{P}_{\text{in}}(x)$ et $\mathbb{P}_{\text{out}}(x)$. La structuration de l'espace latent permet d'y remédier. En effet, les échantillons InD sont censés y être distribués selon la loi *a priori* $p(z)$ (GMM). Nous définissons maintenant un second *a priori* pour les échantillons OoD selon $p'(z) = \mathcal{N}(z|m', I_\kappa)$. Choisir m' suffisamment éloigné des centroïdes $\{m^y\}_{y\in\mathcal{Y}}$ conduit à assigner deux espaces distincts aux InD et aux OoD. Par ailleurs, la distance $d = \max_y \|m^y - m'\|$ joue un rôle essentiel dans les performances de la méthode (voir les résultats analytiques de [9]). Ce paramètre est totalement maîtrisé par l'utilisateur qui peut choisir m' conformément à la valeur souhaitée pour d. Afin de déplacer, dans l'espace latent, les représentants des échantillons OoD à proximité de m', nous allons utiliser à nouveau le principe du VAE permettant de calculer une ELBO :

$$\begin{aligned} \text{elbo}_{\text{VAE}}(x) &= \log p_{\phi,\theta}(x) - \text{KL}[q_\phi(z|x)\|p_\theta(z|x)] \quad (7)\\ &= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(z|x)] \\ &\quad - \text{KL}[q_\phi(z|x)\|p'_\phi(z)] \quad (8)\\ &= -\mathcal{L}_{\text{VAE}}(\phi,\theta;x) \end{aligned}$$

Le premier terme prend la même forme que précédemment et la KL s'écrit :

$$\begin{aligned} \text{KL}[q_\phi(z|x)\|p'_\phi(z)] &= \frac{1}{2}\|m' - \mu_\phi(x)\|^2 - \frac{\kappa}{2} \quad (9)\\ &+ \frac{1}{2}(tr\Lambda_\phi(x) - \log\det\Lambda_\phi(x)) \end{aligned}$$

Conformément à nos hypothèses, nous rappelons que nous ne disposons pas d'un ensemble « pur » d'OoD qui pourrait nous permettre d'entraîner ce VAE. Nous disposons en revanche d'un ensemble mixte *non étiqueté* $\mathcal{S}_{\text{mix}}$ contenant un mélange de données InD et OoD et d'un classifieur (CVAE) déjà entraîné à l'aide des données InD de $\mathcal{S}_{\text{in}}^{\text{train}}$. Nous proposons d'en tirer profit en réalisant un ajustement fin (*fine-tuning*) avec la fonction de coût [2, 9] :

$$\begin{aligned} \mathcal{L} = &\frac{1}{|\mathcal{S}_{\text{in}}^{\text{train}}|}\sum_{(x,y)\in\mathcal{S}_{\text{in}}^{\text{train}}} \mathcal{L}_{\text{CVAE}}(\phi,\theta;x,y) \quad (10)\\ &+\frac{\alpha}{|\mathcal{S}_{\text{mix}}\cup\mathcal{P}|}\sum_{x\in\mathcal{S}_{\text{mix}}\cup\mathcal{P}} \mathcal{L}_{\text{VAE}}(\phi,\theta;x) \end{aligned}$$

À chaque nouveau lot $\mathcal{S}_{\text{mix}}$ traité, l'état du modèle est repris tel qu'il était à l'issue de son ajustement sur le jeu d'entraînement pour la classification. Le rôle et le contenu de l'ensemble $\mathcal{P}$ seront explicités dans la section 4. Le premier terme de (10) permet de conserver la configuration initiale de l'espace latent et le deuxième terme la complète en ajoutant une zone pour les OoD. Au moment du test, les échantillons OoD devraient ainsi être attirés vers le centroïde m' alors que les échantillons InD devraient rester proches du centroïde correspondant à la classe estimée. Le score permettant la distinction InD/OoD

s'écrit à partir d'un ratio de vraisemblance selon :

$$s(x) = \log \frac{\widetilde{\mathbb{P}}_{\text{in}}(\mu_\phi(x))}{\widetilde{\mathbb{P}}_{\text{out}}(\mu_\phi(x))} \approx \frac{1}{2}\|m' - \mu_\phi(x)\|^2 - \frac{1}{2}\|m^{\hat{y}} - \mu_\phi(x)\|^2 \quad (11)$$

où $\hat{y}$ est la classe estimée par (6) lors de la classification qui est réalisée préalablement à l'ajustement fin. Le score $s(x)$ est ensuite calculé pour tout $x \in \mathcal{S}_{\text{mix}}$. Si $s(x) > T$ alors x sera considéré comme InD et comme OoD sinon. Le seuil T pourra être fixé afin de satisfaire un TPR (*True Positive Rate*) minimum. Il est important de souligner ici une différence majeure avec [5, 1]. Dans notre méthode, $\mathcal{S}_{\text{mix}}$ est l'ensemble (constitué avec les échantillons arrivant au fil de l'eau et traités en bloc) sur lequel nous prenons des décisions alors que dans [5, 1], $\mathcal{S}_{\text{mix}}$ est utilise comme ensemble d'entraînement ce qui expose à une perte d'efficacité lorsque $\mathbb{P}^{\text{train}}_{\text{mix}} \neq \mathbb{P}^{\text{test}}_{\text{mix}}$. Par ailleurs, afin de ne pas entraîner de délai important dans notre procédure nous considérons que $|\mathcal{S}_{\text{mix}}|$ est de l'ordre de quelques dizaines à quelques centaines d'échantillons alors que [5, 1] considèrent un ensemble 100 à 1000 fois plus grand.

Nous montrons dans la section suivante comment implémenter la méthode dans un environnement réaliste avec un gain important des performances par rapport à [2].

4 Résultats expérimentaux

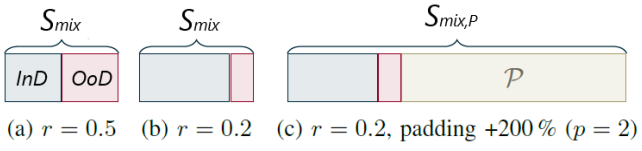
4.1 Protocole expérimental

La méthode proposée vise à identifier les données InD et OoD au sein de l'ensemble $\mathcal{S}_{\text{mix}}$ (données à tester). Par définition, $\mathcal{S}_{\text{mix}} = \mathcal{S}^{\text{in}}_{\text{mix}} \cup \mathcal{S}^{\text{out}}_{\text{mix}}$, où $\mathcal{S}^{\text{in}}_{\text{mix}}$ désigne le sous-ensemble de $\mathcal{S}_{\text{mix}}$ composé uniquement de données InD, tandis que $\mathcal{S}^{\text{out}}_{\text{mix}}$ ne contient que des données OoD. Soit $m = |\mathcal{S}_{\text{mix}}|$ et soit r le ratio d'OoD dans $\mathcal{S}_{\text{mix}}$, alors $|\mathcal{S}^{\text{in}}_{\text{mix}}| = rm$ et $|\mathcal{S}^{\text{out}}_{\text{mix}}| = (1-r)m$. Nous montrons sur la figure 2 (graphe de gauche) que la proportion r d'OoD dans $\mathcal{S}_{\text{mix}}$ est un facteur déterminant pour séparer efficacement les échantillons. La méthode est d'autant plus efficace que ce ratio est important. L'ensemble $\mathcal{S}_{\text{mix}}$ étant totalement inconnu, nous n'avons aucune garantie sur la valeur de r (il est toutefois fixé lors des expérimentations pour établir les performances du modèle). Afin de contourner ce problème, nous proposons ici de compléter l'ensemble à tester par des échantillons OoD connus (bourrage ou *padding*). Pour ce faire, nous construisons un nouvel ensemble $\mathcal{S}_{\text{mix},\mathcal{P}} = \mathcal{S}_{\text{mix}} \cup \mathcal{P}$, avec $|\mathcal{P}| = pm$, où $\mathcal{P}$ contient des échantillons OoD connus (voir figure 1). Nous ne faisons aucune hypothèse sur une relation entre $\mathcal{S}^{\text{out}}_{\text{mix}}$ et $\mathcal{P}$; en particulier, il n'est pas nécessaire que les échantillons de $\mathcal{P}$ suivent la loi $\mathbb{P}^{\text{test}}_{\text{out}}$. L'impact positif de $\mathcal{P}$ sur les performances est visible sur la figure 2 (droite) où $p = 0\%$ correspond à la méthode [2] appliquée au CVAE.

Nous proposons d'examiner dans la section 4.2 la composition de $\mathcal{P}$ dans le but d'optimiser l'efficacité de la méthode. Nous évaluerons ensuite les performances dans deux scénarios : (i) les échantillons OoD de $\mathcal{S}^{\text{out}}_{\text{mix}}$ proviennent d'une même distribution (ensemble homogène) ; (ii) les OoD peuvent provenir de plusieurs distributions distinctes (ensemble hétérogène), ce qui reflète mieux la réalité. Ce dernier point, ainsi que les stratégies de construction de $\mathcal{P}$, n'ont pas été abordés dans [9].

Les métriques d'évaluation pour la détection d'OoD sont soit l'AUROC, soit le FPR calculé pour un TPR de 95% (FPR@0.95). Le TPR et le FPR sont définis comme suit : $\text{TPR} = \frac{\sum_{x \in \mathcal{S}^{\text{in}}_{\text{mix}}} \mathbb{1}_{s(x)>T}}{|\mathcal{S}^{\text{in}}_{\text{mix}}|}$ et $\text{FPR} = \frac{\sum_{x \in \mathcal{S}^{\text{out}}_{\text{mix}}} \mathbb{1}_{s(x)>T}}{|\mathcal{S}^{\text{out}}_{\text{mix}}|}$. Il convient de noter que seules les données de $\mathcal{S}_{\text{mix}}$ (et non celles de $\mathcal{P}$) sont prises en compte dans les évaluations. Nous évaluerons également les performances de classification (accuracy) selon la formule suivante sur le jeu de test $\mathcal{S}^{\text{test}}_{\text{in}}$:

$$\text{Acc} = \frac{\sum_{(x,y) \in \mathcal{S}^{\text{test}}_{\text{in}}} \mathbb{1}_{y = \arg\min_{y'} \|m^{y'} - \mu_\phi(x)\|^2}}{|\mathcal{S}^{\text{test}}_{\text{in}}|}$$



(a) $r = 0.5$ (b) $r = 0.2$ (c) $r = 0.2$, padding +200 % ($p = 2$)

FIGURE 1 : Composition de $\mathcal{S}_{\text{mix}}$ et $\mathcal{S}_{\text{mix},\mathcal{P}} = \mathcal{S}_{\text{mix}} \cup \mathcal{P}$.

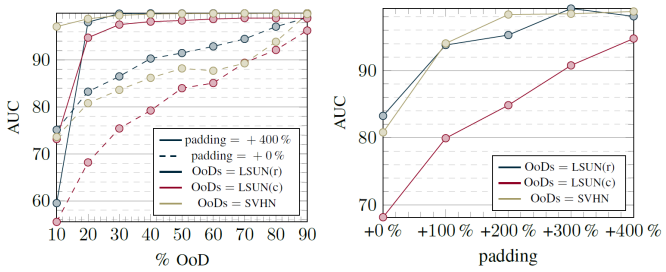


FIGURE 2 : Gauche : Impact de r (% OoD) sur l'AUROC pour $p = 0\%$ et $p = 400\%$. Droite : Impact de p (remplissage) sur l'AUROC pour $r = 20\%$.

	LSUNR		SVHN		LSUNC		CIFAR100		Moyenne	
	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑
C-N	38,0	92,9	**8,9**	**97,8**	28,5	95,5	66,8	78,8	35,6	91,3
C-N-T	**26,2**	94,2	16,0	96,4	**24,8**	**96,0**	59,8	83,5	**31,7**	92,5
C-N-T-R	31,6	94,3	26,4	95,2	34,4	95,0	**58,3**	86,1	37,7	**92,6**
R	32,4	**94,7**	60,5	91,6	62,6	91,9	60,8	**87,6**	54,1	91,5

TABLE 1 : Comparaison des performance pour différents jeux de bourrage sur les batchs *homogènes*, $r = 10\,\%$, $p = 700\,\%$.

L'architecture retenue pour l'encodeur du CVAE est le réseau convolutionnel VGG19 [10]. Nous retenons cette même architecture pour implanter les méthodes présentées table 2. Le décodeur du CVAE est constitué de couches convolutionnelles transposées. Les centroïdes m^y de l'a priori sur z sont fixés à l'entraînement du CVAE après tirage selon une loi normale de variance 100.

Le centroïde m' du second a priori $p'(z)$ est fixé au début de l'ajustement fin selon le même tirage que les centroïdes de l'a priori du CVAE. La pondération des composantes (10) retenue est $\alpha = 0{,}2$.

4.2 Résultats

La table 1 compare les performances obtenues pour différentes compositions de $\mathcal{P}$ en utilisant quatre types d'images : monochromes (C), bruit aléatoire (N), images issues du dataset Textures (T) et du dataset générique 300K random Images

	CIFAR10	LSUNR		SVHN		LSUNC		CIFAR100		Moyenne	
	Acc.	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑	FPR↓	AUC↑
Mahala	90,5	79,4	75,8	90,7	52,3	87,5	58,7	80,6	68,1	84,5	63,7
Typical	91,4	53,6	85,3	73,4	66,3	62,8	71,1	79,1	61,9	67,3	71,1
INN	91,2	47,5	83,6	48,4	88,2	80,9	63,9	91,3	53,7	67,0	72,3
ODIN	**91,8**	42,4	82,7	52,6	84,9	36,0	86,6	63,5	76,2	48,6	82,6
FT(0.1)	91,7	34,8	92,1	23,8	94,9	25,2	96,0	58,8	83,7	35,7	91,7
FT(0.1*)	91,7	26,2	94,2	16,0	96,4	24,8	96,0	59,8	83,5	31,7	92,5
FT(0.3)	91,7	28,1	93,3	16,1	96,8	**21,7**	96,5	**57,9**	85,5	30,9	93,0
FT(0.3*)	91,7	**16,7**	**96,5**	**8,1**	**98,1**	21,8	**96,6**	58,4	**85,6**	**26,3**	**94,2**

TABLE 2 : Comparaison de la méthode du *fine tuning* avec les méthodes classiques de détection d'OoD. Quatre configurations de composition de $\mathcal{S}_{\text{mix},\mathcal{P}}$ sont présentées : homogène/hétérogène, $r = 10\,\%/r = 30\,\%$. Bourrage de 700 % composé de C-N-T.

(R). Chaque combinaison testée inclut un nombre identique d'images de chaque type. Nous constatons que l'ajout de T à C-N améliore les performances, tandis que l'ajout de R a un effet marginal sur l'AUROC et peut dégrader le FPR. Par conséquent, nous construirons $\mathcal{P}$ à partir de C-T-N.

La table 2 compare notre méthode aux méthodes suivantes :

Mahala [6] Seuillage de la distance de Mahalanobis entre les centroïdes des classes et la variable latente.

Typical Seuillage bilatéral de l'estimation de $\log p(x|y)$ avant *fine tuning* du CVAE.

INN [8] Réseau inversible, même métrique que Typical.

ODIN [7] Seuillage des sorties softmax calculées après perturbation de l'entrée et application d'une coefficient de température aux logits.

Nous notons FT(r^*) notre méthode lorsque l'ensemble $\mathcal{S}_{\text{mix}}^{\text{out}}$ est homogène et FT(r) lorsqu'il est hétérogène, avec r représentant la proportion d'échantillons OoD dans $\mathcal{S}_{\text{mix}}$. Par exemple, FT(0.3) signifie que $\mathcal{S}_{\text{mix}}^{\text{out}}$ est hétérogène et que $r = 30\%$. Ainsi, dans FT(0.3), un batch est composé de 38 échantillons de chacun des quatre jeux OoD (LSUNr, LSUNc, SVHN, CIFAR-100), soit un total de 152 échantillons, ainsi que de 360 échantillons InD. En revanche, dans la configuration FT(0.3*), $\mathcal{S}_{\text{mix}}^{\text{out}}$ contient 152 échantillons issus d'un seul jeu OoD. L'hypothèse d'un batch hétérogène est plus réaliste que celle d'un batch homogène, mais elle n'avait pas été étudiée dans [9].

On relève que le passage d'un batch homogène à un batch hétérogène (par exemple de FT(0.1*) à FT(0.1)) entraîne une baisse des performances de détection d'OoD, avec une augmentation du FPR de 31.7% à 35.7%. Cette dégradation s'explique par la réduction du nombre d'échantillons disponibles par jeu OoD (un facteur 4 ici). Cependant, il est intéressant de constater que les performances de la configuration FT(0.3) restent proches de celles obtenues avec FT(0.1*), bien que le nombre d'échantillons par jeu OoD soit de 38 dans le premier cas contre 51 dans le second. La présence d'échantillons OoD issus de sources variées semble donc bénéfique, en exploitant une forme alternative de remplissage. Cette propriété est particulièrement intéressante pour des applications pratiques.

Nos résultats montrent des performances nettement supérieures aux méthodes concurrentes, avec une AUROC supérieure à 90 % en moyenne sur l'ensemble des jeux même dans le cas le plus défavorable (FT(0.1)). Nous tirons ainsi profit du traitement par lot, même lorsque ce dernier contient peu d'échantillons OoD et issus de sources diversifées.

5 Conclusion

Nous proposons une méthode combinant classification et détection d'OoD via la structuration de l'espace latent d'un classifieur génératif. Testée dans des conditions réalistes, elle offre des performances compétitives et surpasse les méthodes existantes, sous réserve de compatibilité avec un traitement par lot.

Références

[1] Xuefeng DU, Zhen FANG, Ilias DIAKONIKOLAS et Yixuan LI : How does unlabeled data provably help out-of-distribution detection ? *In ICLR*, 2024.

[2] Griffin FLOTO, Stefan KREMER et Mihai NICA : The tilted variational autoencoder : Improving out-of-distribution detection. *In ICLR*, 2023.

[3] Eduardo Dadalto Camara GOMES, Florence ALBERGE, Pierre DUHAMEL et Pablo PIANTANIDA : Igeood : An information geometry approach to out-of-distribution detection. *In ICLR 10*, 2022.

[4] Dan HENDRYCKS, Mantas MAZEIKA et Thomas G. DIETTERICH : Deep Anomaly Detection with Outlier Exposure. *In ICLR*, 2019.

[5] Julian KATZ-SAMUELS, Julia B NAKHLEH, Robert NOWAK et Yixuan LI : Training ood detectors in their natural habitats. *In ICML*, 2022.

[6] Kimin LEE, Kibok LEE, Honglak LEE et Jinwoo SHIN : A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *In NeurIPS*. 2018.

[7] Shiyu LIANG, Yixuan LI et R. SRIKANT : Enhancing the reliability of out-of-distribution image detection in neural networks. *In ICLR*, 2018.

[8] Radek MACKOWIAK, Lynton ARDIZZONE, Ullrich KÖTHE et C. ROTHER : Generative classifiers as a basis for trustworthy image classification. *In CVPR*, 2021.

[9] M. OSSONCE, P. DUHAMEL et F. ALBERGE : Adequate structuring of the latent space for easy classification and out-of-distribution detection. *In EUSIPCO*, 2024.

[10] K. SIMONYAN et A. ZISSERMAN : Very deep convolutional networks for large-scale image recognition. *In ICLR*, 2015.

[11] Kihyuk SOHN, Honglak LEE et Xinchen YAN : Learning structured output representation using deep conditional generative models. *In NeurIPS*, 2015.

[12] J. YANG, Kaiyang ZHOU, Yixuan LI et Ziwei LIU : Generalized out-of-distribution detection : A survey. *International Journal of Computer Vision*, 132(12), 2024.