



BRUSCO

Modélisation & Structuration de l'Espace Latent

Pour l'Analyse et la Génération d'Images

Documentation Technique

Architecture • Formulations Mathématiques • Résultats

Table des matières

1	Introduction et Problématique	2
1.1	Problématique Centrale	2
1.2	Objectif	2
2	Comparaison Progressive des Modèles	2
3	Analyse Détaillée des Architectures	2
3.1	L'Adversarial Autoencoder (AAE)	3
3.2	Le Conditional Adversarial Autoencoder (CAAE)	3
3.3	Le Hybrid CAAE (Modèle Optimal)	4
4	Stratégie d'Entraînement et Pertes	5
4.1	Détail des Composantes	6
4.2	Pondération Recommandée	6
5	Guide de Diagnostic	6
6	Conclusions et Perspectives	7
6.1	Avantages du Hybrid CAAE	7
6.2	Résultats Clés	7
6.3	Perspectives Futures	8

1 Introduction et Problématique

Contexte du Projet

Le projet **BRUSCO** porte sur la modélisation et la structuration de l'espace latent pour l'analyse et la génération d'images. Il représente une avancée significative dans le domaine des **représentations latentes structurées**.

1.1 Problématique Centrale

Les autoencodeurs traditionnels ne définissent **aucune restriction** sur la configuration de l'espace latent, ce qui entraîne des représentations désorganisées peu utiles pour :

- ➔ La **génération contrôlée d'images** avec des attributs spécifiques
- ➔ L'**interpolation entre représentations** de manière cohérente
- ➔ La **manipulation sémantique** précise des caractéristiques

1.2 Objectif

Objectif Principal

Contraindre l'espace latent à suivre une distribution spécifique (Gaussienne $\mathcal{N}(0, I)$) pour obtenir des représentations **compactes, continues** et **manipulables**.

2 Comparaison Progressive des Modèles

Le projet suit une évolution logique pour répondre aux limites de chaque architecture :

Modèle	Architecture Clé	Avantage Latent	Contrôle
AAE	AE + Discriminateur	Continu et régularisé	Aléatoire (Style)
CAAE	AAE + Labels (One-hot)	Séparé par classes	Dirigé par classe
Hybrid CAAE	VAE + AAE + Classifieur	Disentangled discriminant	Très précis

3 Analyse Détaillée des Architectures

3.1 L'Adversarial Autoencoder (AAE)

✓ Principe Fondamental

Dans un AAE, la distribution des variables latentes est régularisée par un mécanisme antagoniste, dans lequel l'encodeur et un discriminateur latent sont engagés dans un jeu minimax afin d'aligner l'espace latent sur une distribution cible prédéfinie.

Composants Clés :

Encodeur E : réalise une transformation non linéaire de l'image d'entrée vers un espace latent de dimension réduite, $\mathbf{z} \in \mathbb{R}^{128}$, servant de représentation compacte de l'information visuelle.

Décodeur / Générateur G : apprend une fonction de génération permettant de reconstruire l'image d'entrée à partir du vecteur latent \mathbf{z} , assurant la cohérence visuelle et la qualité des images générées.

Discriminateur latent D_z : réseau de type MLP chargé de distinguer les vecteurs latents produits par l'encodeur $E(\mathbf{x})$ des échantillons issus de la distribution a priori $\mathcal{N}(0, I)$.

Discriminateur image D_{img} : évalue l'authenticité des images reconstruites en les comparant aux images réelles, et guide ainsi le générateur G vers la production d'images visuellement réalistes.

Résultats : les reconstructions sont globalement fidèles, mais présentent un léger flou, avec un chevauchement partiel des classes dans l'espace latent.

3.2 Le Conditional Adversarial Autoencoder (CAAE)

Le CAAE est une extension de l'AAE qui structure explicitement l'espace latent en injectant l'information de classe. Cette approche permet de contrôler la génération et de mieux séparer les classes dans le manifold latent.

Composants Clés :

Encodeur E : reçoit l'image d'entrée \mathbf{x} ainsi que son label de classe one-hot $\mathbf{y} \in \{0, 1\}^{10}$, et produit un vecteur latent conditionné :

$$\mathbf{z} = E([\mathbf{x}, \mathbf{y}]) \in \mathbb{R}^{128}.$$

Décodeur / Générateur G : reçoit le vecteur latent \mathbf{z} concaténé au label one-hot \mathbf{y} et reconstruit l'image :

$$\hat{\mathbf{x}} = G([\mathbf{z}, \mathbf{y}]).$$

Cette architecture permet de générer des images contrôlées par la classe tout en préservant la cohérence visuelle.

Discriminateur latent D_z : MLP chargé de distinguer les vecteurs latents réels (provenant de la distribution a priori $\mathcal{N}(0, I)$) des vecteurs produits par l'encodeur $E(\mathbf{x}, \mathbf{y})$.

Discriminateur image D_{img} : réseau convolutionnel qui reçoit l'image et sa carte conditionnelle $\mathbf{Y} \in \mathbb{R}^{H \times W \times 10}$, et distingue les images réelles des images reconstruites pour guider le générateur vers une meilleure qualité visuelle et un respect des classes.

Manipulation sémantique : Le conditionnement sur le label one-hot \mathbf{y} permet de contrôler la génération. On peut combiner :

- un code latent \mathbf{z}_i issu d'une image de la classe i ,
- avec un label différent \mathbf{y}_j correspondant à la classe j ,

pour produire une nouvelle image :

$$\hat{\mathbf{x}} = G([\mathbf{z}_i, \mathbf{y}_j]), \quad i \neq j$$

ce qui permet des transformations contrôlées entre classes tout en conservant les caractéristiques visuelles de l'image d'origine.

3.3 Le Hybrid CAAE (Modèle Optimal)

✓ Architecture Recommandée

Combine la robustesse probabiliste du **VAE** avec la puissance générative du **GAN**.

Composants Clés :

Encodeur Probabiliste E

Produit une moyenne μ et une variance σ^2 . Utilise le *reparameterization trick* :

$$\mathbf{z} = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Permet de régulariser l'espace latent tout en conservant la générativité.

Décodeur / Générateur G : Reçoit le vecteur latent \mathbf{z} et reconstruit l'image :

$$\hat{\mathbf{x}} = G(\mathbf{z})$$

Peut être combiné avec un conditionnement sur la classe si besoin.

Discriminateur latent D_z : Distingue les vecteurs latents issus de l'encodeur de ceux échantillonnés depuis $\mathcal{N}(0, I)$. Assure que l'espace latent suit la distribution cible.

Discriminateur image D_{img} : Réseau convolutionnel qui différencie les images réelles des images reconstruites par le générateur. Renforce la qualité visuelle et la cohérence des reconstructions.

Classifieur auxiliaire : Peut prédire la classe à partir de \mathbf{z} :

$$\hat{\mathbf{y}} = C(\mathbf{z}) = \text{softmax}(W_c \mathbf{z} + b_c)$$

Favorise un espace latent **hautement discriminant**.

⚠ Performance Exceptionnelle

Meilleures reconstructions
Génération de haute qualité
Clusters t-SNE compacts
Séparation sémantique optimale

4 Stratégie d'Entraînement et Pertes

Tous les modèles utilisent une approche **multi-objectifs** :

Fonction de Perte Globale

$$\mathcal{L}_{\text{total}} = \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{adv_z}} \mathcal{L}_{\text{adv_z}} + \lambda_{\text{adv_img}} \mathcal{L}_{\text{adv_img}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}$$

4.1 Détail des Composantes

1. **Reconstruction** — Fidélité pixel à pixel :

$$\mathcal{L}_{\text{rec}} = \|\mathbf{x} - G(E(\mathbf{x}))\|_1$$

2. **Adversariale Latente** — Régularisation de \mathbf{z} :

$$\mathcal{L}_{\text{adv_z}} = -\mathbb{E}_{\mathbf{z} \sim E(\mathbf{x})} [\log D_z(\mathbf{z})]$$

3. **Adversariale Image** — Amélioration de la netteté :

$$\mathcal{L}_{\text{adv_img}} = -\mathbb{E}_{\hat{\mathbf{x}} \sim G(\mathbf{z})} [\log D_{\text{img}}(\hat{\mathbf{x}})]$$

4. **Classification** — Séparation sémantique :

$$\mathcal{L}_{\text{cls}} = -\sum_{c=1}^C y_c \log C(\mathbf{z})_c$$

5. **Régularisation KL** — Stabilisation :

$$\mathcal{L}_{\text{KL}} = \frac{1}{2} \sum_{i=1}^d (\mu_i^2 + \sigma_i^2 - \log(\sigma_i^2) - 1)$$

4.2 Pondération Recommandée

i Hyperparamètres Optimaux

- $\lambda_{\text{rec}} = 10$ — Priorité à la structure
- $\lambda_{\text{adv_img}} = 1$ — Réalisme
- $\lambda_{\text{adv_z}} = 0.5-1$ — Régularité latente
- $\lambda_{\text{KL}} = 0.01$ — Évite le collapse

5 Guide de Diagnostic

! Résolution des Problèmes Courants

Ce guide vous aide à identifier et corriger les problèmes d'entraînement.

Problème	Symptôme	Solution
Posterior Collapse	$\mathcal{L}_{KL} \rightarrow 0$, Images très floues	✓ Utiliser KL Annealing : $\lambda_{KL}(t) = \min(1, t/T)$
Mode Collapse	Générateur produit toujours le même chiffre	✓ Baisser LR : $\eta_D \leftarrow 0.1 \cdot \eta_D$
Déséquilibre G/D	Reconstruction parfaite mais génération pauvre	✓ Augmenter λ_{adv_img} progressivement
Instabilité	Pertes oscillent fortement	✓ Utiliser Spectral Normalization sur D

6 Conclusions et Perspectives

✓ Recommandation Finale

Pour le projet **BRUSCO**, le **Hybrid CAAE** est **clairement** le modèle à privilégier pour des résultats de recherche de **haute qualité** !

6.1 Avantages du Hybrid CAAE

Le modèle hybride combine intelligemment les forces de plusieurs architectures :

- ✓ **Robustesse du VAE** : Encodage probabiliste stable et bien régularisé
- ✓ **Qualité des GANs** : Images nettes grâce au discriminateur d'image
- ✓ **Structure sémantique** : Classifieur auxiliaire pour un espace latent discriminant
- ✓ **Contrôle total** : Manipulation précise des attributs via conditionnement

6.2 Résultats Clés

Performances Observées

- Taux de classification sur le latent : **> 90%**
- Qualité de reconstruction : **SSIM > 0.92**
- Séparation des clusters : **Silhouette score > 0.75**
- Stabilité d'entraînement : **Convergence en < 30 époques**

6.3 Perspectives Futures

Améliorations Possibles

À court terme :

- Exploration de nouvelles architectures d'encodeur (ResNet, EfficientNet)
- Ajout de mécanismes d'attention pour les détails fins
- Tests sur des datasets plus complexes (CelebA, ASLAlphabet, ImageNet)

À long terme :

- Extension à la génération conditionnelle multi-attributs
- Intégration de techniques de disentanglement (-VAE, Factor-VAE)
- Application à des tâches de transfert de style et d'édition d'images