Less Learning, More Detection: Identity Document Presentation Attack Detection with Frozen Features

Jure Pahor, Peter Peer, Vitomir Štruc, Borut Batagelj
Faculty of Computer and Information Science, University of Ljubljana
Večna pot 113, Ljubljana, Slovenia
jp7316@student.uni-lj.si, peter.peer@fri.uni-lj.si
vitomir.struc@fe.uni-lj.si, borut.batagelj@fri.uni-lj.si

Abstract

Remote identity verification systems are increasingly exposed to physical presentation attacks (PAs), including print, screen, and composite attacks on identity documents. Detecting such attacks, commonly referred to as identity document presentation attack detection (PAD), is challenging because the relevant forensic cues are often local, subtle, and sensitive to acquisition conditions. Recent PAD methods therefore rely on increasingly sophisticated deep learning pipelines, including task-specific heads, learned patch fusion modules, and backbone adaptation. In this work, we revisit identity document PAD from the perspective of feature reuse and ask whether frozen self-supervised foundation model representations already encode the local forensic cues needed for reliable detection. We propose PADCore, a lightweight patch-based framework that decomposes each document into local patches, embeds them using a frozen DINOv2 (or DINOv3) backbone, constructs compact per-class coresets of representative embeddings, and fits a Linear Discriminant Analysis classifier on top. PADCore requires no backbone fine-tuning and can be efficiently re-fitted as new attack data becomes available. On FakeIDet2-db, PADCore matches or surpasses the state-of-the-art FakeIDet2 approach in in-distribution evaluation and achieves stronger generalization on KID34K, while reducing training time from hours to under four minutes. Its patch-based design is also compatible with privacy-preserving workflows based on pre-extracted anonymized document patches.

1. Introduction

Remote identity verification has become a cornerstone of modern digital services, supporting *Know Your Customer* (KYC) procedures in domains such as digital banking, online marketplaces, and government platforms. In this setting, users are typically required to submit an image of an identity document, which must be automatically verified for authenticity. The security of such systems is increasingly challenged by presentation attacks, including print, screen, and composite manipulations, which aim to replicate or alter legitimate documents. As highlighted in recent surveys [15], robust generalization across document types, acquisition conditions, and attack mechanisms remains one of the central open problems in identity document presentation attack detection (PAD).

Recent progress in identity document PAD has largely followed advances in deep learning. Early approaches relied on convolutional neural networks trained end-to-end on document images [8, 6], while more recent methods have adopted Vision Transformers and foundation models, often combined with task-specific fine-tuning strategies [17]. In parallel, patch-based formulations have gained traction, motivated by the observation that forensic cues such as halftone patterns, sub-pixel structures, and material inconsistencies in recent identity documents manifest at a local spatial scale. Methods such as FakeIDet and FakeIDet2 [10, 9] exploit this paradigm by extracting document patches and learning discriminative representations using frozen or partially adapted foundation model backbones combined with learned classification and fusion modules.

At the same time, a growing body of work across computer vision has demonstrated that powerful self-supervised foundation models, such as DINOv2 [11] and its successors [16], encode rich and transferable visual representations that can be effectively reused without task-specific adaptation. In domains such as anomaly detection, approaches like PatchCore [14] and AnomalyDINO [3] have shown that competitive performance can be achieved by combining frozen patch-level features with lightweight, non-parametric or classical statistical models.

Motivated by these observations, we revisit identity document PAD from the perspective of feature reuse. Specif-

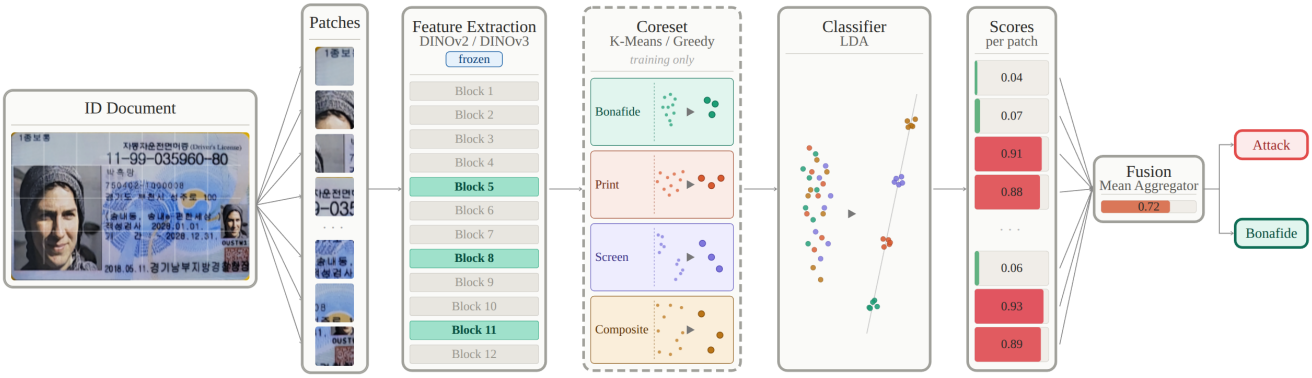

Figure 1. Graphical representation of PADCore, a patch-based approach for identity document presentation attack detection using frozen foundation model features and coreset-based classification. The ID card is taken from the KID34K dataset [12].

ically, we investigate whether the discriminative information required for PAD is already present in frozen foundation model representations, and to what extent effective classification can be achieved without adapting the backbone. Rather than focusing on learning new representations, we treat PAD as a problem of modelling the structure of patch-level feature distributions. To this end, we propose PADCore, a lightweight patch-based PAD framework that combines frozen foundation model features with coreset-based feature selection and classical classification. To facilitate PADCore processing, each document is first decomposed into local patches, embedded using a frozen DINOv2 (or DINOv3) backbone, and represented through a compact per-class coreset constructed in the respective feature space. A Linear Discriminant Analysis (LDA) classifier [18] is then fitted on top of these coresets to perform patch-level classification, with document-level decisions obtained via aggregation, as illustrated in Figure 1.

This design offers several practical advantages. First, it avoids backbone fine-tuning entirely, reducing training time to a few minutes and enabling rapid re-fitting as new attack data becomes available. Second, by operating on pre-extracted patches, it is naturally compatible with privacy-preserving settings such as those considered in FakeIDet2 [9]. Third, the coreset formulation provides a flexible mechanism for controlling the trade-off between efficiency, robustness, and generalization. To evaluate PADCore, we conduct extensive evaluations on the FakeIDet2-db benchmark [9], as well as cross-dataset experiments on DLC-2021 [13] and KID34K [12]. Our results show that PADCore achieves highly competitive performance compared to more complex learned pipelines, particularly in in-distribution settings and under various forms of domain shift, while maintaining significantly lower computational cost. In summary, we make the following key contributions in this paper:

- We introduce PADCore, a lightweight identity document PAD framework that combines frozen foundation model representations with coreset-based classification, eliminating backbone fine-tuning while remaining compatible with privacy-preserving patch-based processing.
- We present a systematic study of frozen patch-level representations for identity document PAD, analyzing the impact of transformer depth, backbone choice, patch size, anonymization level, and coreset construction strategy.
- We show that PADCore matches or surpasses state-of-the-art learned pipelines on FakeIDet2-db and improves generalization on KID34K, while reducing training time from hours to minutes and enabling efficient re-fitting as new attack data becomes available.

2. Related Work

2.1. Image-Level PAD Methods

Image-level approaches to identity document presentation attack detection (PAD) operate on the full document image, learning to classify it as genuine or fake using deep neural networks. Early work on image-level PAD relied on convolutional neural networks, including the pixel-wise supervision framework of Mudgalgundurao *et al.* [8] and the two-stage architecture of Gonzalez and Tapia [6], demonstrating that end-to-end training on document images can effectively detect print and screen attacks under controlled conditions. More recent approaches have adopted Vision Transformers and large-scale pre-trained models, with competition systems fine-tuning CLIP (ViT-L/14) [2] and DINOv2 [11] backbones using parameter-efficient techniques such as LoRA [17] for improved performance.

While these methods benefit from strong global representations, they typically operate on downsampled inputs (e.g., 224×224 or 384×384 pixels), which may limit their ability to capture fine-grained local artefacts. Furthermore, image-level representations can become sensitive to document layout and appearance, potentially affecting generalization across unseen document types and acquisition conditions, as observed in the IJCB 2025 competition [17].

2.2. Patch-Level PAD Methods

Patch-based approaches address these limitations by decomposing the document into local regions and performing classification at the patch level, preserving high-frequency texture information and aligning the analysis with the spatial scale of forensic cues. FakeIDet [10] introduced this paradigm for identity document PAD by extracting non-overlapping patches of 64×64 or 128×128 pixels from anonymized images and classifying each patch using a frozen DINOv2 [11] backbone with a lightweight sigmoid head, aggregating predictions to obtain a document-level decision. FakeIDet2 [9] extended this approach by introducing learned components on top of the frozen backbone. Specifically, it incorporates a trainable patch embedding layer optimized using margin-based losses such as CosFace [19], ArcFace [4], and AdaFace [7], as well as a multi-head self-attention module to model interactions between patches before producing a global representation. This combination yields state-of-the-art performance on the FakeIDet2-db benchmark [9].

PADCore follows the same patch-level paradigm but explores a different point in the design space. Instead of learning task-specific representations or fusion modules, it leverages frozen foundation model features directly and focuses on modelling their distribution using coreset-based selection and classical classification.

2.3. Embedding-Based Approaches in Other Fields

The reuse of frozen foundation model embeddings combined with lightweight decision functions has proven effective across several computer vision tasks. In industrial anomaly detection, PatchCore [14] demonstrated that a memory bank of sampled patch-level features can achieve strong performance without training deep models. Similarly, AnomalyDINO [3] showed that frozen DINOv2 [11] representations can rival more complex learned approaches in few-shot anomaly detection settings.

These works highlight that, in certain regimes, the representational capacity of foundation models can be sufficient, shifting the focus from representation learning to effective modelling of feature space structure. PADCore builds on this insight and adapts it to the identity document PAD setting.

3. Proposed Method: PADCore

In this section, we now present PADCore, the proposed lightweight patch-based framework for identity document presentation attack detection, which operates by modelling the distribution of frozen patch-level features using coreset-based selection and a simple statistical classifier.

3.1. Patch Extraction

Following the patch-level paradigm established by FakeIDet and FakeIDet2 [10, 9], each identity document image is first divided into a regular grid of non-overlapping square patches by sliding a window with stride equal to the patch size. We evaluate two patch sizes, 64×64 and 128×128 pixels, matching the configurations used in prior work to enable direct comparisons. Each extracted patch is resized to 224×224 pixels to conform to the input resolution expected by the backbone. For FakeIDet2-db [9], we use the provided pre-extracted patches directly after resizing across all experimental settings considered.

3.2. Patch Embedding

Each patch is embedded using a frozen self-supervised vision foundation model to produce a compact feature vector. We consider two backbones: DINOv2 [11] and DINOv3 [16], both based on Vision Transformers trained via self-distillation on large-scale image corpora. We use a ViT-S/14 variant for DINOv2 and a ViT-S/16 variant for DINOv3. In all experiments, backbone weights remain fixed, with no task-specific adaptation.

For each patch, we extract the CLS token at the output of a chosen transformer block, which aggregates information from all patch tokens and serves as a global representation. Rather than relying solely on the final layer, we explore representations from different depths of the network in our experiments across all configurations. This is motivated by the observation that earlier layers tend to encode low-level properties such as color and texture, while deeper layers capture more semantic information. Recent analyses of Vision Transformers confirm that this progression also holds in self-attention-based architectures [5]. Since many forensic cues relevant to PAD, such as halftone patterns or sub-pixel structures, are inherently low-level, the choice of feature depth can significantly impact performance.

Because the backbone is frozen, patch embeddings are deterministic and can be computed once and cached. This enables rapid experimentation with downstream components, as coreset construction and classifier fitting operate directly on pre-computed features.

3.3. Coreset Selection

Training directly on all patch embeddings is both computationally inefficient and unnecessary due to the high degree of redundancy in the data. PADCore instead constructs a compact coreset, i.e., a fixed-size subset of representative feature vectors selected independently per class, on which the downstream classifier is trained. This coreset captures the structure of each class distribution while significantly reducing computational cost and enabling efficient re-fitting. We consider two coreset construction strategies, applied independently per class to maintain balance:

- **Greedy farthest–point sampling**, adapted from PatchCore [14], iteratively selects the point that is maximally distant from the current coreset, ensuring broad coverage of the feature space. This strategy preserves the geometric support of the distribution, including outlier regions that may correspond to rare but informative patterns.
- **K-Means** clustering partitions the feature space into k clusters and uses the cluster centroids as representatives. Each centroid summarizes a local neighbourhood, effectively capturing the dominant modes of the distribution. We use Mini-Batch K-Means to ensure scalability to large datasets.

Both methods are implemented with GPU acceleration to handle large patch collections efficiently.

3.4. Attack Classifier

Before classification, each feature dimension of the coreset is standardized to zero mean and unit variance. A multiclass Linear Discriminant Analysis (LDA) classifier [18] is then fitted on the normalized features. LDA learns a projection that maximizes between-class variance while minimizing within-class variance, producing a low-dimensional space in which the four classes (bonafide, print, screen, and composite) are well separated in the resulting feature space. We adopt LDA over alternative lightweight classification heads as it is a natural fit for the coreset representation: K-Means centroids act as smoothed, roughly Gaussian class prototypes, matching the shared-covariance assumptions underlying LDA, which can thus learn class-separating directions directly rather than relying on distances in the raw feature space.

At inference time, each patch from a test document is projected into this space and assigned a class probability. These patch-level predictions are aggregated via mean pooling to produce a single document-level decision, following the strategy used in FakeIDet [10].

4. Experimental Setup

This section describes the datasets, evaluation metrics, and experimental protocols used to assess PADCore under both in-distribution and cross-dataset settings.

4.1. Datasets

We evaluate PADCore on one primary in-distribution benchmark and two cross-dataset benchmarks that differ in document type, acquisition setup, and attack characteristics.

FakeIDet2-db [9] is our primary benchmark. It contains 922,057 patches extracted from 2,000 images of 47 official Spanish identity documents, captured using three smartphones, three acquisition heights, and five illumination conditions. The database includes bonafide samples as well as print, screen, and physical composite attacks. To our knowledge, it is the only publicly available identity document PAD benchmark containing both officially issued bonafide documents and composite attack samples. To preserve the privacy of document holders, the dataset is distributed as pre-extracted anonymized patches rather than full document images for all experimental configurations considered.

DLC-2021 [13] is used for cross-dataset evaluation. It consists of 1,424 video clips of mock identity documents from the MIDV-2020 collection [1], covering color print, grayscale print, and screen recapture attacks across ten document types from multiple countries. Its bonafide samples are researcher-produced mock documents rather than officially issued IDs, which limits the realism of the genuine class under more diverse acquisition conditions.

KID34K [12] is also used for cross-dataset evaluation. It contains 34,662 images of replica Korean ID cards, spanning genuine, screen, and print scenarios across 82 cards and 12 acquisition devices. Although the bonafide samples are replicas rather than official documents, the authors report that replica and real ID feature distributions are closely aligned for the evaluated recognition tasks.

4.2. Evaluation Metrics

We evaluate PADCore according to the ISO/IEC 30107-3 standard for biometric presentation attack detection[1], using the same metrics as prior work on identity document PAD [10, 9, 17]. The Attack Presentation Classification Error Rate (APCER) measures the proportion of attack samples incorrectly classified as bonafide:

$$\mathrm{APCER}_{\mathrm{PAIS}}(\tau) = 1 - \frac{1}{N_{\mathrm{PAIS}}} \sum_{i=1}^{N_{\mathrm{PAIS}}} \mathrm{RES}_i, \qquad (1)$$

where N_{PAIS} is the number of attack samples for a given Presentation Attack Instrument Species (PAIS), such as print, screen, or composite, and $\mathrm{RES}_i = 1$ if the i-th sample is classified as an attack at threshold τ, and 0 otherwise. The Bonafide Presentation Classification Error Rate (BPCER) measures the proportion of bonafide samples incorrectly classified as attacks:

$$\mathrm{BPCER}(\tau) = \frac{\sum_{i=1}^{N_{\mathrm{BF}}} \mathrm{RES}_i}{N_{\mathrm{BF}}}, \qquad (2)$$

where N_{BF} is the number of bonafide samples. The Equal Error Rate (EER) is defined at the operating threshold τ^* where $\mathrm{APCER}(\tau^*) = \mathrm{BPCER}(\tau^*)$, providing a threshold-independent summary of detection performance.

To ensure statistical reliability, we compute metrics using stratified bootstrapping on the test set, following FakeIDet2 [9]. Test instances are resampled 10,000 times with

[1] https://www.iso.org/standard/79520.html

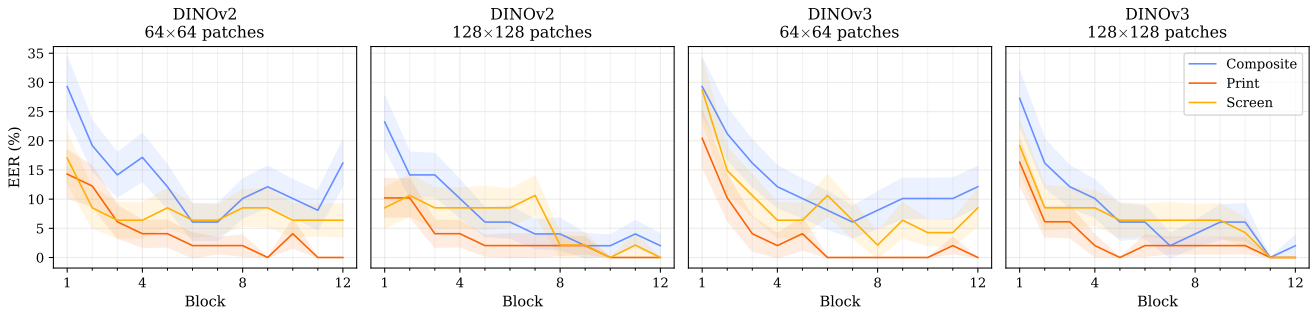

Figure 2. Per-attack-type EER (%) for PADCore (K-Means coreset, 2,000 elements per class) across ViT blocks 1–12 for DINOv2 (left) and DINOv3 (right) on FakeIDet2-db [9] configurations. Values are mean ± std. over 10,000 stratified bootstrap iterations of the test set.

replacement while preserving the original class proportions, and we report the mean and standard deviation of the resulting EER distribution.

4.3. Evaluation Protocol

All experiments on FakeIDet2-db follow the standard protocol of Muñoz-Haro *et al*. [9]. The official train/test splits are provided with the database[2] and are defined at the identity level, with 80% of identities used for training and the remaining 20% for testing. This ensures that no identity appears in both training and test sets. We evaluate all four released configurations, obtained from the combination of two patch sizes (64×64 and 128×128) and two anonymization levels (pseudo-anonymized and fully anonymized).

For cross-dataset evaluation, we again follow FakeIDet2 [9]: PADCore is trained exclusively on FakeIDet2-db and evaluated on DLC-2021 [13] and KID34K [12] as out-of-distribution test sets. For DLC-2021, we use the 30,000-image subset considered in FakeIDet2, while for KID34K we use 17,700 images sampled from both the original train and test sets. Since neither DLC-2021 nor KID34K contains officially issued real IDs, bonafide samples are taken from the FakeIDet2-db test set. This enables direct comparison with FakeIDet2 under the same evaluation conditions, with FakeIDet2 serving as the stronger baseline over FakeIDet [10, 9] in prior comparative experiments.

5. Results and Analysis

In this section, we analyze the design choices of PADCore and evaluate its performance against state-of-the-art methods in both in-distribution and cross-dataset settings.

5.1. Backbone Selection

Figure 2 shows per-attack-type EER across all 12 transformer blocks for both backbones and pseudo-anonymized FakeIDet2-db configurations. The first three blocks consistently yield the worst performance across all settings,

[2] https://github.com/BiDAlab/FakeIDet2-db

Table 1. Per-attack bootstrap EER (%) for PADCore with DINOv2 [11], comparing the final transformer block (12) against multi-layer fusion (blocks 5, 8, and 11). Hyperparameters match Figure 2. Best EER per group shown with **bold** block label; best per-attack EER is highlighted.

Anon.	Patch	Blocks	Composite	Print	Screen	All
Full	64	12	18.2 ± 4.6	0.0 ± 0.0	5.9 ± 3.3	10.1 ± 2.9
		5-8-11	3.6 ± 2.5	1.0 ± 1.5	3.2 ± 2.3	**2.8 ± 1.9**
Pseudo	64	12	14.6 ± 3.9	0.0 ± 0.0	5.3 ± 2.8	8.4 ± 2.3
		5-8-11	4.4 ± 2.4	0.0 ± 0.0	5.4 ± 2.7	**4.0 ± 1.8**
Full	128	**12**	5.1 ± 2.9	0.0 ± 0.0	2.0 ± 2.2	**3.3 ± 1.8**
		5-8-11	3.3 ± 2.5	1.0 ± 1.5	6.5 ± 3.0	3.9 ± 1.9
Pseudo	128	**12**	2.8 ± 2.0	0.0 ± 0.0	0.0 ± 0.0	**1.6 ± 1.3**
		5-8-11	2.2 ± 1.9	0.9 ± 1.4	8.5 ± 3.7	3.6 ± 1.6

Table 2. Per-attack bootstrap EER (%) as in Table 1, but with the DINOv3 [16] backbone.

Anon.	Patch	Blocks	Composite	Print	Screen	All
Full	64	12	15.4 ± 4.4	0.0 ± 0.0	6.7 ± 3.4	9.7 ± 2.7
		5-8-11	7.3 ± 3.5	0.0 ± 0.0	0.0 ± 0.0	**2.6 ± 1.6**
Pseudo	64	12	12.2 ± 3.5	0.0 ± 0.0	6.9 ± 3.1	8.6 ± 2.8
		5-8-11	3.6 ± 2.2	0.0 ± 0.0	0.0 ± 0.0	**1.9 ± 1.3**
Full	128	12	7.3 ± 3.4	0.0 ± 0.0	1.9 ± 2.1	3.6 ± 1.8
		5-8-11	3.4 ± 2.5	0.0 ± 0.0	4.0 ± 2.6	**2.8 ± 1.6**
Pseudo	128	12	1.5 ± 1.8	0.0 ± 0.0	0.0 ± 0.0	0.7 ± 1.0
		5-8-11	0.0 ± 0.0	0.0 ± 0.0	1.5 ± 1.8	**0.7 ± 1.0**

suggesting that very early representations lack sufficient discriminative structure for PAD. Beyond this point, clear attack-specific trends begin to emerge. Print attacks are detected well across most block depths. Composite attacks benefit most from intermediate blocks (4–8), especially in the 64×64 patch size setting, indicating that the color and texture discontinuities at field boundaries require some degree of contextual aggregation for detection. Screen attacks show a different pattern, favoring deeper blocks (8–12), especially in the 128×128 patch size setting, consistent with the more subtle, globally distributed nature of sub-pixel and Moiré artefacts within the evaluated feature hierarchy.

Based on these trends, we experiment with concatenating CLS token features from multiple intermediate blocks

rather than relying on a single layer. We empirically found that concatenating features from blocks 5, 8, and 11 performs well across configurations and attack types, and adopt this as our multi-block setup for comparison against the single final block (block 12).

Tables 1 and 2 show that the multi-block setup yields substantial improvements for composite attacks across virtually all configurations and both backbones, which aligns with the intermediate-block preference observed in Figure 2. These gains occasionally come at a cost for screen attacks: since later blocks tend to encode more discriminative features for this attack type, as observed in Figure 2, their reduced influence in the multi-block concatenation leads to performance degradation, most visibly in the 128×128 settings for DINOv2. Nevertheless, the multi-block approach outperforms the final block in overall EER in most configurations, and when it does not, the difference is negligible. Comparing the two backbones, DINOv3 consistently matches or outperforms DINOv2 across configurations, though the differences are generally modest, suggesting that both backbones encode similarly useful forensic representations.

5.2. Coreset Selection Strategy

Figure 3 shows the combined EER as a function of coreset size per class for both K-Means and greedy farthest-point sampling, along with a no coreset baseline trained directly on all available patch features. K-Means consistently outperforms greedy sampling across all coreset sizes, and both strategies gradually converge toward the no coreset baseline as coreset size increases. This convergence is expected. As the number of K-Means clusters increases, each centroid represents a smaller neighbourhood and increasingly approximates the underlying feature distribution. Similarly, greedy sampling covers a denser portion of the feature space, approaching the behaviour of training on the full feature set. We attribute the advantage of K-Means to the nature of its centroids. Rather than selecting actual observed patch features, K-Means produces computed averages of local neighbourhoods in feature space, smoothing over fine-grained intra-class variation and presenting the downstream LDA classifier with idealized class prototypes. This acts as an implicit regularization, smoothing out the noise and redundancy of the raw feature set before it reaches the classifier, evidenced by the no coreset baseline performing worse than the K-Means coreset strategy.

Rather than treating the coreset size as a hyperparameter to be tuned per dataset, we interpret it as a deployment-dependent parameter controlling a trade-off between prototype smoothing and class-distribution coverage. Small coresets emphasize the dominant modes of each class and suffice when training and test conditions are closely matched: on FakeIDet2-db [9], 2,000 elements per class

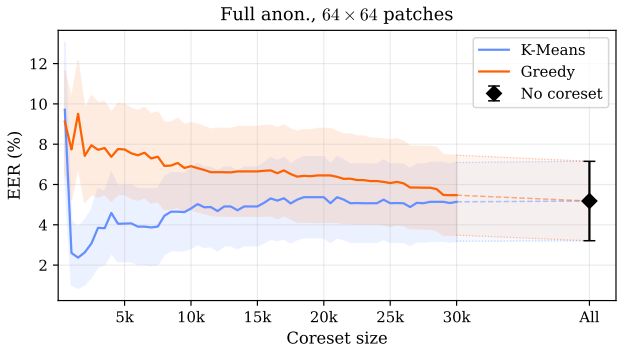

Figure 3. EER (%) vs. per-class coreset size for K-Means and greedy farthest-point sampling on fully-anonymized 64×64 FakeIDet2-db [9] patches, with a no-coreset baseline for reference. Features: concatenated DINOv3 [16] blocks 5, 8, and 11.

proved stable across all four configurations, despite substantial differences in the number of available training patches (e.g., 330,772 patches in the fully-anonymized 64×64 setting versus 64,562 in the fully-anonymized 128×128 one), so a fixed absolute size requires no per-configuration tuning. Larger coresets, in contrast, preserve rarer source-domain modes that may become relevant under distribution shift, at the cost of weaker implicit regularization. We therefore recommend scaling the coreset to the anticipated deployment shift: smaller for stable in-domain use, larger when broader shift is expected, defaulting to larger sizes under uncertainty. Following this rationale, the experiments in Sections 5.3 and 5.4 use 10,000 elements per class, since both evaluate PADCore beyond its training distribution.

5.3. Robustness to Unseen Attack Types

To evaluate robustness to unseen attack mechanisms, we follow the leave-one-attack-out protocol from FakeIDet2 [9]. In each setting, PADCore is trained with one attack type removed and evaluated on all attack categories, allowing us to assess how well the frozen patch-level feature space supports extrapolation beyond the attack classes represented in the coreset. Statistical reliability is assessed by running PADCore 100 times with different random seeds for coreset construction and reporting the mean $\pm$ standard deviation of the resulting EER distribution. All experiments use a K-Means coreset of 10,000 points per class.

Table 3 compares PADCore and FakeIDet2 [9] in both the full training and leave-one-attack-out scenarios on the pseudo-anonymized 64×64 configuration of FakeIDet2-db [9]. When all attack types are available during training, PADCore consistently outperforms FakeIDet2, achieving $3.79 \pm 0.76\%$ overall EER across 100 seeds compared to 8.64% for FakeIDet2, with particularly large gains on composite attacks. The standard deviation across seeds remains modest, indicating that PADCore is stable with respect to coreset initialization on in-distribution data. In

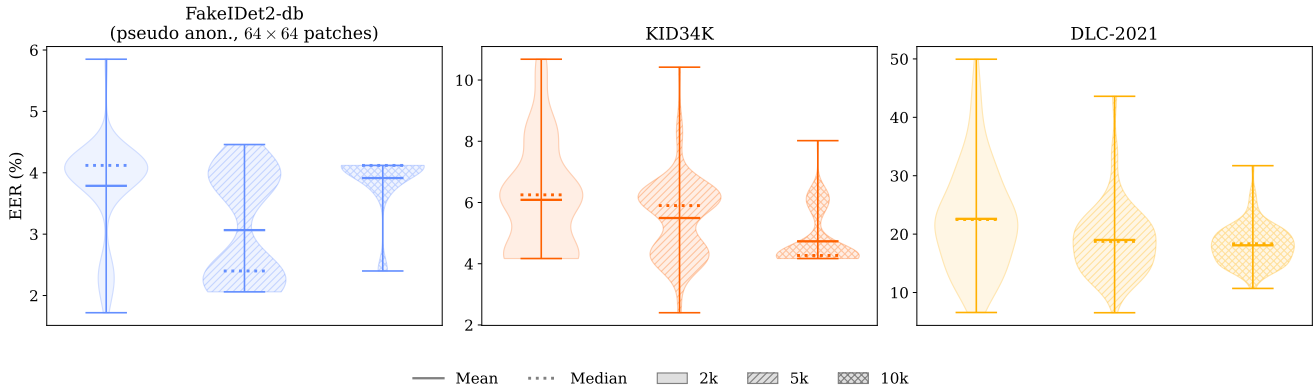

Figure 4. Cross-dataset EER (%) for PADCore using DINOv3 [16] features from blocks 5, 8 and 11 using K-Means coresets of sizes 2,000, 5,000 and 10,000 features per class, where different textures present coreset sizes.

the leave-one-attack-out scenarios, PADCore outperforms FakeIDet2 in most settings, with the notable exception of the no-print scenario, where it degrades sharply on unseen print attacks, driving the overall EER to $17.95 \pm 2.14\%$ versus FakeIDet2's 7.97%. This is an expected consequence of the coreset-based design: without any print patches in the training set, the LDA classifier has no print class prototype to separate against, and print patches (whose forensic artefacts, such as glossy reflections, are subtle compared to the more visually disruptive traces of screen and composite attacks) are likely collapsed into the bonafide class, as they share more low-level texture similarity with genuine document patches than with any of the other attack types present during training. In the no-composite and no-screen scenarios PADCore recovers its advantage, achieving $25.25 \pm 1.63\%$ and $8.89 \pm 1.69\%$ overall EER respectively, compared to 28.24% and 19.93% for FakeIDet2, suggesting that PADCore's frozen features generalize better across unseen attack types than the learned representations of FakeIDet2, with the exception of the print attack class.

5.4. Cross-Dataset Generalization

Table 3. Per-attack EER (%) in leave-one-attack-out training on pseudo-anonymized 64×64 FakeIDet2-db patches. PADCore: mean $\pm$ std. over 100 seeds. FakeIDet2: single run from [9].

Training	Model	Composite	Print	Screen	All
No-Screen	FakeIDet2	7.17	1.00	36.00	19.93
	PADCore	3.7 ± 1.4	0.1 ± 0.4	17.9 ± 3.8	$\mathbf{8.9 \pm 1.7}$
No-Print	**FakeIDet2**	12.25	6.77	3.05	**7.97**
	PADCore	4.0 ± 0.6	33.8 ± 4.4	2.0 ± 1.3	18.0 ± 2.1
No-Comp.	FakeIDet2	50.96	0.00	2.02	28.24
	PADCore	49.8 ± 2.9	0.0 ± 0.0	0.9 ± 1.1	$\mathbf{25.3 \pm 1.6}$
All-Attacks	FakeIDet2	16.29	4.89	5.01	8.64
	PADCore	5.1 ± 1.3	0.0 ± 0.2	2.2 ± 1.4	$\mathbf{3.8 \pm 0.8}$

We next evaluate whether PADCore generalizes beyond the dataset used for training by testing on DLC-2021 [13] and KID34K [12] under the FakeIDet2 cross-dataset protocol [9]. Unlike the in-distribution and leave-one-attack-out experiments, this setting introduces both new document appearances and new acquisition conditions, making coreset stability especially important. Figure 4 analyzes the sensitivity of cross-dataset performance to coreset size and random initialization. Figure 4 shows the distribution of EER across 100 seeds for coreset sizes of 2,000, 5,000, and 10,000 points per class on FakeIDet2-db [9], KID34K [12], and DLC-2021 [13], following the protocol of Section 4.3. While performance on FakeIDet2-db remains tightly concentrated regardless of coreset size, variance on DLC-2021, the dataset least similar to FakeIDet2-db, is substantially higher at small coreset sizes, with individual seeds spanning a wide range of outcomes. Increasing the coreset size progressively narrows this distribution, with 10,000 features per class yielding predictable behaviour and eliminating the near-failure cases, sometimes at the cost of peak performance. We attribute this to the regularizing effect of larger coresets discussed in Section 5.2: with more points, the influence of any single random initialization on the class prototypes is diminished, yielding a more stable approximation of the underlying feature distribution, whereas a poorly initialized small coreset may produce a boundary that aligns arbitrarily well or poorly with the shifted test distribution. For the results below, we therefore report mean and standard deviation across 100 seeds with the 10,000-point K-Means variant, exposing this behaviour rather than masking it with a single run.

Table 4 reveals a clear asymmetry in PADCore's cross-dataset generalization. On KID34K [12], PADCore consistently outperforms FakeIDet2 across all attack types, showing promising generalization to unseen characters and achieving $4.73 \pm 0.85\%$ overall EER compared to $14.47 \pm 2.17\%$. On DLC-2021 [13], however, PADCore struggles on two attack types in particular. Gray-print attacks, which are entirely absent from FakeIDet2-db, mirror the no-print leave-one-attack-out failure mode observed in Table 3. Without gray-print patches in the coreset, the classifier lacks a prototype for this class and likely collapses

Table 4. Cross-dataset per-attack EER (%); both models trained on pseudo-anonymized 64×64 FakeIDet2-db patches. PADCore: mean $\pm$ std. over 100 seeds. FakeIDet2: mean $\pm$ std. over 10,000 stratified bootstrap iterations of test set.

	DLC-2021		KID34K	
PAI	**FakeIDet2**	PADCore	FakeIDet2	**PADCore**
Screen	4.57 ± 0.54	27.61 ± 4.21	14.44 ± 1.38	4.14 ± 0.29
HQ-Print	11.99 ± 1.11	12.59 ± 4.63	15.31 ± 2.27	5.88 ± 1.16
Print	10.97 ± 1.26	11.79 ± 5.01	6.32 ± 1.12	4.56 ± 1.04
Gray-Print	10.59 ± 1.24	21.86 ± 5.36	–	–
All	$\mathbf{9.56 \pm 0.83}$	18.08 ± 3.43	14.47 ± 2.17	$\mathbf{4.73 \pm 0.85}$

Table 5. Training time for PADCore (DINOv3 [16], blocks 5/8/11) vs. FakeIDet2 [9]. FakeIDet2 range reflects best- and worst-case scenarios from reported epoch times and early stopping.

Anon.	Patch	Model	Feat. Extr.	Rest	Total
Full	64	FakeIDet2	–	–	1 h 32 min – 7 h 42 min
		PADCore	2 min 25s	9.2s	**2 min 35s**
Full	128	FakeIDet2	–	–	1 h 8 min – 2 h 27 min
		PADCore	31.7s	5.4s	**37.1s**
Pseudo	64	FakeIDet2	–	–	1 h 25 min – 7 h 56 min
		PADCore	3 min 10s	9.5s	**3 min 20s**
Pseudo	128	FakeIDet2	–	–	1 h 5 min – 2 h 44 min
		PADCore	41.8s	5.5s	**47.3s**

it into bonafide. This is compounded by the fact that FakeIDet2-db print attacks are produced on glossy laminated paper, whose texture and reflective properties differ substantially from the matte gray prints present in DLC-2021, making the available print coreset a poor proxy even for these attacks. Screen attacks on DLC-2021 also prove difficult ($27.61 \pm 4.21\%$), which we attribute to the different recapture conditions: FakeIDet2-db screen attacks are produced using a high-quality XDR display, whereas DLC-2021 screen attacks involve lower-quality screens whose sub-pixel and Moiré artefacts differ substantially from those seen during training, a distribution shift that the frozen coreset features cannot compensate for.

5.5. Computational Efficiency

Table 5 reports training times for PADCore and FakeIDet2 [9] across all 4 publicly available FakeIDet2-db [9] configurations. FakeIDet2 was trained on an NVIDIA RTX 3080 (10GB), and PADCore on an NVIDIA RTX 4090 (24GB). Even accounting for the hardware difference, the gap in training time is not marginal but structural. FakeIDet2's training time spans a wide range due to early stopping, between 1 hour 5 minutes and 7 hours 56 minutes depending on configuration and convergence, whereas PADCore completes in under 4 minutes across all configurations, with the most patch-dense setting (pseudo-anonymized, 64×64) requiring only 3 minutes 20 seconds in total. The dominant cost in PADCore is feature extraction, which is a one-time forward pass through the frozen backbone; the subsequent coreset construction, standardization, and LDA fitting together contribute under 10 seconds in every configuration. Notably, these timings were obtained with a coreset of 10,000 points per class, i.e., five times larger than the 2,000 we found sufficient in Section 5.2, demonstrating that even with redundancy deliberately introduced into the coreset, PADCore remains orders of magnitude faster to train than FakeIDet2.

6. Conclusion

Recent advances in identity document presentation attack detection (PAD) have largely focused on learning task-specific representations through fine-tuning and specialized architectures. In this work, we revisited this design space from the perspective of feature reuse and showed that strong performance can be achieved by directly modelling the structure of frozen patch-level representations extracted from self-supervised vision foundation models. Without adapting the backbone, PADCore matches or surpasses FakeIDet2 [9] in in-distribution evaluation and demonstrates strong generalization to unseen document types in KID34K [12], providing a competitive alternative to more complex learned pipelines across evaluation settings.

Beyond accuracy, PADCore offers substantial practical advantages. Training completes in under four minutes even in the most demanding configuration, compared to several hours for FakeIDet2, and the model can be efficiently re-fitted as new attack data becomes available. Its patch-based formulation is also naturally compatible with privacy-preserving settings, as it operates exclusively on pre-extracted anonymized patches without requiring access to full document images.

At the same time, our results highlight an inherent limitation of the coreset-based formulation. When an attack type is entirely absent during training, the model lacks a corresponding prototype in feature space, which can lead to failure modes such as the collapse of unseen attacks into the bonafide class. This behavior, observed in the leave-one-attack-out print scenario and in gray-print attacks on DLC-2021 [13], suggests that the diversity of the coreset plays a central role in determining generalization performance.

Overall, our findings indicate that, for identity document PAD, the representational capacity of modern foundation models may already be sufficient, shifting the focus toward effective modelling of feature space structure. A promising direction for future work is to extend this framework toward open-set detection, where previously unseen attack types can be identified without explicit supervision.

7. Acknowledgement

Supported in parts by the ARIS Research Programmes P2–0250 (B) and P2-0214 (B) as well as the ARIS project J2-50065 (A) DeepFake DAD.

References

[1] K. Bulatov, E. Emelianova, D. Tropin, N. Skoryukina, Y. Chernyshova, A. Sheshkus, S. Usilin, Z. Ming, J.-C. Burie, M. Luqman, and V. Arlazarov. Midv-2020: a comprehensive benchmark dataset for identity document analysis. *Computer Optics*, 46:252–270, 03 2022.

[2] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev. Reproducible scaling laws for contrastive language-image learning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829, 2022.

[3] S. Damm, M. Laszkiewicz, J. Lederer, and A. Fischer. AnomalyDINO: Boosting Patch-based Few-Shot Anomaly Detection with DINOv2 . In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1319–1329, Los Alamitos, CA, USA, Mar. 2025. IEEE Computer Society.

[4] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4685–4694, 2019.

[5] T. Dorszewski, L. Tětková, R. Jenssen, L. K. Hansen, and K. K. Wickstrøm. From colors to classes: Emergence of concepts in vision transformers. *arXiv preprint arXiv:2503.24071*, 2025.

[6] S. Gonzalez and J. E. Tapia. Forged presentation attack detection for id cards on remote verification systems. *Pattern Recognition*, 162:111352, 2025.

[7] M. Kim, A. K. Jain, and X. Liu. Adaface: Quality adaptive margin for face recognition. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18729–18738, 2022.

[8] R. Mudgalgundurao, P. Schuch, K. Raja, R. Ramachandra, and N. Damer. Pixel-wise supervision for presentation attack detection on identity document cards. *IET Biometrics*, 11(5):383–395, 2022.

[9] J. Muñoz-Haro, R. Tolosana, J. Fierrez, R. Vera-Rodriguez, and A. Morales. Privacy-aware detection of fake identity documents: methodology, benchmark, and improved algorithms (fakeidet2). *Information Fusion*, 128:103969, 2026.

[10] J. Muñoz-Haro, R. Tolosana, R. Vera-Rodriguez, A. Morales, and J. Fierrez. Fakeidet: Exploring patches for privacy-preserving fake id detection. In *2025 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–9, 2025.

[11] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. Featured Certification.

[12] E.-J. Park, S.-Y. Back, J. Kim, and S. S. Woo. Kid34k: A dataset for online identity card fraud detection. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 5381–5385, New York, NY, USA, 2023. Association for Computing Machinery.

[13] D. V. Polevoy, I. V. Sigareva, D. M. Ershova, V. V. Arlazarov, D. P. Nikolaev, Z. Ming, M. M. Luqman, and J.-C. Burie. Document liveness challenge dataset (dlc-2021). *Journal of Imaging*, 8(7), 2022.

[14] K. Roth, L. Pemula, J. Zepeda, B. Scholkopf, T. Brox, and P. Gehler. Towards total recall in industrial anomaly detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14298–14308, 2021.

[15] E. M. Ruiz, J. E. Tapia, R. T. Soto, and C. Busch. Identity card presentation attack detection: A systematic review, 2025.

[16] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, and P. Bojanowski. Dinov3, 2025.

[17] J. E. Tapia, M. Nieto, J. M. Espin, A. S. Rocamora, J. Barrachina, N. Damer, C. Busch, M. Ivanovska, L. Todorov, R. Khizbullin, L. Lazarevich, A. Grishin, D. Schulz, S. Gonzalez, A. Mohammadi, K. Kotwal, S. Marcel, R. Mudgalgundurao, K. Raja, P. Schuch, S. Patwardhan, R. Ramachandra, P. C. Pereira, J. R. Pinto, M. Xavier, A. Valenzuela, R. Lara, B. Batagelj, M. Peterlin, P. Peer, A. Muhammed, D. Nunes, and N. Goncalves. Second competition on presentation attack detection on ID card. In *2025 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10, Osaka, Japan, 2025.

[18] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien. Linear discriminant analysis: A detailed tutorial. *Ai Communications*, 30:169–190,, 05 2017.

[19] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.