arXiv:2507.14706v1 [cs.LG] 19 Jul 2025

FRAUD IS NOT JUST RARITY: A CAUSAL PROTOTYPE ATTENTION APPROACH TO REALISTIC SYNTHETIC OVERSAMPLING

A PREPRINT

Claudio Giusti
Department of Mathematics and Computer Science
University of Catania
Catania, CT 95125
claudio.giusti@studium.unict.it

Luca Guarnera
Department of Mathematics and Computer Science
University of Catania
Catania, CT 95125
luca.guarnera@unict.it

Mirko Casu
Department of Mathematics and Computer Science
University of Catania
Catania, CT 95125
mirko.casu@phd.unict.it

Sebastiano Battiato
Department of Mathematics and Computer Science
University of Catania
Catania, CT 95125
sebastiano.battiato@unict.it

July 22, 2025

ABSTRACT

Detecting fraudulent credit card transactions remains a significant challenge, due to the extreme class imbalance in real-world data and the often subtle patterns that separate fraud from legitimate activity. Existing research commonly attempts to address this by generating synthetic samples for the minority class using approaches such as GANs, VAEs, or hybrid generative models. However, these techniques, particularly when applied only to minority-class data, tend to result in overconfident classifiers and poor latent cluster separation, ultimately limiting real-world detection performance. In this study, we propose the Causal Prototype Attention Classifier (CPAC), an interpretable architecture that promotes class-aware clustering and improved latent space structure through prototype-based attention mechanisms and we will couple it with the encoder in a VAE-GAN allowing it to offer a better cluster separation moving beyond post-hoc sample augmentation. We compared CPAC-augmented models to traditional oversamplers, such as SMOTE, as well as to state-of-the-art generative models, both with and without CPAC-based latent classifiers. Our results show that classifier-guided latent shaping with CPAC delivers superior performance, achieving an F1-score of 93.14% percent and recall of 90.18%, along with improved latent cluster separation. Further ablation studies and visualizations provide deeper insight into the benefits and limitations of classifier-driven representation learning for fraud detection. The codebase for this work will be available at final submission.

1 Introduction

The escalation of cyber threats has made anomaly detection central in computer security. Organizations face increasingly sophisticated attacks, from targeted intrusions and APTs to advanced fraud schemes [1, 2, 3]. Malicious actions are rare and often hidden within massive volumes of legitimate activity, making minority-class detection one of the key challenges. Automated systems must identify new threats without excessive false positives, while ensuring interpretability for compliance [4, 5, 6, 7]. As a result, research has focused on data-driven and machine learning approaches that address class imbalance and adversarial adaptation [8]. Large-scale analyses [9] highlight that most cyber incidents are not headline-grabbing, but sector risks differ greatly. Fraud types keep evolving, including affiliate marketing abuse [10], and user response is shaped by liability and reimbursement differences across countries [11].

Detecting fraud, especially in financial systems such as e-commerce settings [12] where it is rare, remains a central challenge. Severe class imbalance limits standard classifiers, which often fail to capture rare events or sacrifice recall. Similar problems arise in domains like deepfake detection [13, 14, 15]. New biases such as 'impostor bias' [16] further complicate anomaly detection. Traditional machine learning models such as Logistic Regression [17], Random Forest [18], and XGBoost [19] are widely used, but their effectiveness drops with heavy imbalance, often requiring sophisticated sampling or cost-sensitive learning. Among oversampling strategies, two main families dominate:

- **SMOTE**-based methods [20]: Synthesize new minority samples via interpolation, balancing the training set. They are effective but can produce redundant or overly smooth data.
- **Generative models** (VAEs [21], GANs [22], DMs [23, 24]): These generate diverse samples from learned distributions, but often require significant tuning and are usually trained only on minority data, which can limit diversity and generalization.

Deep classifiers are effective for fraud detection [25], but are often black-boxes, making interpretability difficult in sensitive domains. Generative methods typically focus on augmenting the minority class without shaping the latent space for better decision-making. To address these limitations, we propose the Causal Prototype Attention Classifier (CPAC), a lightweight and interpretable architecture that uses prototype-based reasoning and feature attention for robust classification under imbalance. Coupling CPAC (or similar classifiers) with generative model encoders such as VAE-GAN enables latent space shaping that maximizes class separability and interpretability, outperforming SOTA oversampling methods in both clustering and detection metrics. We validate our approach on the Kaggle Credit Card Fraud Detection [26] dataset, benchmarking CPAC-augmented models against traditional classifiers, SMOTE, and generative oversamplers, as well as MLP-based latent classifiers. The main contributions of this work are as follows:

- We present the CPAC, an interpretable classifier that combines prototypes and attention for reliable fraud detection under extreme class imbalance.
- We introduce a classifier-guided latent shaping approach by attaching a classifier to the encoder of a VAE-GAN, enforcing class-aware clustering and improving downstream classification performance.
- We included the CPAC to the encoder of the VAE-GAN to improve the results of the generic classifiers by exploiting the inner qualities of the CPAC.
- We prove and discuss how training generative models only on fraud data proves ineffective despite the high performances that these might lead the classifiers, causing overconfidence and poor representation of the actual data.

This work will be structured as such: in Section 2 we present the current SOTA of the literature and all the works that pushed and inspired ours. In Section 3 we list and explain all the techniques and models that this work uses and presents. Later, in Section 4 we explain and analyze all the results we obtained. Section 5 explores the importance of each component in the proposed architecture by removing them and explaining why they are critical. In Section 6 we analyze and motivate why current SOTA might represent a liability for fraud detection, despite the good metrics and overall performances. Ultimately, Section 7 concludes the paper with some hints at some plausible future works.

2 Related Work

Research in fraud detection and anomaly identification has evolved along several key dimensions: data-level oversampling, deep generative modeling, and the development of interpretable or explainable classifiers [27]. Below, we review foundational and recent advances in these areas, with particular focus on techniques most relevant to the design of robust and interpretable fraud detection systems.

2.1 Data-Level Oversampling and Generative Models

Handling extreme class imbalance in fraud detection has long been addressed at the data-level. Early work introduced SMOTE [28], with refinements and surveys over the years [29]. More recently, deep generative models, including VAEs, GANs, and hybrid approaches have been applied to rebalance fraud datasets. For instance, Tang et al. [30] proposed combining GANs and VAEs to simulate realistic transaction flows for anomaly detection, showing improved detection of rare fraud behaviors. Complementary surveys summarized the landscape of GAN-based augmentation techniques for credit card fraud detection, highlighting a diversity of architectures and strategies [31]. Despite these advances, most generative methods focus on augmenting the minority class alone, which may lead to overconfident or overfit classifiers and limited cluster separation in latent space.

2.2 Prototypes, Attention, and Explainability

Beyond data augmentation, prototype-based networks provide intrinsic interpretability by comparing inputs to learned class exemplars. ProtoPNet [32] pioneered this for image classification, while subsequent work further quantified the visual or semantic attributes that drive similarity [33]. ProSeNet extended prototype reasoning to sequential data [34], but all such methods must guard against mismatches between prototypes and actual data features [35]. Attention mechanisms have likewise been adopted for per-feature weighting and have been proposed as explanation tools, though their reliability as explanations remains debated [36, 37]. Model-agnostic post-hoc methods such as LIME [38] and SHAP [39] also remain popular for explaining classifier decisions.

2.3 Recent Detection and Generative Oversampling Approaches

Recent years have witnessed a proliferation of generative oversampling strategies and detection methods for credit card fraud detection task, with most approaches focused on synthesizing minority class (fraud) data to address the severe class imbalance typical of this domain. Below, we summarize the methodological contributions and experimental setups of recent and representative works in this area. Rakhshaninejad et al. (2021) [40] propose an ensemble method that uses a weighted voting system algorithm to enforce and build more reliable classifiers for detecting frauds. Wang et al. (2022) [41] proposed the use of Unrolled Generative Adversarial Networks (Unrolled GAN) for the oversampling of fraudulent transactions. Their method, designed to overcome issues such as mode collapse in classical GANs, generates synthetic fraud samples to augment the minority class. The Unrolled GAN is explicitly trained only on the fraudulent samples, and the generated data is added to the original dataset before training downstream classifiers. Their experiments demonstrate that Unrolled GAN-based oversampling improves classification results over classical methods like SMOTE, highlighting the capacity of deep generative models to capture minority class distributions. Ding et al. (2023) [42] present a hybrid model that combines a Variational Autoencoder (VAE) with adversarial training (VAE-GAN) to generate synthetic fraud transactions. Their model learns the distribution of the minority class (fraud) and is trained solely on fraudulent examples, which are then used to augment the training set for downstream classifiers. Shi et al. (2025) [43] propose a class-imbalance-aware VAE with a transformer-based attention mechanism (Bal-VAE-Attention). Their model employs a loss function with class-aware weights to better learn from minority samples, and generates synthetic frauds for augmentation after training. Unlike earlier works, their results show that employing architectural or loss-based corrections can produce more robust synthetic samples and improved downstream detection rates. Ahmed et al. (2025) [44] propose a hybrid data sampling approach for credit card fraud detection by combining SMOTE with Edited Nearest Neighbors (ENN) [45]. Their method, evaluated on the Kaggle credit card dataset, demonstrates that this hybrid technique can significantly enhance the performance of ensemble models (including RF, KNN, and AdaBoost) and a voting ensemble. By first oversampling the minority class and then using ENN to remove noisy samples, they achieve high scores in accuracy, precision, recall, F1, and AUC, outperforming many traditional oversamplers and showing that careful data balancing is crucial for robust fraud detection. Overall, the prevailing trend in recent literature is to leverage generative models, typically trained only on minority class data, as advanced oversamplers.

2.4 Gaps: Classifier-Guided Latent Shaping

Despite the progress above, no prior work in credit card fraud detection (or related tabular noise detection) has employed a trainable classifier to explicitly shape or cluster the latent space during generative model training. Existing approaches either use auxiliary classifiers as side objectives, focus only on data-level augmentation, or employ prototype/attention mechanisms outside the generative learning loop. The limitation of modeling the minority class in isolation, potentially restricting their capacity to generate synthetic frauds that are truly discriminative with respect to the global data distribution; this motivates the approach we introduce in this work, integrating a classifier, more specifically a Causal Prototype Attention Classifier (CPAC), directly into the latent space of a VAE-GAN, constitutes an effective method for achieving class-aware, interpretable latent structure in the context of imbalanced anomaly detection.

3 Methodologies

As briefly stated before, this research will focus mainly on introducing the Causal Prototype Attention Classifier (CPAC) as a new way to detect frauds and compare its performances to standard models and to introduce a supervised way to shape the latent space to offer improved clusterization. Then we will introduce how we used a classifier to influence and help the VAE-GAN latent space, shape a better representation of the two classes. Ultimately, it will be shown how the CPAC classifier might be a better fit for a classification head due to its nature and its structure and its results will be compared to other classification heads.

3.1 Dataset and Preprocessing

We conduct our experiments on the publicly available Credit Card Fraud Detection dataset, hosted on Kaggle [26]. This dataset, released by Worldline and the Machine Learning Group of ULB, contains 284,807 anonymized transactions made by European cardholders in September 2013. Only 492 of these are labeled as fraudulent, resulting in an extreme class imbalance (approximately 0.17% fraud rate). Each transaction includes 30 features, where 28 have been transformed via principal component analysis (PCA) to protect confidentiality. The remaining two features are the Time and Amount. The target variable Class is binary, with 1 indicating fraud and 0 otherwise. To ensure each feature contributes uniformly to model training, we apply robust normalization to all input features. Specifically, for each feature x, we compute its median $\tilde{x}$ and interquartile range $\text{IQR}(x) = Q_3(x) - Q_1(x)$, and normalize using Equation 1:

$$x_{\text{norm}} = \frac{x - \tilde{x}}{\text{IQR}(x)} \tag{1}$$

This transformation centers features around zero and scales them while remaining robust to outliers a crucial property in fraud detection where anomalies naturally exhibit extreme values. For features where $\text{IQR}(x) = 0$, we default to a unit divisor to prevent numerical instability. After normalization, we split the dataset into 70% training and 30% validation sets as shown in Table 1. Stratified sampling will be used to maintain class distribution across splits.

Table 1: Dataset split by class (70% train, 30% validation).

Set	Normal (0)	Fraud (1)	Total
Train (70%)	199,020	344	199,364
Validation (30%)	85,295	148	85,443
Total	284,315	492	284,807

3.2 Oversampling Strategies

To address the extreme class imbalance in credit card fraud detection, we implemented and compared two oversampling strategies: SMOTE and a custom Variational Autoencoder–GAN (VAE–GAN) pipeline. Both techniques were used to synthetically augment the minority (fraudulent) class, and the generated samples were added only to the training set, leaving the evaluation set only with pure transactional data, emulating a real-life deploy scenario. We chose, 50, 75, 100 samples to generate for two main reasons: the first one is that using a higher number of samples could lead classifiers to overfit, especially if the number of generated frauds is higher than the original number in the dataset and the synthetic data is not of high quality. The second reason is that most of times, despite the increasing number of frauds in the training set, the models inevitably plateau as we will see in the next section.

3.2.1 SMOTE-Based Oversampling

The Synthetic Minority Over-sampling Technique (SMOTE) is a widely-used baseline for addressing class imbalance. Rather than learning the data distribution, SMOTE interpolates directly between existing minority-class samples. For two fraud instances $\mathbf{x}_i$ and $\mathbf{x}_j$, it generates a synthetic sample $\tilde{\mathbf{x}}$ along the line connecting them (Equation 2):

$$\tilde{\mathbf{x}} = \mathbf{x}_i + \alpha \cdot (\mathbf{x}_j - \mathbf{x}_i), \qquad \alpha \sim \mathcal{U}(0, 1), \tag{2}$$

where α is uniformly sampled. This process, repeated with each sample's k nearest neighbors, creates new minority points distributed across the feature space. In our experiments, we generated 50, 75, and 100 synthetic frauds using SMOTE and merged them into the training set. While SMOTE is simple and effective, it can produce overly smooth or redundant samples, especially when the minority class has complex or non-linear structure. As shown in Figure 1, SMOTE's interpolated samples often "connect the dots" between real fraud clusters, potentially resulting in synthetic points that are too similar to the originals. Despite remaining a strong baseline, SMOTE can be outperformed by generative models that better capture the underlying data distribution in highly imbalanced settings.

3.2.2 VAE–GAN Oversampling

The Variational Autoencoder-Generative Adversarial Network (VAE-GAN) has emerged as a powerful approach for generating synthetic data in imbalanced classification problems (its structure is visible in Figure 2). In accordance with prevailing practice in fraud detection, we employ the VAE-GAN exclusively as a minority-class oversampler: it is trained using only genuine fraud transactions, then used to synthesize new fraud-like samples that supplement the training set. The VAE-GAN consists of three neural modules: an encoder E_ϕ, a decoder D_θ, and a discriminator C_ψ.

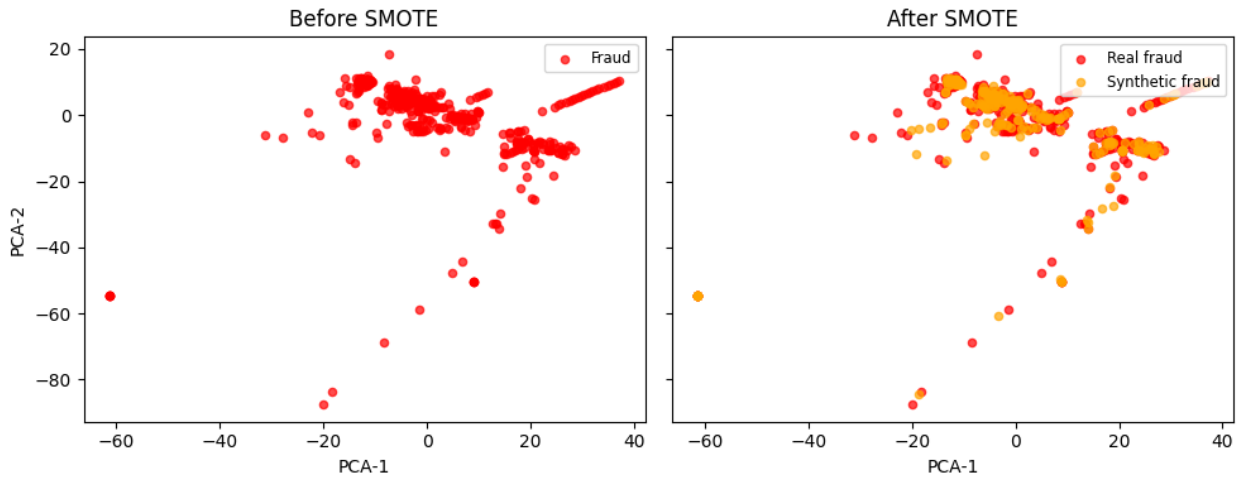


Figure 1: PCA plots comparing frauds distribution before and after SMOTE oversampling.

The encoder maps an input $\mathbf{x} \in \mathbb{R}^d$ through multiple hidden layers to the parameters of a multivariate Gaussian: mean vector $\boldsymbol{\mu}$ and log-variance $\log \boldsymbol{\sigma}^2$. Using the reparameterization trick, the latent code (Equation 3) is sampled as

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I). \tag{3}$$

The decoder D_θ reconstructs the input from $\mathbf{z}$, while the discriminator C_ψ distinguishes real from reconstructed samples. Training proceeds by minimizing a weighted sum of three losses:

- The **VAE loss** (Equation 4), which includes both a reconstruction error and a Kullback–Leibler divergence term:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\|\mathbf{x} - D_\theta(\mathbf{z})\|_2^2\right] + \beta \cdot \text{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z}))\,, \tag{4}$$

  where $p(\mathbf{z})$ is the standard normal prior and β controls the KL penalty.
- The **GAN loss** (Equation 5) for the discriminator, encouraging C_ψ to distinguish real fraud samples from reconstructions:

$$\mathcal{L}_{\text{GAN}} = -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log C_\psi(\mathbf{x})] - \mathbb{E}_{\hat{\mathbf{x}}\sim D_\theta(\mathbf{z})}[\log(1 - C_\psi(\hat{\mathbf{x}}))]\,. \tag{5}$$

- The **generator adversarial loss** (Equation 6), which pushes the decoder to generate samples that the discriminator cannot distinguish from real frauds:

$$\mathcal{L}_{\text{Adv}} = -\mathbb{E}_{\hat{\mathbf{x}}\sim D_\theta(\mathbf{z})}[\log C_\psi(\hat{\mathbf{x}})]\,. \tag{6}$$

During training, the encoder and decoder are optimized together to minimize both reconstruction and adversarial losses, while the discriminator is trained to distinguish real from generated samples. Early stopping and validation are used to ensure generalization. After training, the decoder generates new fraud samples by sampling from the learned latent distribution, augmenting the training set for subsequent classification. This VAE-GAN-based oversampling is widely used for its ability to produce more realistic and varied fraud examples than simpler interpolation techniques like SMOTE. In our experiments, we generated and merged 50, 75 and 100 synthetic fraud samples into the training data. However, as illustrated in Figure 3, such oversampling often concentrates the synthetic data in a narrow region of latent space, which can make downstream classifiers either overconfident or poorly calibrated, and limits cluster separation. This limitation provides the motivation for our classifier-guided methods.

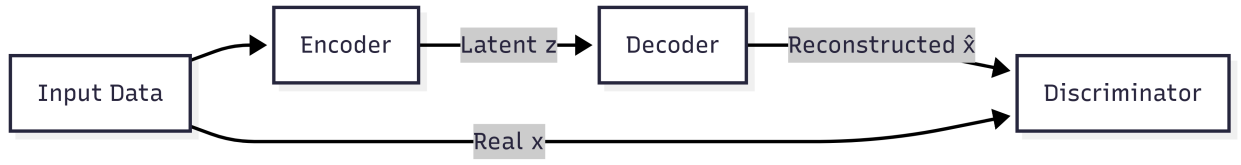


Figure 2: Diagram portraying the structure of the VAE-GAN.

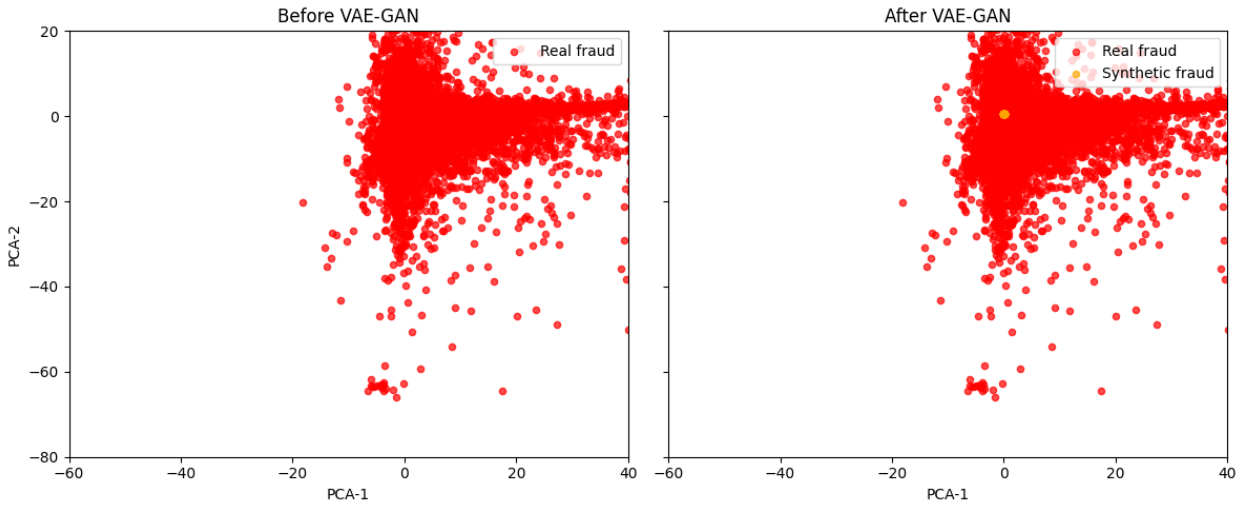


Figure 3: PCA plots comparing frauds distribution before and after VAE-GAN oversampling.

3.2.3 SMOTE vs VAE-GAN Oversampling

The fundamental difference between SMOTE and VAE–GAN oversampling lies in how synthetic minority-class examples are generated and distributed. SMOTE creates new samples by interpolating between real frauds and their nearest neighbors, filling gaps and uniformly spreading synthetic points within known clusters. While easy to use and computationally light, SMOTE may also generate borderline samples that overlap with the majority class. In contrast, VAE–GAN learns a latent representation and generates new frauds by sampling from a global latent prior, which often leads to denser clustering around the main fraud mode but can miss peripheral fraud patterns. VAE–GAN models require more careful design and tuning, but can create more realistic, nonlinear samples that better reflect complex feature relationships. Therefore, SMOTE is suited for quick and broad coverage of known minority regions, whereas VAE–GAN offers more expressive synthetic data at the cost of increased complexity and training effort. The choice ultimately depends on the application's requirements and available resources.

3.3 Baseline Classifiers

To benchmark the performance of our generative and prototype-based approaches, we evaluated three standard classifiers widely used in fraud detection: Logistic Regression, Random Forest, and XGBoost. These models provide strong baselines due to their interpretability, ensemble nature, and ability to handle imbalanced data with appropriate modifications. To assess the performance of all classifiers in this work, we report four standard metrics: precision, recall, F1-score, and AUC-ROC. Each metric captures a different aspect of performance for highly imbalanced classification problems such as fraud detection.

3.4 Causal Prototype Attention Classifier (CPAC)

The Causal Prototype Attention Classifier (CPAC) embeds interpretable, class-aware structure directly into a lightweight neural module. Given an input $\mathbf{x} \in \mathbb{R}^d$, CPAC learns two prototype vectors (Equation 7)

$$\mathbf{p}_0,\ \mathbf{p}_1\ \in\ \mathbb{R}^d \tag{7}$$

representing the centroids of the non-fraud and fraud classes. An attention network (Equation 8)

$$\mathbf{w}\ =\ \mathrm{Att}(\mathbf{x})\ =\ \sigma\big(W_2\,\mathrm{ReLU}(W_1\mathbf{x}+b_1)+b_2\big)\quad\in(0,1)^d \tag{8}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the latent input, W_1 and W_2 are learnable weight matrices, b_1 and b_2 are biases, ReLU is the rectified linear unit, and σ is the sigmoid function. The output $\mathbf{w} \in (0,1)^d$ is a per-feature attention mask. It produces a per-feature mask highlighting dimensions most predictive of fraud. A learnable scale $\alpha > 0$ adjusts sensitivity, and we compute weighted squared distances (Equation 9).

$$d_c(\mathbf{x}) \;=\; \alpha \sum_{i=1}^{d} w_i \left(x_i - p_{c,i}\right)^2 \quad , \; c \in \{0, 1\}. \tag{9}$$

where w_i is the i-th element of the attention vector $\mathbf{w}$, x_i is the i-th feature of the latent vector $\mathbf{x}$, $p_{c,i}$ is the i-th coordinate of the prototype for class c ($c = 0$ for non-fraud, $c = 1$ for fraud), α is a learnable scaling factor, and d is the latent dimensionality.

Interpreting negative distances as logits (Equation 10),

$$\boldsymbol{\ell}(\mathbf{x}) = \begin{bmatrix} -d_0(\mathbf{x}) \\ -d_1(\mathbf{x}) \end{bmatrix} \quad \Longrightarrow \quad \hat{y} \;=\; \mathrm{softmax}\big(\boldsymbol{\ell}(\mathbf{x})\big)_1, \tag{10}$$

where $d_0(\mathbf{x})$ and $d_1(\mathbf{x})$ are the weighted distances to the non-fraud and fraud prototypes, respectively, $\boldsymbol{\ell}(\mathbf{x})$ is the vector of negative distances used as logits, and $\mathrm{softmax}(\cdot)_1$ denotes the softmax probability for class 1 (fraud). This yields the predicted fraud probability $\hat{y} \in (0, 1)$.

The term "causal" here is used loosely to suggest that the attention weights may help highlight which latent features have a higher impact on the outcome. Its structure can be visualised in Figure 4.

3.4.1 Training Loss

To counter the extreme imbalance, we adopt the Focal Loss [46], which adds two hyperparameters (Equation 11), α_{FL} and γ to the standard binary cross-entropy:

$$\begin{aligned} \mathcal{L}_{\mathrm{FL}}(y, \hat{y}) = \; & -\; \alpha_{\mathrm{FL}}\,(1 - \hat{y})^{\gamma}\, y\, \log \hat{y} \\ & -\; (1 - \alpha_{\mathrm{FL}})\, \hat{y}^{\gamma}\, (1 - y)\, \log(1 - \hat{y}) \end{aligned} \tag{11}$$

Where:

- $\alpha_{\mathrm{FL}} \in [0, 1]$ balances the importance of the two classes. We set $\alpha_{\mathrm{FL}} = 0.95$ to give more weight to the minority (fraud) class.
- $\gamma \geq 0$ is the focusing parameter. When $\gamma = 0$, $\mathcal{L}_{\mathrm{FL}}$ reduces to ordinary cross-entropy. As γ grows, well-classified examples (where $\hat{y}$ is close to the true label) incur much smaller loss, forcing the model to concentrate on harder, often minority-class cases. We performed a grid search for the most optimal values for α_{FL} and γ and we found out that the best results are obtained with 0.95 and 2.0 respectively.

By tuning α_{FL} and γ, Focal Loss both re-weights the underrepresented class and focuses learning on its most difficult examples, which is crucial in fraud detection. Model selection uses a composite score (Equation 12):

$$S \;=\; 0.50\,\mathrm{Precision} \;+\; 0.50\,\mathrm{Recall} \tag{12}$$

on the validation set. We checkpoint the CPAC weights whenever S improves, and invoke early-stopping after p epochs without gain.

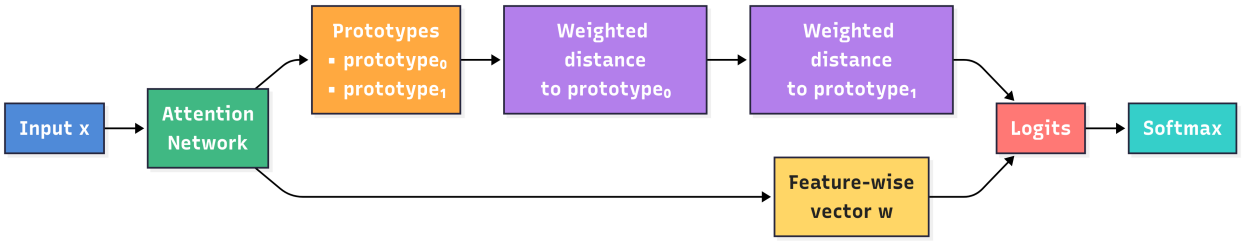


Figure 4: Architecture of the CPAC model. Each input is compared to class prototypes using an attention-weighted distance, followed by softmax scoring.

3.5 Adaptive Threshold Selection via Differentiable Agent

To improve classification under class imbalance, we introduce a differentiable agent that adaptively learns the optimal classification threshold τ^* for maximizing the F1 score. Instead of using the default $\tau = 0.5$, we parameterize the threshold as a scalar θ, mapping it to $\tau = \sigma(\theta)$ (where σ is the sigmoid function) so that τ lies in (0,1). For each sample i, with predicted probability p_i and ground truth $y_i \in \{0, 1\}$, the agent approximates a hard threshold with a smooth sigmoid function (Equation 13):

$$\hat{y}_i = \sigma\big(\beta(p_i - \tau)\big), \tag{13}$$

where $\hat{y}_i$ is the soft binary prediction for sample i, p_i is the predicted probability, τ is the learned threshold, and β controls the sharpness of the sigmoid (making the function step-like). The loss for optimizing the threshold is given by Equation 14:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{y}_i - y_i\right)^2, \tag{14}$$

where N is the batch size, $\hat{y}_i$ is the soft prediction, and y_i is the true binary label. By minimizing this loss via gradient descent and monitoring validation F1, the agent effectively learns an optimal threshold suited to the classifier's output probabilities. This method consistently yielded improved F1 scores, often with a learned threshold above 0.7, especially when paired with classifiers such as CPAC or those using VAE–GAN encoders.

3.6 VAE-GAN with Classification Heads

Most state-of-the-art techniques [41, 42, 43], train the generative model on only the fraud (minority) class. This classical approach has a crucial limitation: it fails to expose the model to the characteristics of non-fraudulent (majority) data, resulting in a generator that can merely interpolate among known frauds, often producing synthetic samples that are near-duplicates, lacking true discriminative power. The latent space learned in such a setup is inevitably narrow and uninformative about what actually distinguishes fraud from normality. In contrast, our approach is fundamentally different. We jointly train the VAE-GAN with a classification head on the full dataset, even though the generative (VAE-GAN) component is optimized only on the minority class. The critical distinction is that the encoder, which is shared between the generator and the classifier head, receives supervised feedback from both classes via the classification objective. This means that the latent space, as learned by the encoder, encodes information about the entire data distribution,both fraud and non-fraud. This approach fundamentally differs from simply training on the minority class. The VAE-GAN and classifier pipeline is fully aware of both classes, since the encoder's parameters are influenced by the generative loss applied to fraud samples and the classification loss computed over the entire dataset. Focusing the generative loss on the minority class is a deliberate choice that enables the decoder to specialize in high-fidelity, targeted synthesis of frauds. At the same time, the classifier head continuously regularizes the encoder, ensuring that it organizes the latent space to distinguish between both fraud and non-fraud classes. As a result, the latent space does not lose information about the normal class; rather, it is shaped to maximize class discrimination. The encoder integrates the objectives of both components, while the decoder utilizes this enriched representation to generate synthetic frauds that are not only realistic but also truly distinct within the broader data context. Thus, although the generative focus is on the minority class, the overall supervision ensures that the synthetic samples are robust, generalizable, and valuable for distinguishing between classes.

3.6.1 Classifier Heads (MLP Variants)

We consider three multilayer perceptron (MLP) heads, as shown in Figure 5, of increasing complexity, each implementing a function $h_\theta : \mathbb{R}^d \to [0, 1]$, with $d = 2$ in our experiments:

1. **MLPHead1**: One hidden layer with 32 units (ReLU), output through a sigmoid (Equation 15):
$$h_1(z) = \sigma\left(W_2 \,\mathrm{ReLU}(W_1 z + b_1) + b_2\right) \tag{15}$$
where $W_1 \in \mathbb{R}^{32\times 2}$, $b_1 \in \mathbb{R}^{32}$, $W_2 \in \mathbb{R}^{1\times 32}$, $b_2 \in \mathbb{R}$.
2. **MLPHead2**: One hidden layer with 64 units, batch normalization, dropout ($p = 0.2$), ReLU and sigmoid (Equation 16):
$$h_2(z) = \sigma\left(W_2 \,\mathrm{Dropout}\big(\mathrm{ReLU}(\mathrm{BN}(W_1 z + b_1))\big) + b_2\right) \tag{16}$$
with $W_1 \in \mathbb{R}^{64\times 2}$, $b_1 \in \mathbb{R}^{64}$, $W_2 \in \mathbb{R}^{1\times 64}$, $b_2 \in \mathbb{R}$.

3. **MLPHead3**: Two hidden layers with 128 and 64 units (ReLU), output through a sigmoid (Equation 17):

$$h_3(z) = \sigma\left(W_3\,\mathrm{ReLU}(W_2\,\mathrm{ReLU}(W_1 z + b_1) + b_2) + b_3\right) \tag{17}$$

where $W_1 \in \mathbb{R}^{128\times 2}$, $b_1 \in \mathbb{R}^{128}$, $W_2 \in \mathbb{R}^{64\times 128}$, $b_2 \in \mathbb{R}^{64}$, $W_3 \in \mathbb{R}^{1\times 64}$, $b_3 \in \mathbb{R}$.

These structures will all be tested and evaluated as potential heads for our encoder.

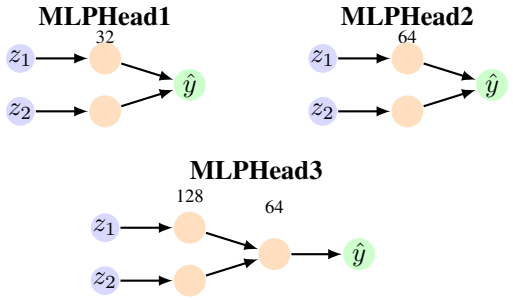


Figure 5: Architectures of the three MLP heads used for classification on the 2-dimensional latent space. Hidden units are indicated above each hidden layer.

3.6.2 Joint Training Procedure

The model is trained in two coordinated phases at each epoch:

1. **VAE-GAN update**: For each mini-batch (from all classes), the encoder maps x to $z, \mu, \log\sigma^2$, and the decoder reconstructs x_{rec}. The discriminator receives both x and x_{rec} and tries to distinguish real from generated data. The VAE-GAN is trained with the following loss (Equation 18):

$$\mathcal{L}_{\text{VAE-GAN}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{KL}} + \mathcal{L}_{\text{GAN}} \tag{18}$$

where $\mathcal{L}_{\text{recon}}$ is MSE between x and x_{rec}, $\mathcal{L}_{\text{KL}}$ is the KL divergence, and $\mathcal{L}_{\text{GAN}}$ is the adversarial loss for the generator decoder.

2. **Classifier head update**: Using the same mini-batch, the encoder's mean μ is passed to the classifier head h_θ to predict the label y. The head is trained with binary cross-entropy (Equation 19):

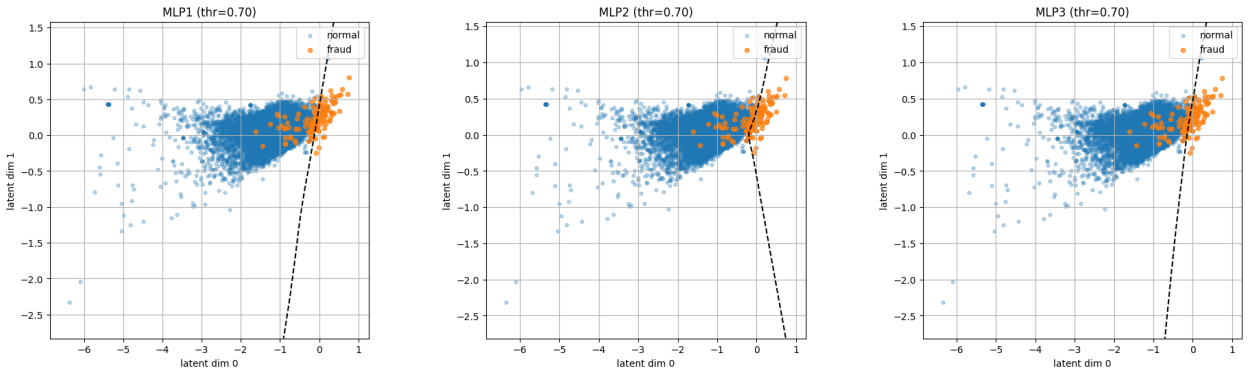
$$\mathcal{L}_{\text{clf}}(y, \hat{y}) = -y \log \hat{y} - (1-y)\log(1-\hat{y}), \quad \hat{y} = h_\theta(\mu) \tag{19}$$

The gradient of $\mathcal{L}_{\text{clf}}$ flows back not only to the classifier head parameters θ, but also to the encoder parameters. As a result, the encoder is explicitly encouraged to organize the latent means μ so that different classes become more separable for the downstream classifier. This joint training guides the encoder to learn a latent representation where fraud and non-fraud samples are more easily discriminated.

Despite all three experiments with all the MLPs (as shown in Figures 6a, 6b, 6c) clearly suggest at a slight cluster separation, the overlap between the two classes is still significant. The first and third MLPs produces similar results despite the different structures while the second MLP is the only one leaning into a polynomial decision boundary that could help separating the two clusters. These results prove that we need a more powerful network capable of adapting into the encoder logic and working on a non-linear and more complex space. Due to the big overlap between the two classes, using this as a way to generate new frauds to train models only causes the latter to perform badly.

3.7 VAE-GAN with CPAC Head

Similarly to how we did with the MLPs, we tested how the CPAC behaves as a classification head paired with the encoder. With the MLPs, there was a slight tendency to inner cluster separation in the encoder but the overlap was still significant. CPAC's structure, is very akin with the encoder, because its prototypes might offer an anchor point where the centroid of each cluster might find its position pushing for a better separation. The training is similar to what has already been done with the MLPs.



(a) MLPHead1 (b) MLPHead2 (c) MLPHead3

Figure 6: PCA plots showing cluster separation in the latent space for different MLP heads.

3.7.1 Joint Training Procedure

At each training epoch, we alternate between:

1. **VAE-GAN training:** For each batch, the encoder E_ϕ encodes x to $(z, \mu, \log \sigma^2)$, the decoder G_ψ reconstructs $x_{\text{rec}} = G_\psi(z)$, and the discriminator D_ω distinguishes x from x_{rec}. The standard VAE-GAN loss (Equation 20) is:
$$\mathcal{L}_{\text{VAE-GAN}} = \mathcal{L}_{\text{recon}}(x, x_{\text{rec}}) + \mathcal{L}_{\text{KL}}(\mu, \log \sigma^2) + \mathcal{L}_{\text{GAN}}(D_\omega(x_{\text{rec}}), 1) \tag{20}$$
where $\mathcal{L}_{\text{recon}}$ is mean squared error (MSE), $\mathcal{L}_{\text{KL}}$ is the KL divergence, and $\mathcal{L}_{\text{GAN}}$ is the adversarial loss for the generator.
2. **CPAC update:** In the same batch, the encoder's mean μ is passed to the CPAC head, which computes distances (Equation 21) to two learnable prototypes $(\mathbf{p}_0, \mathbf{p}_1)$ via feature-wise attention weights $\mathbf{w}$:
$$d_c = \alpha \sum_{j=1}^{d} w_j \left(\mu_j - (\mathbf{p}_c)_j\right)^2, \qquad c \in \{0, 1\} \tag{21}$$
where $w_j \in (0, 1)$ are attention weights (from a neural branch), and α is a learnable scaling parameter. Fraud probability is given by Equation 22:
$$\hat{y} = \text{Softmax}(-d_0, -d_1) \tag{22}$$
The CPAC is trained to minimize the binary cross entropy (BCE) loss. To further regularize learning, two penalties are added:
   - **Scale penalty:** encourages the attention scaling parameter to stay bounded (Equation 23):
$$\mathcal{L}_{\text{scale}} = \lambda_{\text{scale}} \cdot \|\alpha\|^2 \tag{23}$$
   - **Prototype anchoring:** aligns each prototype to the centroid of its class in latent space, encouraging the encoder to cluster samples around the correct prototype (Equation 24):
$$\mathcal{L}_{\text{anchor}} = \lambda_{\text{anchor}} \left(\|\mathbf{p}_0 - \bar{\mu}_0\|^2 + \|\mathbf{p}_1 - \bar{\mu}_1\|^2\right) \tag{24}$$
   where $\bar{\mu}_0$, $\bar{\mu}_1$ are the means of latent vectors in the current batch with $y = 0$ and $y = 1$ respectively.

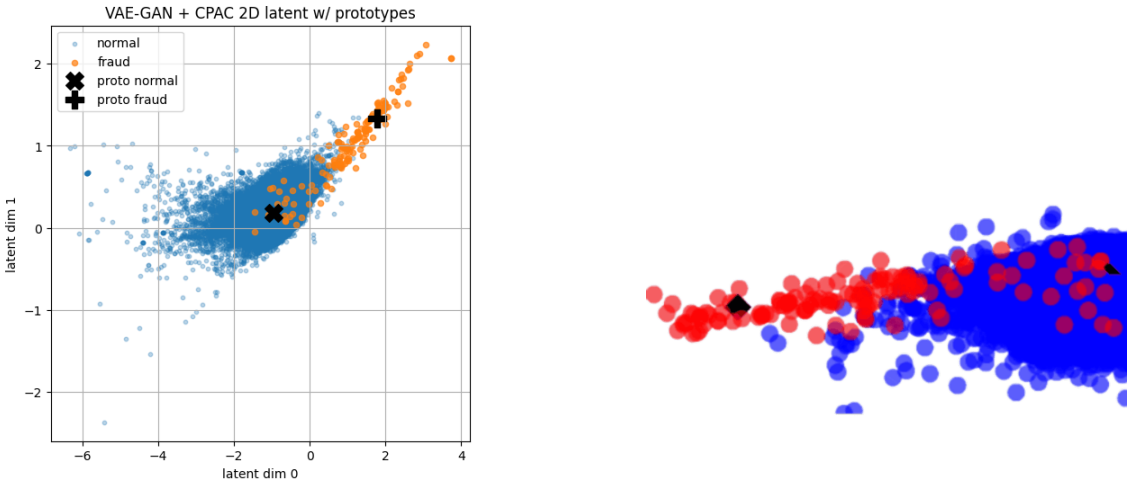
The total loss (Equation 25) for CPAC becomes:
$$\mathcal{L}_{\text{CPAC-total}} = \mathcal{L}_{\text{clf}} + \mathcal{L}_{\text{scale}} + \mathcal{L}_{\text{anchor}} \tag{25}$$
and the overall optimization alternates VAE-GAN and CPAC updates. Just like we did for the Focal Loss we performed a grid-search and found the best λ_{scale} and λ_{anchor} to be respectively 0.001 and 0.01. When using CPAC as a classification head within our VAE-GAN architecture, we switch to binary cross entropy (BCE) loss. Just like the MLPs, the gradient of the CPAC loss gets backpropagated to both the CPAC parameters and also to the encoder parameter. After each epoch, we evaluate the CPAC on a held-out validation set. Early stopping is applied based on recall, conditional on maintaining a minimum precision, as in other head experiments.

4 Experimental Results

Given the proposed methodologies, we dwelve into all the experiments with the models and methods listed in Section 3. First we test the CPAC against the other three classifiers and the results obtained by other works. Then, the results obtained with the VAE-GAN with a classification head will be analysed, explored and compared to other oversampling techniques.

4.1 Preliminary Results

Unlike MLP heads, the CPAC's architecture and losses (especially prototype anchoring) push the encoder to organize latent means μ into two tight, well-separated clusters, each surrounding its prototype. The attention weights $\mathbf{w}$ add interpretability, highlighting which latent features are most influential for distinguishing frauds. This process results in a latent space that is both discriminative and interpretable. As we can see on Figure 7a the CPAC was able to improve separation creating two distinct clusters with each prototype being anchored to its cluster centroid, essentially acting like one. There are a few misclassification which is common for an imbalanced task and a very little overlap(visible in Figure 7b) which is caused by borderline transactions. This is the first step to a more aware encoder that knows what actually constitutes a fraud.



(a) 2D PCA: Encoder with CPAC head.

(b) 3D cluster overlap.

Figure 7: Latent space visualizations for the Encoder with CPAC head: (a) PCA, (b) 3D overlap.

4.2 Oversampling results and experiments

Now, we will cover the results obtained with the two state-of-the-art oversampling strategies [20, 21] applied to Logistic Regression, Random Forest, XGBoost and CPAC. First they will be tested without oversampling, to asses how oversampling can aid overall performances; then they will be tested with SMOTE oversampling and VAE-GAN.

4.2.1 No Oversampling

We tested the models on the non augmented dataset. Without augmentation, as we can see in Table 2, the Logistic Regression achieves a relatively high Precision but low Recall, meaning that oversampling could aid its training and boost its performances. Random Forest achieves the highest precision with a strong recall metric, suggesting that its performances could only benefit with oversampling. XGBoost seems the most stable, achieves the highest recall and as well as the Random Forest, its performances will only improve with an enriched dataset. The CPAC, despite being more stable than the Logistic Regression, still struggles to reach metrics comparable to the previous two, clearly urging for a richer dataset to help its prototypes find the right anchor point in its representation.

4.2.2 SMOTE Oversampling Performances

SMOTE oversampling with 50, 75, and 100 synthetic fraud samples led to noticeable performance improvements for all classifiers as noticeable in Table 3, compared to training on the original imbalanced data. Random Forest consistently

Table 2: Benchmark results on the original (non-augmented) dataset.

Model	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
Logistic Regression	92.96	79.05	84.66	95.53
Random Forest	**98.87**	80.07	87.22	92.75
XGBoost	96.70	**88.51**	**92.21**	**96.80**
CPAC	87.20	73.65	79.85	95.80

achieved the highest precision, while XGBoost provided the best balance between recall and F1-score, especially at higher levels of oversampling. This confirms XGBoost's robustness to class imbalance with targeted augmentation. Logistic Regression improved in precision but remained limited in recall and F1. The CPAC classifier showed strong AUC-ROC but was outperformed by Random Forest and XGBoost on recall and F1. Overall, these results highlight that moderate SMOTE oversampling benefits all standard classifiers, with ensemble methods maintaining a clear advantage in fraud detection.

Table 3: Benchmark results using SMOTE oversampling with 50, 75, and 100 synthetic fraud samples.

# Samples	Model	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
50	Logistic Regression	92.51	82.76	87.00	95.47
50	Random Forest	**98.25**	87.84	**92.41**	93.40
50	XGBoost	95.10	**87.49**	90.95	94.07
50	CPAC	85.37	70.95	77.49	**95.70**
75	Logistic Regression	91.42	82.76	86.58	95.66
75	Random Forest	**97.48**	88.51	92.53	93.04
75	XGBoost	95.22	**88.51**	**91.59**	**95.81**
75	CPAC	86.36	77.03	81.43	94.30
100	Logistic Regression	92.45	82.42	86.76	95.54
100	Random Forest	**98.25**	87.84	92.41	93.38
100	XGBoost	97.06	**88.17**	**92.15**	94.58
100	CPAC	85.16	73.65	78.99	**97.30**

4.2.3 VAE-GAN Oversampling Performances

VAE–GAN-based oversampling, using 50, 75, and 100 synthetic fraud samples, further enhanced classifier performance as shown in Table 4. XGBoost achieved the highest recall and F1-score across most settings, leveraging the higher-quality synthetic data. Random Forest remained the most precise, while CPAC delivered robust recall and AUC-ROC, particularly at 75 synthetic samples. Logistic Regression continued to underperform compared to ensemble methods. These results demonstrate that generative oversampling with VAE–GAN yields more effective and discriminative synthetic data than SMOTE, especially for highly imbalanced fraud detection.

Table 4: Benchmark results using a VAE–GAN oversampler with 50, 75, and 100 synthetic fraud samples.

# Samples	Model	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
50	Logistic Regression	92.49	80.73	85.66	93.92
50	Random Forest	**98.12**	85.13	90.61	93.08
50	XGBoost	96.70	**88.51**	**92.21**	**96.72**
50	CPAC	90.49	87.48	88.93	95.46
75	Logistic Regression	93.12	77.70	83.72	94.30
75	Random Forest	**98.95**	82.43	89.01	92.71
75	XGBoost	97.46	88.17	**92.31**	**96.58**
75	CPAC	91.65	**88.84**	90.19	95.56
100	Logistic Regression	93.05	77.36	83.45	93.70
100	Random Forest	**98.95**	82.43	89.01	92.70
100	XGBoost	96.70	**88.51**	**92.21**	96.20
100	CPAC	89.11	86.47	87.74	**96.73**

4.2.4 SMOTE vs VAE-GAN Results Observations

Comparative analysis shows that VAE–GAN-generated synthetic frauds provide greater improvements in recall and F1-score than SMOTE, particularly for XGBoost and CPAC. SMOTE easily fills gaps within clusters but can generate less realistic points at class boundaries. VAE–GAN, while more complex, generates realistic samples that support better model generalization. CPAC, when paired with VAE–GAN, shows pronounced gains in recall and AUC-ROC, especially with more synthetic samples. However, excessive oversampling can lead to plateauing or overfitting, highlighting the need to find an optimal level of augmentation.

4.3 VAE-GAN with CPAC Head as Oversampler

Observing Table 5 we notice how using the VAE–GAN+CPAC model to generate synthetic fraud data for downstream classifiers, Logistic Regression remained the least robust, while overall metrics were slightly lower than previous oversampling strategies. This is a positive indication that the generated synthetic samples are more realistic and less likely to cause overfitting. Standard models showed improved robustness, with performance gains plateauing as the number of synthetic samples increased. Random Forest achieved the highest precision, but XGBoost provided more balanced and reliable performance across all metrics, making it the preferred model to pair with the VAE–GAN+CPAC oversampler for practical fraud detection.

Table 5: Benchmark results using a VAE-GAN+CPAC oversampler with 50, 75, and 100 synthetic fraud samples.

# Samples	Model	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
50	Logistic Regression	92.20	79.38	84.64	93.45
50	Random Forest	**98.29**	88.51	**92.85**	93.98
50	XGBoost	96.38	**90.18**	92.48	**96.82**
75	Logistic Regression	91.47	79.72	84.62	93.29
75	Random Forest	**98.27**	88.17	**92.63**	93.94
75	XGBoost	96.38	**90.17**	92.48	**97.07**
100	Logistic Regression	92.27	79.72	84.90	92.60
100	Random Forest	**98.29**	**88.51**	**92.85**	93.60
100	XGBoost	96.70	**88.51**	92.21	**96.37**

4.3.1 Applying Oversampling before VAE-GAN+CPAC Training

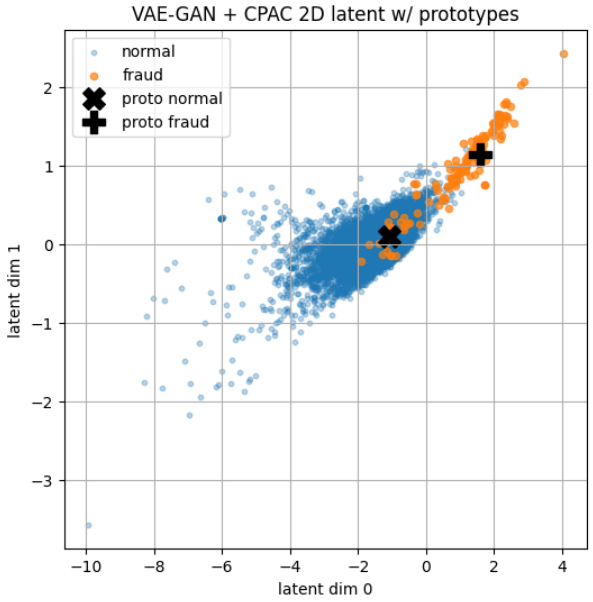
To further improve latent space separation in the VAE-GAN+CPAC framework, we explored the effect of applying a slight oversampling to the fraud class before pipeline training. Specifically, we employed SMOTE, as it produces a broader spread of synthetic fraud samples across the cluster, offering a better foundation for the encoder to learn generalizable boundaries. While it might seem counter intuitive given our claims about minority only oversampling, in this case we need a slight "accentuation" of the frauds samples defined in the overlap in order to allow the model to detect them better and further reduce the overlap. Pre-training augmentation with VAE-GAN tended to generate samples in a narrower region of the minority class, providing less diversity and thus limited improvement in latent separation. Guided by previous experiments, we selected 75 SMOTE-generated samples as the optimal amount, since further augmentation did not yield additional gains (see Figure 9). This approach visibly reduced cluster overlap in the latent space (Figures 8a, 8b) and improved overall model performance. Thus, SMOTE-based pre-training oversampling proved more effective than VAE-GAN augmentation in this context, with 75 synthetic samples offering the best balance between diversity and separation.

4.3.2 Results Obtained with Pre-Training Oversampled VAE-GAN+CPAC

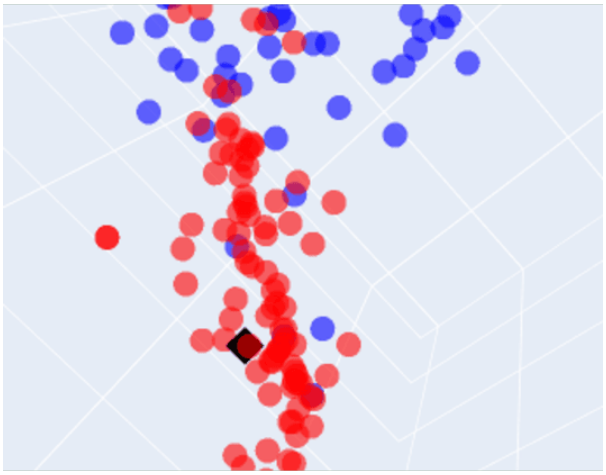
In these experiments, we assessed the effect of applying SMOTE-based oversampling to the minority class before training the VAE-GAN+CPAC pipeline. As shown in Table 6, the performance improvements over the standard VAE-GAN+CPAC pipeline (Table 5) are modest, with only slight gains in F1-score and AUC-ROC, particularly for XGBoost. This suggests that VAE-GAN+CPAC is already effective at modelling and separating the latent space for fraud detection without pre-training augmentation. Nonetheless, pre-training oversampling provides a qualitative benefit by further clarifying class boundaries in the latent space, as evident in our cluster visualizations. Overall, this configuration offers a practical advantage, producing clearer latent representations that support downstream classifiers and reinforcing the robustness of the VAE-GAN+CPAC approach.

Table 6: Benchmark results using a pre-training oversampled VAE–GAN+CPAC oversampler with 50, 75, and 100 synthetic fraud samples.

# Samples	Model	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
50	Logistic Regression	91.55	80.06	84.88	93.33
50	Random Forest	**98.27**	88.17	**92.63**	93.61
50	XGBoost	96.38	**90.17**	92.48	**96.91**
75	Logistic Regression	91.87	79.72	84.76	93.17
75	Random Forest	**98.29**	88.51	**92.85**	93.95
75	XGBoost	95.98	**88.85**	92.11	**97.28**
100	Logistic Regression	91.63	80.39	85.13	93.11
100	Random Forest	**98.69**	88.17	92.79	93.60
100	XGBoost	96.38	**90.18**	**93.14**	**96.88**



(a) 2D PCA: CPAC head, SMOTE pre-oversampling (75 samples).



(b) 3D cluster overlap: SMOTE pre-oversampling (75 samples).

Figure 8: Latent space visualizations with CPAC head and SMOTE pre-oversampling. (a) PCA, (b) 3D overlap.

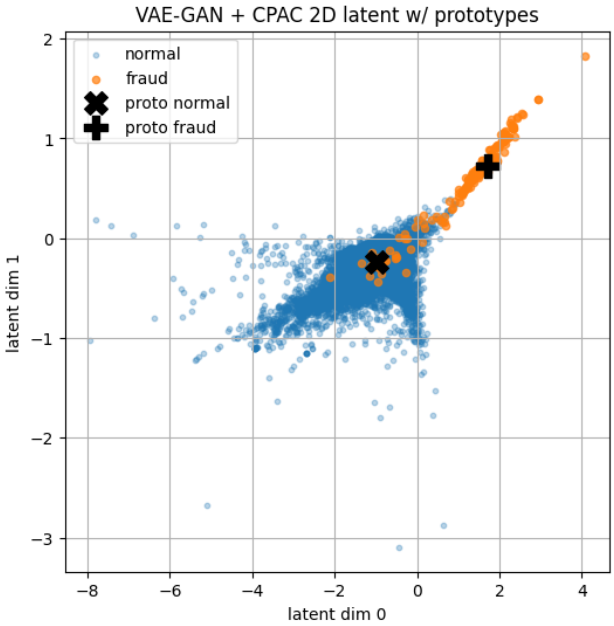


Figure 9: 2D PCA plot of the cluster representation of the Encoder with CPAC head trained with SMOTE pre-oversampling over 100 samples.

4.4 Comparison with State of the Art Approaches

To evaluate the effectiveness of our approach, we benchmarked the best model obtained in this work (XGBoost) trained with pre-training oversampled VAE-GAN+CPAC data against recent state-of-the-art methods [42, 43, 44]. We conducted these experiments on these works instead of others listed in Section 2 because they are more recent. Since the code for the selected works is not available we reproduced their classification settings at the best of our abilities and accordingly to the information given; Ding et al. [42] uses an XGBoost as a baseline classifier just like us, so the

reported results of our models reflects theirs with our generated dataset proving that our method greatly improves the performances. Shi et al. [43] uses a multi-head attention classifier in their generative pipeline to directly classify the samples reporting perfect metrics, so, we took just the classifier to test with our generated data. Ahmed et al. [44] uses a voting ensemble classification method that comprises a Random Forest Classifier, AdaBoost Classifier and KNN, similary reporting perfect metrics. The Table 7 reports the evaluation results of each method compared to ours with their relative setup: Shi et al. uses a 75/25 split for the dataset and balances the training set so that the minority class reaches the same number of the majority class. Ahmed et al. uses a 80/20 split and balances the training set as well. We ran both tests and compared the results with each method and our XGBoost outperforms in every metric. Table 8 reports the results obtained with our setup (70/30 split and only 100 generated samples in the training set) and even in this instance our model outperforms the other proposed methodologies. The main difference between our proposed method of training and the related works's is that we do not balance the training set because we believe that the models can only benefit from learning an extreme imbalanced distribution during training, adapting it to real world applications even appropriately.

Table 7: Benchmark XGBoost evaluation performances trained with pre-training oversampled VAE-GAN+CPAC data, compared to selected recent works with their setup and splits.

Work	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
Shi et al. [43] (2025)	79.50	78.86	79.18	97.56
Ours	**97.58**	**90.24**	**93.60**	**97.83**
Ahmed et al. [44] (2025)	93.75	76.53	84.27	97.37
Ours	**95.44**	**90.81**	**93.00**	**98.10**

Table 8: Benchmark XGBoost evaluation performances trained with pre-training oversampled VAE-GAN+CPAC data at 100 samples, compared to selected recent works with our setup.

Work	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
Shi et al. [43] (2025)	78.76	77.70	78.23	**97.56**
Ours	**96.38**	**90.18**	**93.14**	96.88
Ahmed et al. [44] (2025)	94.59	70.95	81.08	96.29
Ours	**96.38**	**90.18**	**93.14**	**96.88**

5 Ablation Study

In order to establish the relevance of each component in our method, we systematically analyze the contribution of each key component in our VAE-GAN+CPAC architecture. By selectively removing or disabling certain elements, we demonstrate that every part of the model is essential for achieving discriminative and robust latent representations, particularly in the context of extreme class imbalance.

5.1 Effect of Removing the CPAC Head

To evaluate the impact of the CPAC head, we trained the VAE-GAN with the same settings as our main pipeline, but without the CPAC head. In this configuration, the encoder is updated only via generative objectives, with no explicit supervision guiding the latent space. Our results show that (as shown in Figure 10), without the classifier, the latent representations of fraud and normal transactions remain heavily overlapped, making downstream classification significantly less accurate. This experiment confirms that the CPAC supervision is crucial for inducing clear separation between classes and for shaping the latent space in a way that supports reliable detection.

5.2 Effect of Removing the Attention Mechanism from CPAC

The ablation in Figure 11 illustrates the effect of disabling the attention mechanism in the Causal Prototype Attention Classifier (CPAC), reducing it to a pure prototype-based classifier. Without feature-wise attention, each latent dimension is weighted equally when computing distances to the class prototypes. The resulting latent space is visibly less expressive: data points for both normal and fraud classes collapse along a near-linear manifold, with limited separation between the classes and their prototypes. This collapse indicates that the model is unable to adaptively emphasize the most discriminative latent features, leading to suboptimal cluster separation and reduced interpretability. Both class prototypes tend to be positioned close to the respective cluster means, but the lack of per-dimension weighting prevents effective partitioning of ambiguous or borderline samples. As a result, the discriminative power of the model

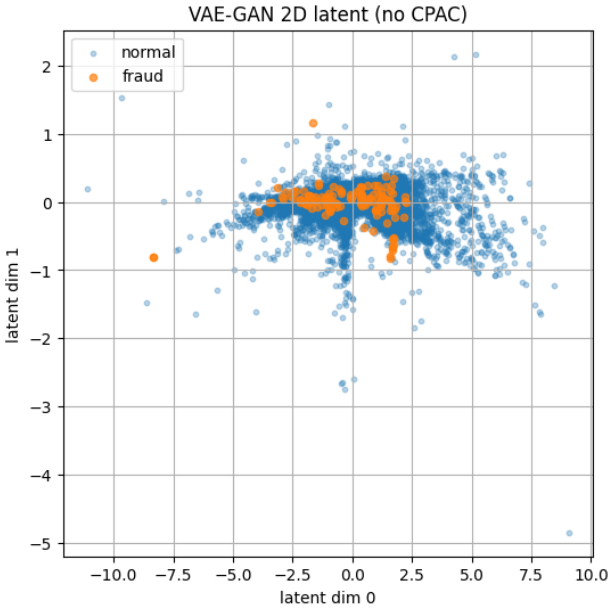


Figure 10: 2D PCA plot of the cluster representation of the Encoder without CPAC.

is noticeably diminished compared to the full CPAC, where the attention mechanism enables more nuanced, non-linear separation of minority-class samples.

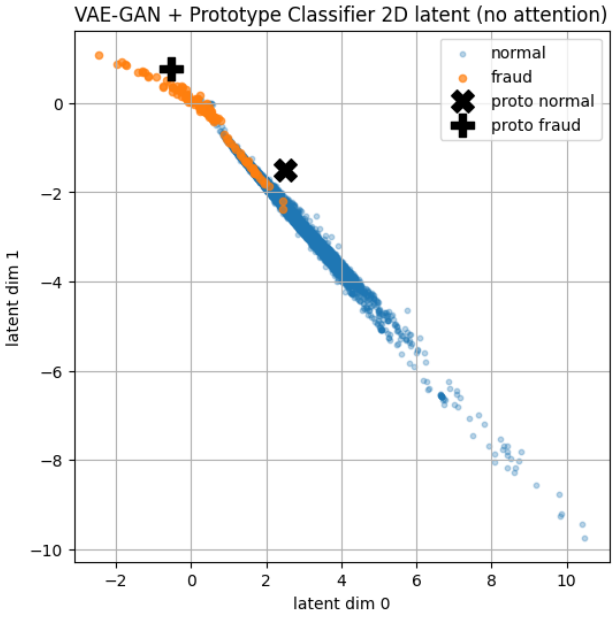


Figure 11: 2D PCA plot of the cluster representation of the Encoder of VAEGAN+CPAC without the attention mechanism.

5.3 Effect of Removing the Prototypes from CPAC

Figure 12 shows the latent space organization when the CPAC architecture is ablated to remove its prototype mechanism, leaving only the attention branch. In this setting, class discrimination relies exclusively on feature-wise weighting, with no explicit anchoring to learned class prototypes. The resulting latent representations display a pronounced collapse along a narrow, near-linear manifold, with both normal and fraud samples largely overlapping. The absence of

prototypes deprives the classifier of distinct, class-specific anchors in the latent space, severely restricting its ability to drive separation between classes. Although the attention branch allows the model to adaptively weight latent features, this alone proves insufficient for robust clustering, especially in the presence of extreme class imbalance. Consequently, the discriminative structure of the latent space deteriorates, making class boundaries ambiguous and reducing interpretability.

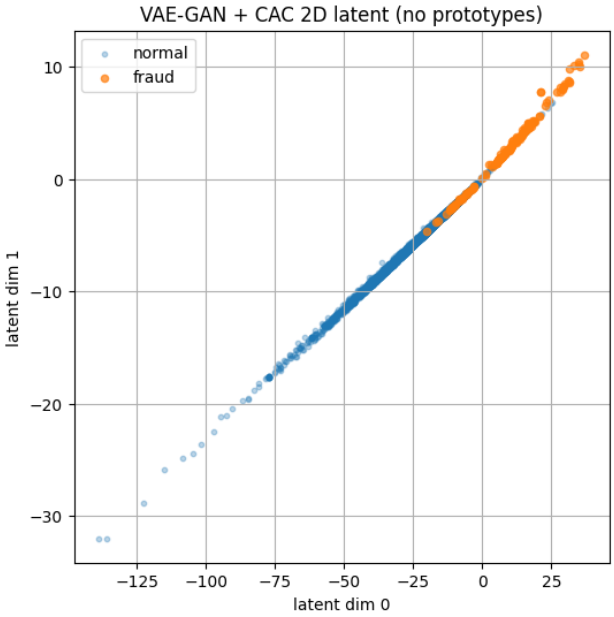


Figure 12: 2D PCA plot of the cluster representation of the Encoder of VAEGAN+CPAC without prototypes.

5.4 Effect of Removing the Anchor and Scale Penalties

Figure 13 visualizes the latent space structure when the CPAC classifier is trained without the scale and anchor regularization terms. In this setting, both the attention mechanism and learnable class prototypes remain active, but the model no longer receives explicit constraints on the spread and positioning of prototypes with respect to the cluster means. The resulting latent representations maintain a moderate level of separation between normal and fraud clusters, with prototypes positioned near the centers of their respective class distributions. However, the boundaries between clusters are less crisp than in the fully regularized setting, and the spread of both clusters along the main latent direction increases. The absence of scale and anchor penalties allows prototypes to drift from the empirical cluster centers and can result in more diffuse class boundaries, reducing both interpretability and the sharpness of latent cluster assignments.

5.5 Effect of Using Focal Loss for the CPAC

To further explore these ablations, we trained the VAE-GAN+CPAC pipeline using Focal Loss in place of standard binary cross-entropy as a loss function for the CPAC. As shown in Figure 14a, this modification results in a more distinct and well-separated clustering of fraud (minority) and normal (majority) transactions in the latent space, with prototypes more cleanly anchoring their respective clusters. The 3D latent visualization (Figure 14b) further confirms improved class-wise separation and greater dispersion of the minority class, compared to previous experiments using BCE (see Section 4). Despite this evident structural improvement, Table 9 reveals a slight decrease in downstream classification metrics across all evaluated classifiers and augmentation regimes. For instance, with 100 synthetic fraud samples, F1-scores for XGBoost and Random Forest remain above 92%, but are marginally lower than those achieved with BCE-based training. This modest drop in quantitative performance can be attributed to the nature of Focal Loss: by prioritizing hard-to-classify minority samples, it encourages the model to push fraud examples away from the decision boundary, at the expense of global average accuracy and sometimes increased variability in the majority class predictions.

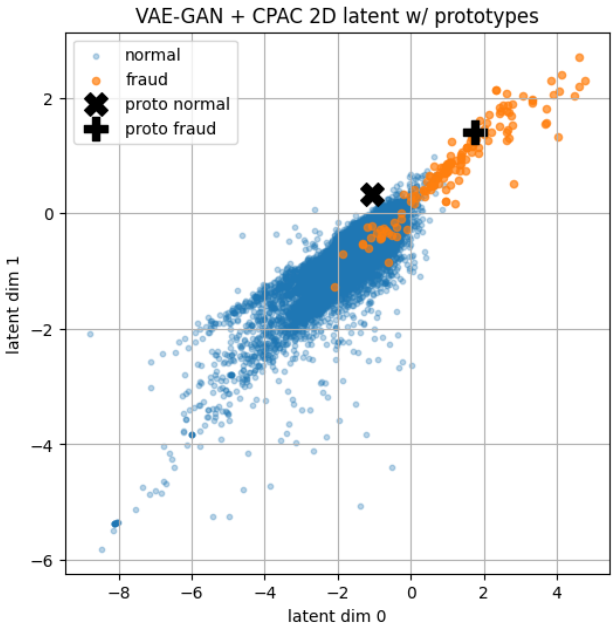
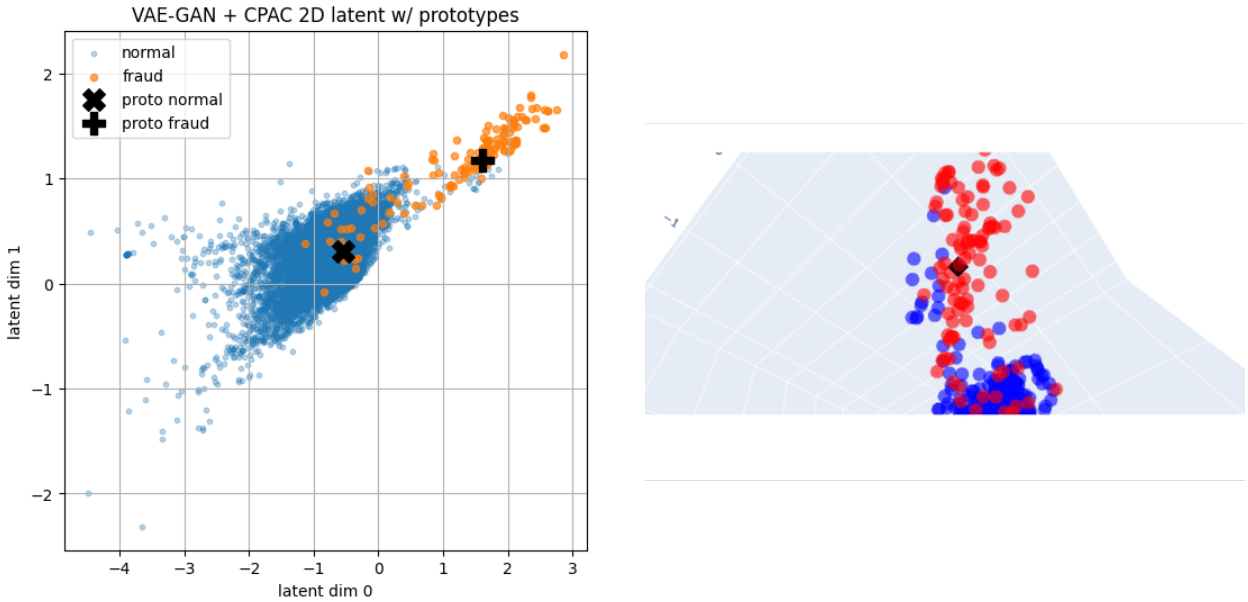


Figure 13: 2D PCA plot of the cluster representation of the Encoder of VAEGAN+CPAC without scale and anchor penalties.



(a) 2D PCA plot of the cluster representation of the Encoder of VAEGAN+CPAC using Focal Loss for CPAC.

(b) 3D plot visualisation of the overlap for the Encoder of VAE-GAN+CPAC using Focal Loss for CPAC.

Figure 14: Latent space visualizations for VAE-GAN+CPAC using Focal Loss: (a) 2D PCA plot; (b) 3D overlap visualization.

6 Discussion: Rethinking Oversampling in Fraud Detection

Despite significant advances in generative oversampling, most notably with SMOTE and VAE-GAN variants, most current state-of-the-art approaches in credit card fraud detection adopt a common strategy: training oversamplers exclusively on the minority (fraud) class. This "minority-only" paradigm is widespread, underpinned by the rationale

Table 9: Benchmark results using VAE–GAN+CPAC oversampler with Focal Loss with 50, 75, and 100 synthetic fraud samples.

# Samples	Model	Precision (%)	Recall (%)	F1-score (%)	AUC-ROC (%)
50	Logistic Regression	91.31	79.04	84.10	93.29
50	Random Forest	**98.27**	88.17	**92.63**	93.95
50	XGBoost	96.32	**88.51**	92.05	**96.35**
75	Logistic Regression	92.54	79.05	84.52	92.57
75	Random Forest	**98.27**	88.17	92.63	93.92
75	XGBoost	97.14	**89.19**	**92.79**	**96.64**
100	Logistic Regression	92.27	79.72	84.90	92.96
100	Random Forest	**98.29**	**88.51**	**92.85**	93.59
100	XGBoost	97.06	88.17	92.15	**97.35**

that a focused model can better capture rare fraudulent patterns and help rebalance the dataset. However, our results suggest that this strategy may have important limitations when applied to real-world fraud detection. Training an oversampler solely on fraud data risks generating synthetic samples that closely mimic observed frauds, rather than learning the nuanced differences between normal and fraudulent activity. As a result, such generated data may reflect interpolations within the minority class, lacking the true discriminative boundaries that are critical for effective classification. Deep and more complex classifiers rather than standard ones trained on these oversampled datasets often display overly optimistic metrics, sometimes failing to generalize robustly to new, unseen data or even overfitting even with few generated samples; simpler and classic classification approaches like the ones we used in this work struggle to improve their metrics especially in term of recall given the similarity between the synthetic frauds and the real ones. Both SMOTE and vanilla VAE–GAN oversamplers succeed in enriching the minority class and boosting downstream classifier scores. However, SMOTE's impressive precision and recall are largely artifacts of its linear interpolation, which generates highly similar fraud examples and trains classifiers to be overconfident within a narrow region of feature space. The unsupervised VAE–GAN, while producing more realistic fraud instances, still suffers from overconfidence and mode collapse, generating synthetic cases that closely mimic the original frauds, limiting generalization. When applying our CPAC classifier to these augmented datasets, we confirm a shared limitation: traditional oversamplers simply replicate, rather than expand, the fraud distribution. In contrast, our proposed VAE-GAN+CPAC pipeline is designed to address these limitations. By training on the entire dataset, including both fraud and normal transactions, and integrating a CPAC head to provide explicit, supervised class information, our approach encourages the encoder to structure its latent space to distinguish fraud from non-fraud. This design is not simply an auxiliary feature, but a central objective of the model: the classifier head directly shapes the latent representations so that generated synthetic frauds meaningfully reflect the learned class boundaries. During inference, only the minority samples are generated, but the key insight is that the model's holistic training allows it to create more realistic and discriminative synthetic data. This approach yields several practical advantages, as reflected in our benchmarks:

- **Reduced Overfitting and Overconfidence:** Models trained with our VAE-GAN+CPAC-generated frauds exhibit more stable and realistic performance, avoiding the inflated precision or recall sometimes observed with traditional oversamplers.
- **Improved Generalization:** The modest drop in certain metrics compared to some SOTA methods is, in fact, evidence of better generalization. Our synthetic frauds help downstream classifiers learn the true structure of the data, rather than simply memorizing training examples.
- **No Plateau Effect:** While standard oversamplers quickly reach a performance plateau or even degrade as more synthetic samples are added, our approach enables incremental improvements until larger sample sizes begin to induce overfitting, as expected.
- **Cluster Separation and Interpretability:** The explicit supervision provided by CPAC results in clearer separation between fraud and non-fraud clusters in latent space, facilitating interpretability and model transparency.

A key insight is that fraud can only be understood in the context of normal transactions, its definition is inherently relational. Oversamplers that ignore this context may generate synthetic data that resides within the convex hull of observed frauds, without sufficiently capturing the critical distinctions needed for robust classification. This can result in models that are prone to memorization, rather than effective discrimination.

6.1 Our Contribution

By training the VAE-GAN+CPAC on the full dataset and using the CPAC head to guide the latent representation, we bridge the gap between oversampling and supervised discriminative learning. The synthetic frauds produced by our approach are not naive copies, but instead reflect meaningful, learned differences between classes. This allows

downstream classifiers, particularly XGBoost in our experiments, to achieve strong, generalizable performance, even under extreme class imbalance. In summary, our findings support the view that the field would benefit from moving beyond the traditional minority-only oversampling paradigm toward more context-aware, discriminative approaches. Our VAE-GAN+CPAC pipeline offers a principled step in this direction.

7 Conclusions and Future Works

This work has demonstrated the limitations of minority-class-only oversampling for fraud detection and highlighted the benefits of classifier-guided, class-aware latent shaping using the Causal Prototype Attention Classifier (CPAC) within a VAE-GAN framework. Our results show that relying solely on synthetic augmentation of the minority class can lead to overconfident, poorly generalized models, whereas supervised feedback and prototype-driven clustering provide more meaningful separation and robust detection performance, as reflected in improved F1-score, recall, and AUC. The CPAC approach offers unique interpretability and flexibility, allowing both effective visualization and strong performance under extreme class imbalance. These findings advocate for a shift away from traditional oversampling techniques towards architectures that explicitly leverage class structure, supervised objectives, and interpretability. Looking ahead, there are several promising directions for advancing this line of research. Deeper or more complex neural classifiers, either as standalone detectors or as encoder heads, could potentially capture more nuanced fraud patterns. The integration of denoising strategies, such as Denoising Autoencoders, may further enhance confidence and reduce latent overlap. Broader validation across other imbalanced noise detection tasks and domains, as well as the development of richer, more transparent explanation methods, represent important next steps. Finally, exploring alternative strategies for latent space shaping, such as contrastive or manifold-regularized objectives, may yield further gains, especially in low-label or highly imbalanced settings. Overall, our approach not only advances performance metrics but also addresses the critical needs of interpretability, reliability, and resilience. While focused on credit card fraud, the proposed methods and insights generalize to a wide range of anomaly detection challenges, paving the way for more robust and trustworthy machine learning systems.

Acknowledgments

This study has been partially supported by SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] Robin Sommer and Vern Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *2010 IEEE Symposium on Security and Privacy*, 2010. doi: 10.1109/SP.2010.25.

[2] S. García, M. Grill, J. Stiborek, and A. Zunino. An empirical comparison of botnet detection methods. *Computers & Security*, 45:100–123, 2014. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2014.05.011. URL https://www.sciencedirect.com/science/article/pii/S0167404814000923.

[3] Monowar H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita. Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys & Tutorials*, 16(1):303–336, 2014. doi: 10.1109/SURV.2013.052213.00046.

[4] Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur.*, 3(3): 186–205, 2000. ISSN 1094-9224. doi: 10.1145/357830.357849. URL https://doi.org/10.1145/357830.357849.

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41 (3), 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL https://doi.org/10.1145/1541880.1541882.

[6] Robert A. Bridges, Tarrah R. Glass-Vanderlan, Michael D. Iannacone, Maria S. Vincent, and Qian (Guenevere) Chen. A Survey of Intrusion Detection Systems Leveraging Host Data. *ACM Comput. Surv.*, 52(6), 2019. ISSN 0360-0300. doi: 10.1145/3344382. URL https://doi.org/10.1145/3344382.

[7] Junfeng Peng, Ziwei Cai, Zhenyu Chen, Xujiang Liu, Mianyu Zheng, Chufeng Song, Xiongyong Zhu, Yi Teng, Ruilin Zhang, Yanqin Zhou, Xuyang Lv, and Jun Xu. An trustworthy intrusion detection framework enabled by ex-post-interpretation-enabled approach. *Journal of Information Security and Applications*, 71:103364, 2022. ISSN 2214-2126. doi: https://doi.org/10.1016/j.jisa.2022.103364. URL https://www.sciencedirect.com/science/article/pii/S2214212622002095.

[8] Andrew McCarthy, Essam Ghadafi, Panagiotis Andriotis, and Phil Legg. Defending against adversarial machine learning attacks using hierarchical learning: A case study on network traffic attack classification. *Journal of Information Security and Applications*, 72:103398, 2023. ISSN 2214-2126. doi: https://doi.org/10.1016/j.jisa.2022.103398. URL https://www.sciencedirect.com/science/article/pii/S2214212622002423.

[9] Sasha Romanosky. Examining the costs and causes of cyber incidents. *Journal of Cybersecurity*, 2(2):121–135, 2016. doi: 10.1093/cybsec/tyw001. URL https://doi.org/10.1093/cybsec/tyw001.

[10] Peter Snyder and Chris Kanich. Characterizing fraud and its ramifications in affiliate marketing networks. *Journal of Cybersecurity*, 2(1):71–81, 2016. doi: 10.1093/cybsec/tyw006. URL https://doi.org/10.1093/cybsec/tyw006.

[11] Ingolf Becker, Alice Hutchings, Ruba Abu-Salma, Ross Anderson, Nicholas Bohm, Steven J Murdoch, M Angela Sasse, and Gianluca Stringhini. International comparison of bank fraud reimbursement: customer perceptions and contractual terms. *Journal of Cybersecurity*, 3(2):109–125, 2018. doi: 10.1093/cybsec/tyx011. URL https://doi.org/10.1093/cybsec/tyx011.

[12] Salvatore Carta, Gianni Fenu, Diego Reforgiato Recupero, and Roberto Saia. Fraud detection for E-commerce transactions by employing a prudential Multiple Consensus model. *Journal of Information Security and Applications*, 46:13–22, 2019. ISSN 2214-2126. doi: https://doi.org/10.1016/j.jisa.2019.02.007. URL https://www.sciencedirect.com/science/article/pii/S2214212618304216.

[13] Zong Ke, Shicheng Zhou, Yining Zhou, Chia Hong Chang, and Rong Zhang. Detection of AI deepfake and fraud in online payments using gan-based models. *arXiv preprint arXiv:2501.07033*, 2025. URL https://arxiv.org/abs/2501.07033.

[14] Irene Amerini, Mauro Barni, Sebastiano Battiato, Paolo Bestagini, Giulia Boato, Vittoria Bruni, Roberto Caldelli, Francesco De Natale, Rocco De Nicola, Luca Guarnera, et al. Deepfake media forensics: Status and future challenges. *Journal of Imaging*, 11(3):73, 2025.

[15] Ambu Sharma and Rajesh Tiwari. Banking in the Age of Deepfakes: Evaluating Perceptions of Deepfake Fraud Risks. In *Navigating the World of Deepfake Technology*, pages 454–469. IGI Global Scientific Publishing, 2024. doi: 10.4018/979-8-3693-5298-4.ch023.

[16] Mirko Casu, Luca Guarnera, Pasquale Caponnetto, and Sebastiano Battiato. GenAI mirage: The impostor bias and the deepfake detection challenge in the era of artificial illusions. *Forensic Science International: Digital Investigation*, 50:301795, 2024. ISSN 2666-2817. doi: https://doi.org/10.1016/j.fsidi.2024.301795.

[17] David W Hosmer, Stanley Lemeshow, and Rodney X Sturdivant. *Applied Logistic Regression*. Wiley, 3rd edition, 2013. doi: 10.1002/9781118548387.

[18] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

[19] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. pages 785–794, 2016. doi: 10.1145/2939672.2939785.

[20] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. doi: 10.1613/jair.953.

[21] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2014. URL https://arxiv.org/abs/1312.6114.

[22] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/f033ed80deb0234979a61f95710dbe25-Paper.pdf.

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[24] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[25] Yixuan Li, Feng Xie, Jian Wang, and Yajie Cai. Deep learning for credit card fraud detection. *Journal of Financial Data Science*, 1(1):1–12, 2018. doi: 10.3905/jfds.2019.1.1.001.

[26] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit Card Fraud Detection Dataset, 2015. URL https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud. Accessed via Kaggle. Published by Worldline and the Machine Learning Group of ULB.

[27] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy*, 23(1), 2021. ISSN 1099-4300. doi: 10.3390/e23010018. URL https://www.mdpi.com/1099-4300/23/1/18.

[28] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. doi: 10.1613/jair.953.

[29] Alberto Fernández, Salvador García, Mikel Galar, Ricardo C. Prati, Bartosz Krawczyk, and Francisco Herrera. SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-Year Anniversary. *Journal of Artificial Intelligence Research*, 61:863–905, 2018. doi: 10.1613/jair.1.11192.

[30] Tingting Tang, Jing Yao, Yuhan Wang, Qiang Sha, Huan Feng, and Zhi Xu. Application of Deep Generative Models for Anomaly Detection in Complex Financial Transactions. *arXiv preprint arXiv:2504.15491*, 2025. URL https://arxiv.org/abs/2504.15491.

[31] Emilija Strelcenia and Simant Prakoonwit. A Survey on GAN Techniques for Data Augmentation to Address the Imbalanced Data Issues in Credit Card Fraud Detection. *Machine Learning and Knowledge Extraction*, 5(1): 304–329, 2023. doi: 10.3390/make5010019.

[32] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems, editor = H. Wallach and H. Larochelle and A. Beygelzimer and F. d'Alché-Buc and E. Fox and R. Garnett*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/adf7ee2dcf142b0e11888e72b43fcb75-Paper.pdf.

[33] Meike Nauta, Annemarie Jutte, Jesper Provoost, and Christin Seifert. This Looks Like That, Because ... Explaining Prototypes for Interpretable Image Recognition. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 441–456, Cham, 2021. Springer International Publishing.

[34] Ming Yao, Pengxu Xu, Huamin Qu, and Liu Ren. Interpretable and Steerable Sequence Learning via Prototypes. *arXiv preprint arXiv:1907.09728*, 2019. URL https://arxiv.org/abs/1907.09728.

[35] Andreas Hoffmann, Carlo Fanconi, Raphael Rade, and Jan Kohler. This Looks Like That... Does It? Shortcomings of Latent Space Prototype Interpretability in Deep Networks. *arXiv preprint arXiv:2105.02968*, 2021. URL https://arxiv.org/abs/2105.02968.

[36] Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357.

[37] Sarah Wiegreffe and Yuval Pinter. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1002.

[38] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. doi: 10.1145/2939672.2939778.

[39] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[40] Morteza Rakhshaninejad, Mohammad Fathian, Babak Amiri, and Navid Yazdanjue. An ensemble-based credit card fraud detection algorithm using an efficient voting strategy. *The Computer Journal*, 65(8):1998–2015, 05 2021. ISSN 0010-4620. doi: 10.1093/comjnl/bxab038. URL https://doi.org/10.1093/comjnl/bxab038.

[41] D. Wang and Y. Yao. Unrolled GAN-Based Oversampling of Credit Card Dataset for Fraud Detection. In *2022 2nd International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 722–727, Dalian, China, 2022. IEEE. doi: 10.1109/ICAICA55605.2022.9844421. URL https://ieeexplore.ieee.org/document/9844421.

[42] Weijun Ding, Qian Kang, and Yuling Feng. Credit Card Fraud Detection Based on Improved Variational Autoencoder Generative Adversarial Network. *IEEE Access*, 11:84545–84556, 2023. doi: 10.1109/ACCESS.2023.10210017. URL https://ieeexplore.ieee.org/document/10210017.

[43] Si Shi, Wuman Luo, and Giovanni Pau. An attention-based balanced variational autoencoder method for credit card fraud detection. *Applied Soft Computing*, 177:113190, 2025. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2025.113190. URL https://www.sciencedirect.com/science/article/pii/S1568494625005010.

[44] Khanda Hassan Ahmed, Stefan Axelsson, Yuhong Li, and Ali Makki Sagheer. A credit card fraud detection approach based on ensemble machine learning classifier with hybrid data sampling. *Machine Learning with Applications*, 20:100675, 2025. ISSN 2666-8270. doi: https://doi.org/10.1016/j.mlwa.2025.100675. URL https://www.sciencedirect.com/science/article/pii/S2666827025000581.

[45] R. Alejo, J. M. Sotoca, R. M. Valdovinos, and P. Toribio. Edited Nearest Neighbor Rule for Improving Neural Networks Classifications. In Liqing Zhang, Bao-Liang Lu, and James Kwok, editors, *Advances in Neural Networks - ISNN 2010*, pages 303–310. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13278-0.

[46] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, ICCV '17, pages 2980–2988, New York, NY, USA, 2017. IEEE. doi: 10.1109/ICCV.2017.324.