
This is the **published version** of the bachelor thesis:

Sellart Sánchez, Ares. *Adversarial Stamps for Document Understanding*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Grau en Intel·ligència Artificial)

This version is available at <https://ddd.uab.cat/record/336314>

under the terms of the  license.

Adversarial Stamps Generation for Image Classification

Ares Sellart

June 28, 2026

Abstract

Deep learning models for image classification are highly vulnerable to adversarial perturbations, raising concerns about their robustness in real-world applications. While most attacks rely on global noise, localized perturbations such as adversarial patches provide a more practical and reusable alternative.

This work proposes a generative approach to adversarial patch construction using a Variational Autoencoder (VAE), modeling stamp-like perturbations within a structured latent space. The method integrates generated stamps into input images and optimizes them through backpropagation of the classification loss to induce misclassification, while preserving visual plausibility.

The proposed framework aims to produce structured and effective adversarial perturbations, offering insight into the trade-off between imperceptibility and attack strength, and highlighting vulnerabilities in modern image classification systems.

Keywords: adversarial attacks, variational autoencoder, image classification, adversarial patches, deep learning, perturbations

1 INTRODUCTION

In recent years, deep learning models have achieved remarkable performance in image classification tasks. However, despite their high accuracy, these models are vulnerable to carefully designed perturbations known as adversarial attacks. These attacks consist of small, often imperceptible modifications to the input data that can lead a model to produce incorrect predictions.

This vulnerability raises important concerns regarding the reliability and robustness of machine learning systems, especially in critical applications such as autonomous driving, medical diagnosis, and security systems.

Among the different types of adversarial attacks, a particularly interesting scenario involves the use of localized perturbations, such as small patches or “stamps”, that are placed on an image to intentionally mislead a classifier. Unlike global noise-based attacks, these perturbations are spatially constrained and, when optimized across a diverse set of input images simultaneously, can be reused across different inputs, making them more practical in real-world situations.

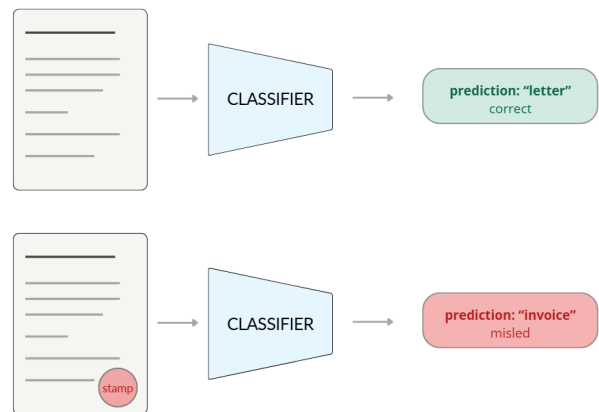


Fig. 1: Conceptual illustration of the attack: the same document, passed through the same classifier, is correctly classified when unmodified but misclassified once a small adversarial stamp is added.

This project focuses on the generation of such adversarial stamps. Specifically, a Variational Autoencoder (VAE) will be employed to learn a structured representation of stamp-like patterns. By constraining the generated perturbations to remain visually similar to natural patterns while optimizing their adversarial effect, the goal is to produce effective yet subtle attacks against a given image classification model.

The study aims to explore the balance between visual imperceptibility and adversarial effectiveness, contributing to a better understanding of both the limitations of current

- Contact E-mail: ares.sellart@autonoma.cat
- Supervised by: Dimosthenis Karatzas (Escola Enginyeria UAB)
- Academic Year 2025/26

classifiers and the potential of models in adversarial settings.

1.1 Objectives

The main objective of this project is to develop a method for generating adversarial perturbations in the form of localized stamps capable of misleading an image classification model. To achieve this goal, the following specific objectives are defined:

- To study and understand the fundamentals of adversarial attacks in deep learning, with a focus on image classification.
- To implement and analyze a Variational Autoencoder (VAE) capable of learning a latent representation of stamp-like patterns.
- To design a loss function that includes both a reconstruction term and a constraint (L2 loss) to ensure that generated stamps remain visually similar to the original inputs.
- To integrate the generated stamps into a classification pipeline by placing them at a fixed position in the input images.
- To evaluate the effectiveness of the adversarial stamps in causing targeted misclassification.
- To analyze the trade-off between visual similarity (imperceptibility) and adversarial strength.

The final goal is to demonstrate that generative models can be used to produce structured and reusable adversarial perturbations, highlighting potential vulnerabilities in deep learning systems.

2 STATE OF THE ART

Deep learning models, particularly convolutional neural networks (CNNs), have achieved state-of-the-art performance in image classification tasks. However, their vulnerability to adversarial attacks has become a major research topic in recent years.

Adversarial attacks were first systematically studied by Goodfellow et al. [1], who demonstrated that adding small perturbations to input images can cause neural networks to misclassify them with high confidence, introducing the Fast Gradient Sign Method (FGSM). More advanced methods followed, such as Projected Gradient Descent (PGD) and the Carlini & Wagner attack [2], which aim to produce minimal yet highly effective adversarial noise. These methods typically generate global perturbations distributed across the entire image.

In contrast, localized adversarial attacks focus on modifying only a specific region of the image. The adversarial patch introduced by Brown et al. [3] demonstrated that a small, visible patch placed in an image can consistently fool classifiers regardless of surrounding content, and can be physically implemented and reused across multiple inputs.

More recent research has explored generative models — GANs and VAEs — to learn structured perturbations that lie within the data distribution, making them less detectable and more realistic. The Variational Autoencoder, introduced by Kingma and Welling [4], combines an encoder-decoder architecture with a latent space regularized through a Kullback–Leibler divergence term, making it particularly suitable for generating smooth variations of input data such as constrained adversarial stamps.

Despite this progress, most existing approaches either generate global perturbations that lack physical plausibility or localized patches that are not constrained to remain visually natural [3, 5], and limited work has explored structured generative models to produce controlled, reusable adversarial perturbations. This gap motivates the approach proposed in this project, which combines localized adversarial attacks with the representational capabilities of generative models.

3 METHODOLOGY

The methodology of this project is structured into several stages, combining generative modeling and adversarial evaluation.

3.1 Dataset Description

Two datasets are used in this project.

For training the generative model, the SPODS (Stamp on Document Segmentation) dataset is used. It contains document images with various types of stamps, exhibiting variability in shape, texture, and appearance (text stamps, circular stamps, logo stamps, and stamps with fine handwritten details). The original SPODS dataset was previously processed using an automated cropping tool to isolate individual stamp instances from the full document images, removing most of the surrounding document content. The resulting cropped stamps were further converted to grayscale format and resized to 128×128 pixels as part of this project, producing the final dataset of isolated grayscale stamp images used for training the generative model.



Fig. 2: Examples of real stamps from the SPODS dataset after preprocessing (grayscale, 128×128), illustrating the variability in shape and texture used to train the generative model.

For the adversarial attack evaluation, the RVL-CDIP (Ryerson Vision Lab Complex Document Information Processing) dataset [?] is used. It consists of 400,000 grayscale document images across 16 classes: letter, memo, email, filefolder, form, handwritten, invoice, advertisement, budget, news article, presentation, scientific publication, questionnaire, resume, scientific report, and specification. From this dataset, a subset of 7,038 images was selected by filtering only those classified by the target model with greater than 95% confidence, ensuring a clean baseline for the attack evaluation.

3.2 Target Classifier

The target classifier is a ResNet-50 model fine-tuned on the RVL-CDIP dataset which achieves high accuracy across all 16 document classes. The model follows the standard ResNet-50 architecture with a custom classification head consisting of two fully connected layers with ReLU activation and dropout, outputting 16 class logits. The classifier is kept completely frozen throughout all experiments (its weights are never updated). It serves solely as the attack target.

3.3 Generative Model

The generative model is responsible for learning a structured latent representation of stamp images that supports realistic random sampling from the prior $\mathcal{N}(0, \mathbf{I})$. Three variants were developed progressively, each addressing limitations identified in the previous one.

The first variant is a standard VAE trained with low KL weight ($\lambda_{KL} = 0.1$). With low KL regularization, the encoder maps each stamp to arbitrary regions of latent space without being constrained to follow $\mathcal{N}(0, \mathbf{I})$, so random samples land in regions the decoder has never seen during training.

To address this, the model was retrained with high KL weight ($\lambda_{KL} = 1.0$) and residual blocks added after each convolutional layer in both encoder and decoder, providing additional capacity and easier gradient flow. This realigns the latent space with the prior, but pixel-wise reconstruction loss alone is subject to spectral bias: when uncertain about fine details such as thin ink strokes or crisp borders, the decoder defaults to blurry average values rather than committing to sharp outputs.

The final variant adds a discriminator network to create a VAE-GAN. Unlike pixel-wise loss, the discriminator evaluates stamp realism holistically — it learns from real stamps that authentic ones have sharp edges, crisp text, and high contrast, and flags blurry outputs as fake, directly counteracting the decoder’s spectral bias. The total training objective becomes:

$$L_{\text{total}} = L_{\text{recon}} + \lambda_{KL} \cdot L_{KL} + \lambda_{\text{adv}} \cdot L_{\text{adv}} \quad (1)$$

Several stability measures were required: the discriminator learning rate was set 100 times lower than the VAE (10^{-6} vs 10^{-4}), updated only every 5 batches, and $\lambda_{\text{adv}} = 0.05$. The model was trained for 300 epochs followed by 200 fine-tuning epochs. Results are reported in Section 4.

3.4 Adversarial Attack Pipeline

The first attack pipeline optimizes a latent vector z such that its decoded stamp, when pasted onto any document, causes the frozen ResNet-50 to predict a specific target class. For each of the 16 target classes, one adversarial stamp is found that fools the classifier on documents from any of the other 15 classes.

The total loss combines an attack term and a regularization term:

$$L_{\text{total}} = L_{\text{attack}} + \lambda \cdot L_{\text{gen}} \quad (2)$$

The attack loss is the cross entropy between the classifier logits and the target class. The generation loss is the L2

norm of z , keeping it close to the origin of the latent space and preventing the decoded stamp from drifting outside the realistic stamp distribution. The weight $\lambda = 0.1$ balances the two objectives.

At each iteration, z is decoded through the frozen VAE-GAN decoder into a 128×128 grayscale stamp, resized to 64×64 via bilinear interpolation, converted to 3-channel RGB by channel replication, and normalized with ImageNet statistics to match the document preprocessing. The stamp is pasted onto a randomly sampled batch of 128 documents from the other 15 classes (Section 3.4.1). The patched documents are passed through frozen ResNet-50, the loss is computed, and gradients are backpropagated to z only — neither the VAE-GAN nor ResNet weights are ever updated. This is repeated for 2000 iterations with Adam optimizer at learning rate 0.05.

Two z initialization strategies were compared: encoding a real stamp from SPODS, which starts the optimization from a meaningful region of the latent space, and sampling z randomly from $\mathcal{N}(0, \mathbf{I})$, which is only viable with the VAE-GAN since the low-KL VAE has a misaligned latent space where random samples produce noise. Both strategies are evaluated in Section 4.

3.4.1 Stamp integration into document images

Once the adversarial stamp is generated by the decoder, it must be integrated into the document image in a way that preserves the gradient flow necessary for backpropagation. The stamp is resized using bilinear interpolation — a differentiable operation that preserves the connection between the stamp pixels and the latent vector z in PyTorch’s computation graph — converted from single-channel grayscale to a 3-channel representation by replicating values across channels, and normalized using ImageNet mean and standard deviation statistics.

The stamp is then pasted onto the document through direct pixel replacement. The document tensor is first cloned to avoid in-place modifications that would break PyTorch’s autograd mechanism, and the target region is overwritten with the normalized stamp values with no blending. In the z -optimization pipeline, the position was fixed at the bottom-right corner across all experiments to isolate the effect of stamp appearance from its spatial location; this constraint is later revisited in the end-to-end architecture (Section 3.5).

3.5 End-to-End Architecture

The z -optimization pipeline has two structural limitations: each optimization run is class- and stamp-specific, so a different stamp for the same class requires a full rerun from a new initialization, and the decoder’s representational capacity is never exploited by the attack itself, since it remains frozen throughout and acts purely as a static shape constraint.

The proposed end-to-end architecture addresses both limitations by making the decoder itself the optimization target rather than z . Instead of searching for one optimal z per class, the decoder weights are trained directly to map any $z \sim \mathcal{N}(0, \mathbf{I})$ to a stamp that fools ResNet into predicting the target class.

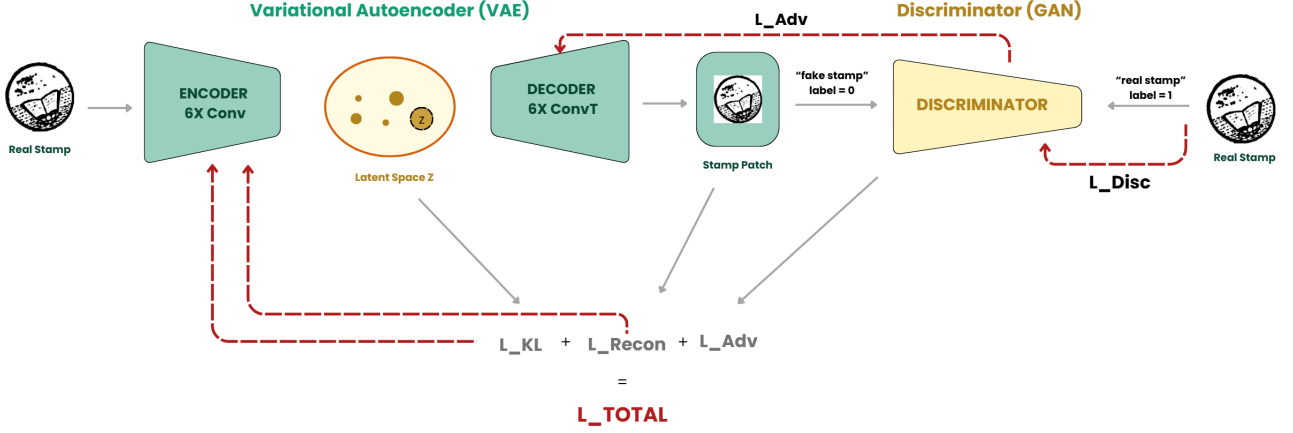


Fig. 3: VAE-GAN architecture (Phase 1). The encoder maps a real stamp to a latent vector z , regularized toward $\mathcal{N}(0, \mathbf{I})$ via L_{KL} . The decoder reconstructs the stamp from z (L_{recon}), while a discriminator provides an adversarial realism signal (L_{adv}) on both reconstructed and randomly-sampled outputs.

The architecture consists of three components. A set of 16 generators $G_0, \dots, G_{15}$, one per document class, each initialized from the pretrained VAE-GAN decoder weights and trained independently to map random z to a stamp causing ResNet to predict its target class; no class-conditioning mechanism is needed, since class specificity is encoded directly in each generator’s weights. A discriminator D , initialized from the pretrained VAE-GAN discriminator and kept completely frozen throughout training, serving as a fixed realism evaluator that provides gradient signal without being updated itself. The frozen ResNet-50 classifier, unchanged from the z -optimization pipeline, providing the attack gradient signal.

The training objective combines two losses:

$$L_{total} = L_{attack} + \lambda \cdot L_{realism} \quad (3)$$

where L_{attack} is the cross entropy between ResNet logits and the target class, and $L_{realism} = \text{BCE}(D(\text{stamp}), 1)$. No reconstruction loss is used, since there is no specific stamp to reconstruct — the generator is free to produce any output satisfying both constraints. The weight $\lambda = 0.05$ matches the adversarial loss weight used in Phase 1.

For each target class c , G_c is trained for 2000 iterations with batch size 128. At each iteration, a random z is sampled and decoded into a stamp, pasted onto a batch of 128 documents from the other 15 classes (using the procedure of Section 3.4.1), and passed through frozen ResNet; the loss is backpropagated to update only G_c .

3.6 Evaluation Protocol

The evaluation measures the attack success rate (ASR) of each stamp independently. For a given target class c , the ASR is defined as the fraction of evaluation documents predicted as c by the frozen ResNet-50 after the stamp is applied:

$$\text{ASR}(c) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}[\hat{y}_i = c] \quad (4)$$

where N is the total number of evaluation documents and $\hat{y}_i$ is the predicted class for document i after stamp appli-

cation. For each target class, evaluation is performed on documents from the other 15 classes.

The evaluation set was constructed according to two criteria. First, documents must be correctly classified by the frozen target model with high confidence, ensuring that any misclassification observed after the stamp is applied can be attributed to the adversarial perturbation itself rather than to pre-existing model uncertainty. Second, the evaluation set must be balanced across the 16 classes, so that no class dominates the reported results and ASR values are directly comparable across classes.

Applying the first criterion to the full RVL-CDIP test set, documents were filtered to retain only those classified by the target model with greater than 95% confidence, producing a clean subset of 7,038 images. Applying the second criterion to this subset, exactly 134 documents per class are used for evaluation — the maximum number that can be drawn equally from all 16 classes, since this is the number of correctly classified images available for the smallest class, scientific report. Using the same number of documents per class prevents classes with more available images from dominating the evaluation.

Generalization. All evaluation documents were never seen during attack optimization, so the reported ASR measures genuine generalization rather than overfitting.

Baseline. The random chance baseline is 6.25% (1/16 classes).

4 EXPERIMENTAL RESULTS

A note on evaluation. Throughout this section, attack effectiveness is reported via the attack success rate (ASR), as defined in Section 3.6. It is important to note that ASR measures only whether the classifier was fooled, and does not by itself capture whether the resulting stamp remains visually or semantically plausible. As shown later in this section, a high ASR can be achieved at the cost of perceptual stamp quality, and the two are not automatically aligned.

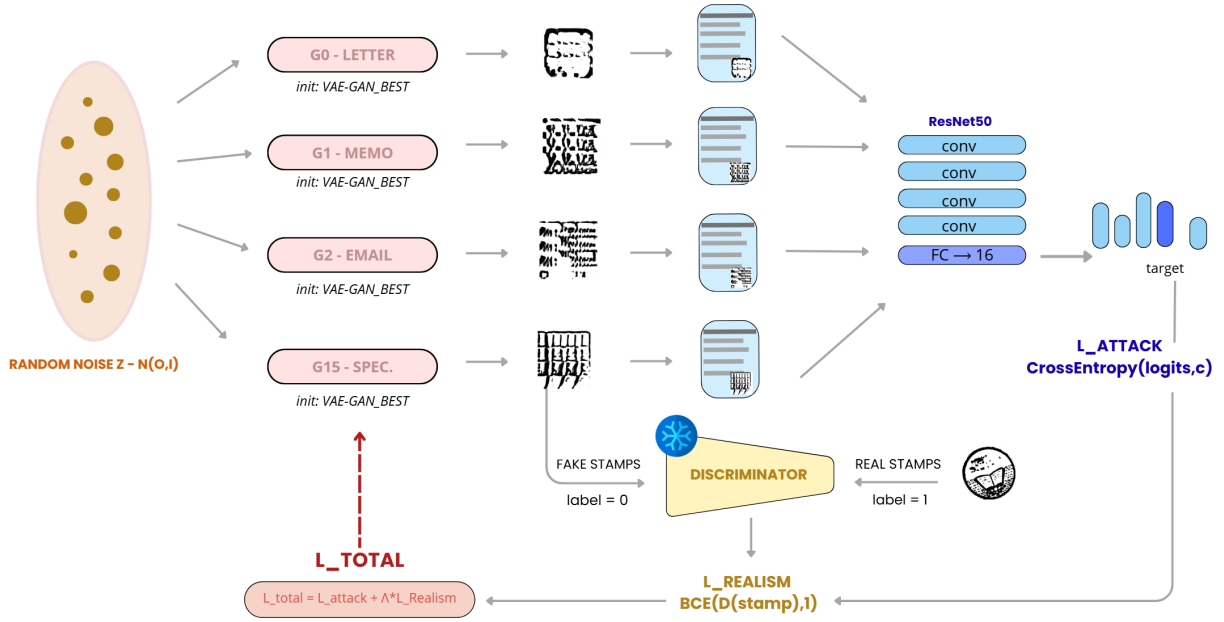


Fig. 4: End-to-end attack architecture (Phase 2). A generator G_c , initialized from the Phase 1 decoder, maps random $z \sim \mathcal{N}(0, \mathbf{I})$ to a stamp. The stamp is pasted onto a document batch and passed through the frozen ResNet-50 (L_{attack}) and the frozen discriminator (L_{realism}). Gradients update only G_c .

High-KL VAE + ResBlocks — latent interpolation

High KL VAE — latent interpolation between two random stamps



VAE-GAN — latent interpolation

VAE-GAN — latent interpolation



Fig. 5: Latent space interpolation between two stamps for the High-KL VAE with residual blocks (top) and the VAE-GAN (bottom). Each row shows the decoded output along a linear path between two latent vectors. The High-KL VAE produces a smooth but consistently noisy transition, while the VAE-GAN interpolation is visibly sharper at every step, with more defined edges and higher contrast.

4.1 Generative Model Results

Three generative model variants were compared on reconstruction quality and random sampling quality from $\mathcal{N}(0, \mathbf{I})$ (Figure 5). The low-KL VAE reconstructs real stamps with excellent fidelity, but random sampling fails completely: all samples collapse to the same blurry average, since the latent space occupied by real stamps does not overlap with the prior. Raising the KL weight and adding residual blocks corrects this misalignment, producing organized and diverse random samples, but they remain noisy due to spectral bias. The VAE-GAN resolves both issues simultaneously: its discriminator loss converges to approximately 0.5 (the theoretical equilibrium), and random samples show sharp, well-defined, diverse stamps consistent with real SPODS examples. This confirms the VAE-GAN as the only variant whose latent space is both correctly aligned with the prior and capable of producing high-quality outputs, and it is used as the generative backbone for all subsequent attack

experiments.

4.2 Z-Optimization Attack Results

The z-optimization attack was evaluated across four configurations isolating compute budget, generative model architecture, and z initialization strategy. All results use 134 documents per source class against the 6.25% random chance baseline.

Method 1a (low-KL VAE, encoded stamp, CPU, 300 iterations, batch 8) serves as the baseline, reaching 69.6% overall ASR, with a strong positive correlation between class confidence and ASR. Method 1b uses the same configuration with 2000 iterations and batch 128 on GPU, improving to 88.6% — confirming the optimization process itself was a bottleneck in the baseline.

Method 2 isolates the generative model by switching to the VAE-GAN under the same GPU conditions, reaching 84.6% — slightly below Method 1b, likely because the

TABLE 1: Attack success rate (%) per class and method. All results evaluated on 134 documents per source class. Random chance baseline: 6.25%.

Class	Method 1a	Method 1b	Method 2	Method 3
	VAE low-KL, enc., CPU	VAE low-KL, enc., GPU	VAE-GAN, enc., GPU	VAE-GAN, rnd z , GPU
letter	65.2	85.8	83.8	52.7
memo	66.4	92.8	80.0	80.5
email	64.4	68.6	71.1	50.0
filefolder	69.8	96.0	92.4	93.9
form	97.5	98.6	98.9	98.9
handwritten	60.3	97.8	94.7	60.5
invoice	70.2	99.0	91.4	62.4
advertisement	65.0	94.3	96.2	86.9
budget	94.9	96.0	95.3	61.0
news article	56.8	97.2	78.3	97.9
presentation	94.6	98.8	64.2	97.4
sci. publication	82.8	98.5	96.2	94.1
questionnaire	60.2	89.5	87.1	93.0
resume	80.4	89.6	89.2	75.8
sci. report	0.9	17.4	16.5	93.8
specification	84.2	97.2	58.9	85.2
Overall	69.6	88.6	84.6	80.5

VAE-GAN’s stronger KL regularization compresses stamp encodings into a denser latent region, making clean adversarial patterns harder to find.

Method 3 isolates the z initialization strategy by sampling randomly from $\mathcal{N}(0, \mathbf{I})$ instead of encoding a real stamp, reaching 80.5% overall but revealing a fundamentally different attack mechanism: scientific report improves dramatically (16.5% $\rightarrow$ 93.8%) while several otherwise-easy classes degrade. This divergence, together with the Pearson correlation dropping to $r = -0.31$ (from a strong positive correlation with encoded initialization), indicates that random- z attacks succeed by overwhelming the classifier’s decision rather than specifically activating target-class neurons.

Qualitatively, all generated stamps are dark, high-contrast and ink-textured, since such patterns produce the strongest gradient signal. Encoded-stamp initialization preserves a recognizable stamp-like appearance throughout optimization, while random- z stamps look more abstract and aggressive, though still constrained to the stamp distribution by the decoder. In successful cases, ResNet’s confidence for the target class shifts from near 0% before stamp application to near 100% after; encoded-stamp initialization tends to produce higher-confidence successful predictions than random z , consistent with the interpretation that encoded stamp finds a strong class-specific signal while random z wins by confusing the classifier rather than convincing it.

4.3 End-to-End Attack Results

4.3.1 Main Results

The end-to-end model was evaluated using 16 generators trained with a fixed bottom-right stamp position, each tested on documents from all 15 other classes using 5 random $z \sim \mathcal{N}(0, \mathbf{I})$ samples, averaged per class. The model

achieves an overall ASR of 97.7% (Table 2), a substantial improvement over all z -optimization methods. Performance is consistently high across almost all classes, with 14 out of 16 exceeding 97% ASR; the only notable exception is scientific report at 80.5%, consistent with the difficulty of this class observed across all methods.

TABLE 2: End-to-end attack success rate (%) per target class, fixed bottom-right position. Results averaged over 5 random $z \sim \mathcal{N}(0, \mathbf{I})$ samples per class. Evaluated on 134 documents per source class.

Target class	ASR (%)
letter	98.1
memo	97.4
email	99.7
filefolder	99.4
form	99.6
handwritten	98.1
invoice	97.1
advertisement	99.0
budget	97.9
news article	99.5
presentation	99.6
sci. publication	98.9
questionnaire	99.2
resume	99.9
sci. report	80.5
specification	99.0
Overall	97.7

4.3.2 Position Ablation

Two ablation studies tested the role of stamp position. In the first, fixed bottom-right generators were evaluated at 6 different positions; in the second, a new set of 16 generators

was trained with a random position at every iteration and evaluated at the same 6 positions (Table 3).

Fixed-position generators show strong dependence on the training position, falling from 97.7% at bottom-right to 84.6% at bottom-left, 82.1% at top-right, and as low as 63.9% at random positions — these generators learn position-specific adversarial features that do not transfer to other locations, likely because ResNet has learned position-sensitive representations from real documents. Random-position generators show markedly more uniform behavior across all positions (86.5–97.5%), with center achieving the highest ASR; their average ASR across the four corners (88.1%) exceeds that of fixed-position generators evaluated at non-training corners (82.1%), confirming that random-position training produces more robust, position-invariant patterns. The random-position model is adopted as the final model, reflecting a realistic deployment scenario where the attacker cannot control exact stamp placement.

TABLE 3: Position ablation — overall ASR (%) for fixed-position and random-position generators evaluated at different stamp positions.

Position	Fixed	Random
Bottom-right (train pos.)	97.7	88.6
Bottom-left	84.6	87.5
Top-right	82.1	86.5
Center	76.7	97.5
Top-left	64.0	89.8
Random	63.9	95.2

4.3.3 Size Ablation

Fixed bottom-right generators were evaluated with stamp sizes of 16×16 , 32×32 , 64×64 (training size), and 96×96 pixels, resizing the stamp to each target size before pasting while keeping generators unchanged (Table 4). Results show a sharp optimum at the training size, with performance collapsing at smaller sizes (1.0% at 16×16 , 2.6% at 32×32) as the stamp carries too few pixels of adversarial signal, and degrading at 96×96 (57.4%) as the oversized

stamp overwrites contextual document features the generator relies on. This ablation evaluates generators trained at 64×64 and resized at test time; a fully fair comparison would require retraining at each size, left as future work.

TABLE 4: Size ablation — overall ASR (%) for fixed bottom-right generators evaluated at different stamp sizes.

Patch size	Overall ASR (%)
16×16	1.0
32×32	2.6
64×64 (training size)	97.1
96×96	57.4

4.3.4 Architecture Ablation: VAE-GAN vs Plain GAN

To assess whether VAE-GAN pretraining is necessary, a plain GAN baseline was trained using only the decoder component of the VAE architecture — no encoder, no KL loss, no reconstruction loss — with an equivalent discriminator. Stability required raising the discriminator learning rate to 5×10^{-5} , updating it every 3 batches instead of 5, and applying label smoothing (0.9 instead of 1.0 for real labels); with these adjustments, training converged to a stable equilibrium near the theoretical Nash equilibrium of $\log(2) \approx 0.693$.

The plain GAN generator was used to initialize 16 generators for Phase 2 under the same random-position setup as the final model, achieving 83.8% overall ASR compared to 95.2% for VAE-GAN initialization — an 11.4 percentage point gap (Table 5). The degradation is concentrated in specific classes rather than spread uniformly: resume (45.8% vs 99.8%) and advertisement (47.7% vs 99.7%) collapse dramatically, while classes such as handwritten, memo, and form show only minor degradation. This indicates that, although the encoder is never used during Phase 2 inference, the VAE-GAN’s joint reconstruction, KL, and adversarial training recipe produces a decoder with weights that are empirically better suited as a Phase 2 initialization than a decoder trained with adversarial loss alone — particularly for

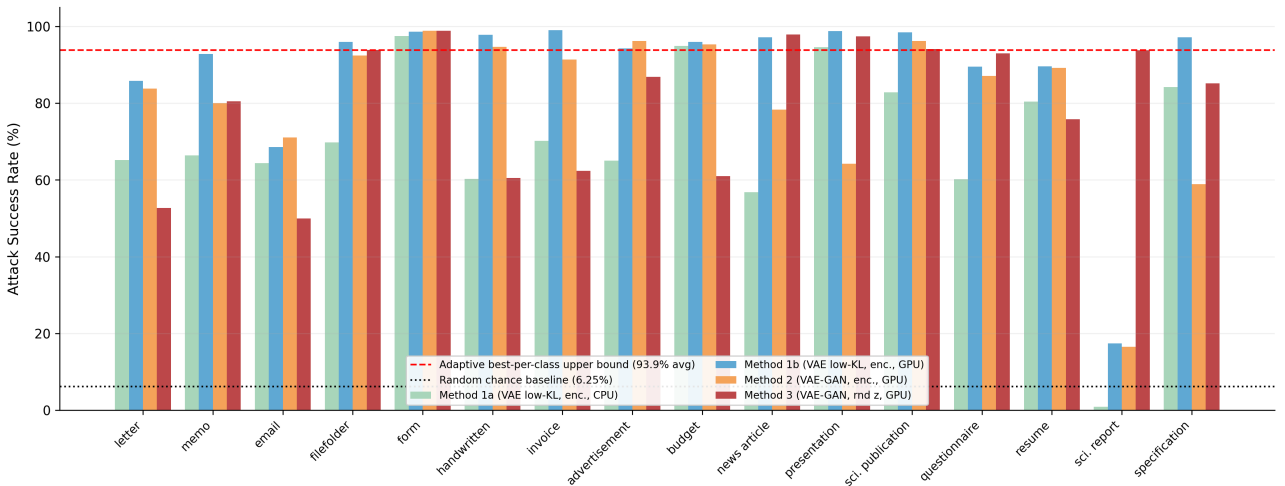


Fig. 6: Attack success rate per class for all four methods. The dashed red line shows the adaptive best-per-class upper bound (93.9% average). The dotted line shows the random chance baseline (6.25%). Results are evaluated on 134 documents per source class.

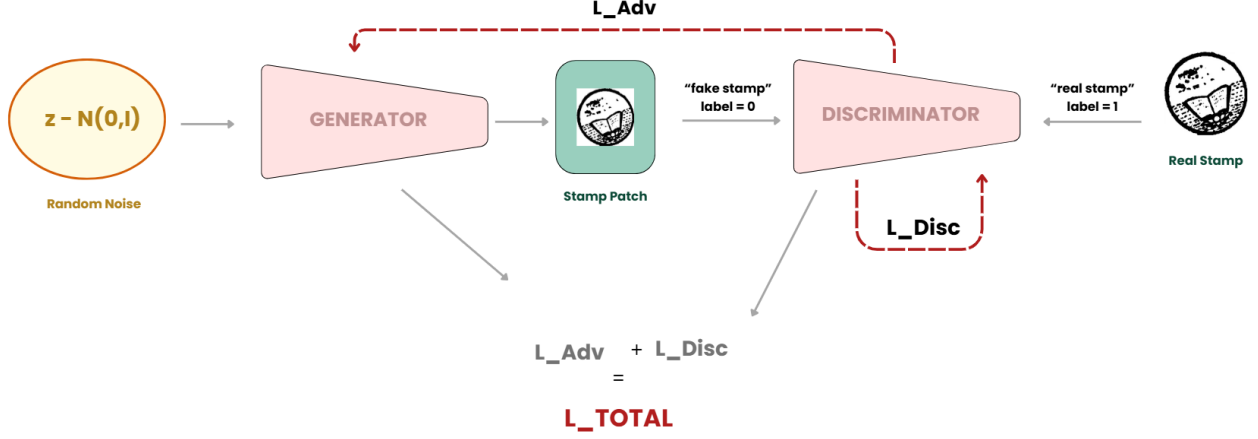


Fig. 7: Plain GAN architecture used in the architecture ablation (Phase 1 alternative). A generator maps random noise $z \sim \mathcal{N}(0, \mathbf{I})$ directly to a stamp, with no encoder, reconstruction, or KL loss involved. The discriminator is trained to distinguish generated stamps from real SPODS stamps, providing an adversarial loss to the generator and a discriminator loss to itself.

structurally distinctive classes where the optimizer is less likely to find an effective solution from a noisier starting point.

TABLE 5: Architecture ablation — ASR (%) per class for the plain GAN baseline, random-position training.

Class	ASR (%)
Letter	95.6
Memo	97.3
Email	87.9
Filefolder	85.7
Form	95.7
Handwritten	98.8
Invoice	86.6
Advertisement	47.7
Budget	92.0
News article	81.4
Presentation	75.7
Sci. publication	81.5
Questionnaire	87.3
Resume	45.8
Sci. report	86.8
Specification	94.5
Overall	83.8

4.3.5 Qualitative Analysis

Generated stamps show high visual quality in low-level image statistics: sharp, high-contrast, free of the blurriness seen in early VAE variants, a direct consequence of the realism loss. However, the stamps no longer resemble recognizable stamp categories such as circular seals or text blocks, instead converging to abstract, texture-like patterns. This loss of semantic content follows from the nature of Phase 2 training: generators are trained exclusively from random z with no reconstruction objective tying outputs to a real reference stamp, and the only realism constraint is a discriminator that evaluates low-level texture statistics with no

mechanism to assess semantic plausibility — leaving the attack loss free to reshape the decoder’s output toward what-ever maximizes ResNet confidence.

A further limitation is mode collapse: different z vectors decoded by the same generator produce nearly identical stamps, since the training objective rewards only attack success with no incentive for diversity. Two mitigation strategies were explored. Increasing the realism loss weight made stamps sharper and more high-contrast — consistent with what the discriminator associates with realism — but did not restore semantic structure, since sharpness was never the missing ingredient. An explicit diversity loss, penalizing pixel-wise similarity between stamps from different z vectors, achieved a measurable increase in diversity, but the generator satisfied it primarily through spatial shifts of the same underlying pattern rather than discovering structurally distinct solutions. These two limitations are revisited and substantially addressed in Section 4.3.6.

4.3.6 Inducing Meaning

The qualitative analysis above revealed that end-to-end generated stamps, while sharp and high-contrast, no longer resemble recognizable stamp categories. This raised the question of how to recover semantic plausibility. Importantly, the framing of this problem required revision: since the end-to-end generators are driven exclusively by random $z \sim \mathcal{N}(0, \mathbf{I})$ with no reconstruction target, there is no specific real stamp being approximated, and therefore no original meaning to *preserve*. The relevant question is instead how to *induce* meaningful, stamp-like structure in an output that is generated from noise.

Morphological operations. The first approach tested a cheap, purely structural intervention: a morphological opening loss, computed as a differentiable approximation of erosion followed by dilation via min- and max-pooling. The intuition is that a solid, coherent shape survives this operation largely unchanged, while a noisy, fragmented pattern loses its small disconnected pieces under erosion and does not recover them under dilation; penalizing the difference

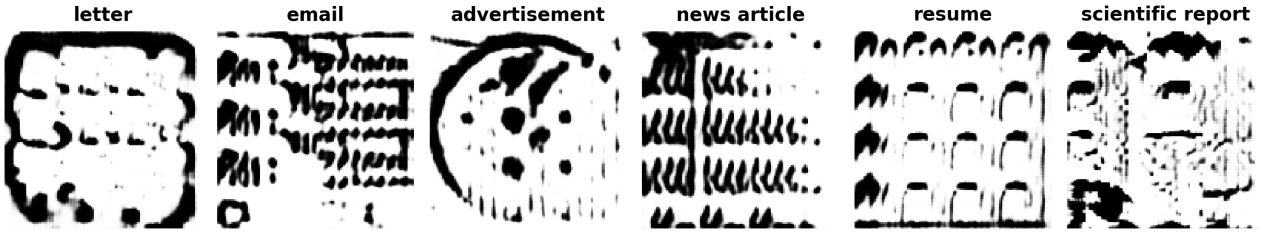


Fig. 8: Example stamps generated by the end-to-end model for six representative target classes, each decoded from a single random $z \sim \mathcal{N}(0, \mathbf{I})$. Stamps are sharp and high-contrast but no longer resemble recognizable stamp categories, instead converging to abstract, texture-like patterns.

between a stamp and its opened version therefore discourages scattered, speckled texture in favor of solid, connected regions:

$$L_{\text{morph}} = \text{mean}(|\text{stamp} - \text{opening}(\text{stamp})|) \quad (5)$$

This loss was added to the existing training objective with weight λ_{morph} , tested on three representative classes (letter, advertisement, resume). Results showed the morphological loss value decreasing steadily over training while ASR remained largely unaffected, indicating that the generator could satisfy both objectives simultaneously rather than trading one off against the other. Qualitatively, the resulting stamps showed visibly more organized, blob-like structure compared to the scattered texture of the baseline — for instance, the letter generator produced a repeating wave-and-blob pattern, and the resume generator produced text-like character fragments alongside a scalloped border. However, increasing the loss weight further (from $\lambda_{\text{morph}} = 1.0$ to 3.0) did not yield a proportionally clearer improvement, and the resulting shapes, while more coherent, still did not resemble specific real-world stamp categories such as circular seals or structured text blocks.

Diagnostic discriminator. To obtain an independent, quantitative measure of the remaining gap, another discriminator was trained from scratch to distinguish real SPODS stamps from the original, unmodified end-to-end outputs (the baseline model, without any structural loss). Unlike the discriminator used during attack training — which only ever learned to separate blurry early-VAE outputs from sharp real stamps, and was never exposed to “sharp but semantically empty” as a category of fake — this new discriminator was trained with the correct labels from the start: real SPODS stamps as real, baseline end-to-end outputs as fake. The idea consists of, rather than retraining a generator from the Phase 1 decoder weights, fine-tuning the already-converged end-to-end generators further, replacing the original discriminator with this new diagnostic discriminator as the realism signal:

$$L_{\text{total}} = L_{\text{attack}} + \lambda_{\text{diag}} \cdot L_{\text{diag}} \quad (6)$$

where $L_{\text{diag}} = \text{BCE}(D_{\text{diag}}(\text{stamp}), 1)$, and the diagnostic discriminator remains frozen throughout. Starting from an already-converged, high-ASR generator rather than from scratch was intended to preserve existing attack effectiveness and output sharpness while testing whether pressure from a discriminator sensitive to the semantic gap could shift the generator toward more genuinely stamp-like structure.

TABLE 6: Diagnostic fine-tuned model — ASR (%) per class, random position.

Class	ASR (%)
Letter	98.2
Memo	95.5
Email	99.2
Filefolder	98.3
Form	99.8
Handwritten	98.4
Invoice	88.9
Advertisement	98.1
Budget	93.7
News article	99.0
Presentation	99.8
Sci. publication	98.8
Questionnaire	98.1
Resume	99.8
Sci. report	83.9
Specification	98.7
Overall	96.8

The full fine-tuning was then scaled to all 16 classes under the same settings, evaluated using the same random-position protocol as the final model (Table 6). The diagnostic fine-tuned model achieves an overall ASR of 96.8% — exceeding the original random-position model (95.2%) by 1.6 percentage points, despite the additional structural constraint. Performance remains consistently high across most classes, with scientific report again the most difficult (83.9%), though notably improved relative to the original random-position model’s 52.3% for this class.

These results indicate that the loss of semantic stamp structure observed in the baseline end-to-end model is not an unavoidable consequence of the attack objective itself, but rather a consequence of the realism signal being too narrow. Once a discriminator capable of detecting the actual semantic gap is introduced, the generator is able to move toward more structurally coherent outputs — in this case visibly more round, seal-like shapes — without sacrificing, and in fact slightly improving, attack effectiveness. This suggests that the diagnostic discriminator approach is a more promising direction than the morphological loss alone for inducing genuine stamp-like structure, and is adopted as the final model going forward.

Future work. Given more time, the results could likely be further improved by tuning the hyperparameters of this

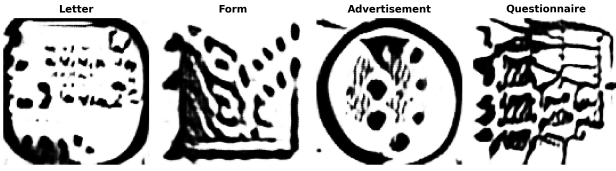


Fig. 9: Example stamps generated by the diagnostic fine-tuned model for four representative target classes, each decoded from a single random $z \sim \mathcal{N}(0, \mathbf{I})$. Compared to the baseline end-to-end stamps (Figure 8), these show visibly more coherent, often round or seal-like structure.

approach — in particular, the weight given to the diagnostic discriminator loss, the number of fine-tuning iterations, and combining it with the morphological loss rather than treating the two as separate alternatives. A more thorough approach could also retrain the diagnostic discriminator iteratively as the generator improves, rather than relying on a single frozen snapshot, which may yield sharper and more consistent results across all classes.

5 DISCUSSION

5.1 End-to-end training outperforms z -optimization, and the apparent trade-off with semantic structure can be resolved

One of the main motivations of this work was to overcome the structural limitations of optimizing a single latent vector against a frozen decoder. The results confirm that training the decoder weights directly substantially improves attack performance: the end-to-end architecture reached up to 97.7% ASR at the training position and 95.2% under random placement, compared to a maximum of 88.6% across all z -optimization variants. This supports the hypothesis that treating the generative model as a trainable component of the attack, rather than a static shape constraint, yields a more effective and reusable attack.

However, this improvement initially came with a notable cost: the end-to-end generators, trained exclusively from random z with no reconstruction objective, gradually lose semantic stamp structure, converging toward abstract, texture-like patterns despite satisfying the discriminator’s realism criterion. This revealed a tension with the proposed objective — a discriminator trained only on low-level texture statistics is not a sufficient proxy for semantic plausibility, leaving the attack loss free to reshape the decoder’s output well beyond what a human observer would recognize as a stamp.

This tension was directly addressed by training a diagnostic discriminator with the correct labels from the start. It confirmed that the original discriminator’s narrow training signal, rather than any fundamental limitation of the attack objective, was responsible for the loss of structure. Fine-tuning the already-converged generators with this discriminator as an additional realism signal recovered visibly more coherent, often round, seal-like structure across all 16 classes, while simultaneously improving overall ASR to 96.8%, slightly above the original random-position model. This indicates that semantic plausibility and attack effectiveness are not inherently in conflict in this setting; the ap-

parent trade-off observed initially was a consequence of an insufficiently informative realism signal, not of the end-to-end training objective itself.

5.2 VAE-GAN pretraining remains valuable even when the encoder is unused

A central design question was whether the full VAE-GAN architecture was justified for the end-to-end attack, given that the encoder is never invoked during Phase 2 training or inference. The architecture ablation directly tested this by comparing VAE-GAN- and plain-GAN-initialized generators, showing a clear advantage for VAE-GAN initialization (95.2% vs 83.8%), with the gap concentrated in classes where the plain GAN failed to converge to an effective solution.

This indicates that, although the encoder itself is architecturally redundant at attack time, the training procedure it enables — joint optimization of reconstruction, KL divergence, and adversarial realism — produces a decoder whose mapping from $\mathcal{N}(0, \mathbf{I})$ to image space is better organized and more reliably fine-tuned than one trained with adversarial loss alone. This distinguishes two separate questions that are easy to conflate: whether a component is used at inference time, and whether the training procedure that produced it remains valuable. Here, the encoder fails the first test but the overall VAE-GAN recipe passes the second.

5.3 Difficulty varies systematically by document class

Across every method evaluated — both z -optimization and end-to-end — scientific report consistently emerged as the hardest target class, with success rates well below average regardless of attack strategy. This persistence across fundamentally different attack mechanisms suggests the difficulty is rooted in the target classifier itself: scientific report has the fewest high-confidence correctly classified images in the filtered subset (134, the minimum across all classes), indicating ResNet-50’s internal representation for this class occupies a small, likely poorly separated region of decision space, plausibly overlapping with the visually similar scientific publication class. A smaller, less robust target region provides a weaker and noisier gradient signal for any adversarial method, regardless of whether it optimizes a single latent vector or trains a full generator.

A related but distinct pattern emerged in the architecture ablation: resume and advertisement showed a severe, isolated drop under plain GAN initialization. Unlike scientific report’s difficulty, which appears across all methods and stems from the classifier’s representation, this drop is specific to the combination of these classes with a particular initialization strategy, suggesting their adversarial optimum is narrower and more dependent on starting from a well-organized latent space — precisely the property the VAE-GAN’s training provides and the plain GAN lacks. Attack difficulty in this setting is therefore not uniform across classes.

6 CONCLUSIONS

This thesis investigated the generation of adversarial stamps using generative models, progressing from a constrained, reconstruction-based attack to a fully trainable end-to-end pipeline. Most of the original objectives were met, though two were revisited along the way: visual similarity to a reference stamp, and fixed-position integration. The end-to-end model gives up pixel-level similarity to a real stamp in exchange for a much stronger attack, and random-position training replaced the fixed position once it proved far more robust.

The resulting trade-off is not quite the one originally intended between imperceptibility and attack strength. It initially appeared closer to a trade-off between attack effectiveness and structural meaning: the end-to-end model reaches up to 97.7% ASR at the training position and 95.2% under random placement, far above the z-optimization baseline, but the generated stamps no longer looked like real stamps. However, this apparent trade-off was largely resolved by introducing a diagnostic discriminator capable of detecting the actual semantic gap, which recovered visibly more coherent, stamp-like structure while further improving ASR to 96.8%. This suggests that attack effectiveness and semantic plausibility are not fundamentally opposed in this setting, but rather depend on designing a sufficiently informative realism signal — a direction with clear room for further refinement, as discussed in Section 4.3.6.

7 ACKNOWLEDGMENTS

Writing this final paragraph is the most beautiful part of completing a thesis. Putting into words a piece of work that owes its shape and meaning to all the people named in the following lines is what motivates me most.

I would like to begin by thanking my supervisor, Dimosthenis Karatzas, for his help and mentorship. Thank you for everything — for the trust you have placed in me and for the path you have helped me create and follow. I admire the work you do and the way you do it. I like to think of your classes or meetings as a bubble: you step in carrying one idea, and somewhere between your questions and your explanations, that idea gets turned inside out, until you step back out holding a different one. You motivate your students to think and understand through your explanations. You are the kind of teacher who reminds you why you chose this path as a student.

I would also like to thank Qi Dong and Lei Kang, who were there from day one, checking in on the project’s progress and helping me see it through.

To my family: my mother Joana, my father Toni, and my aunt Mari, for being by my side throughout the whole process. Thank you for listening to me talk a thousand and one times about stamps and for trying to understand all of my work. Thank you for playing the role of students and turning me into your teacher.

To Martina, for being “my partner in crime”. Thank you for making every day lighter and for sharing so many laughs between coffees or between training runs. Being classmates and lab partners together has been one of the best things this degree has given me.

To Laia, Berta, and Joaquín, for always being there. Thank you for cheering me up when I was tired or when things did not go entirely well, and for celebrating with me when the results were good, sometimes even more excited than I was. I love you.

REFERENCES

- [1] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: <https://arxiv.org/abs/1412.6572>
- [2] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *IEEE Symposium on Security and Privacy*, 2017. [Online]. Available: <https://arxiv.org/abs/1608.04644>
- [3] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, “Adversarial patch,” *arXiv preprint arXiv:1712.09665*, 2017. [Online]. Available: <https://arxiv.org/abs/1712.09665>
- [4] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *International Conference on Learning Representations (ICLR)*, 2014. [Online]. Available: <https://arxiv.org/abs/1312.6114>
- [5] M. Pintore, M. Pintor, D. Karatzas, and B. Biggio, “Counterfeit answers: Adversarial forgery against OCR-free document visual question answering,” *arXiv preprint arXiv:2512.04554*, 2024. [Online]. Available: <https://arxiv.org/abs/2512.04554>