Memorization Detection Benchmark for Generative Image models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Generative models in medical imaging offer significant potential for data augmentation and privacy preservation, but they also pose risks of patient data memorization. This study presents a comprehensive, data-driven approach to evaluate and characterize the memorization behavior of generative models. We systematically compare various network architectures, loss functions, pretraining datasets, and distance metrics to identify optimal configurations for detecting potential privacy concerns in synthetic images. Our analysis reveals that self-supervised contrastive networks using Triplet Margin loss in models like DinoV2, DenseNet121, and ResNet50, when paired with Bray-Curtis or Standardized Euclidean distance metrics, demonstrate superior performance in detecting augmented copies of training images. We further apply our methodology to characterize the memorization behavior of a conditional diffusion image transformer model trained on mammography data. This work contributes a robust framework for evaluating generative models in medical imaging, offering a crucial tool for assessing the risk of patient data leakage in synthetic datasets.[1]

1 Introduction

The advent of generative models has a lot of potential in healthcare and medical imaging initiatives, promising enhanced data sharing, expanded datasets, and improved training data diversity [1]. However, these advancements come with significant privacy implications, especially given the sensitive nature of patient information. A key concern is the phenomenon of model memorization [2, 3], where generative models inadvertently reproduce specific details from their training data, potentially compromising patient confidentiality and undermining the core purpose of synthetic data generation.

Recent research has demonstrated that a wide range of generative models, including GANs, VAEs, and diffusion models, are vulnerable to memorization [4, 5, 6, 7]. Of particular note, diffusion models [8], despite their impressive image quality, have shown a higher propensity for memorization [6]. This finding underscores the intricate interplay between model sophistication, output quality, and data privacy. Furthermore, conventional evaluation metrics such as Inception Score (IS) [9] and Fréchet Inception Distance (FID) [10] fall short in detecting these memorization issues, potentially masking critical privacy vulnerabilities in emerging image generation techniques.

A common misconception is that memorization can be effectively addressed by simply monitoring validation errors and preventing overfitting. However, this approach overlooks the fundamental differences between these two phenomena [11]. While overfitting manifests as a global issue where

[1]The code for this study is available at `https://github.com/molinamarcvdb/ImageFeatureExtractionBenchmark`

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

models excel on training data at the expense of generalization, memorization is a more nuanced problem. It involves the model assigning disproportionately high probabilities to specific training instances. Intriguingly, a model's tendency to memorize can actually increase even as its validation performance improves, particularly during the initial stages of training [11]. This paradoxical relationship highlights the need for specialized strategies to identify and mitigate memorization, distinct from traditional overfitting prevention techniques.

Our research builds upon recent advances in self-supervised contrastive learning for memorization detection [5, 12], offering a comprehensive benchmark. We propose a novel approach to evaluate the efficacy and resilience of self-supervised networks through systematic image augmentations. Our study compares the performance of various state-of-the-art pretrained network architectures, including ResNet50 [13] and DinoV2 [14]. We also investigate the influence of different loss functions, including distance-based and entropy-based formulations, and examine the impact of pretraining on natural versus medical image datasets. By comparing a range of similarity, distance, and information-theoretic metrics, we aim to identify the most sensitive indicators for detecting and characterizing training data memorization. To demonstrate the practical application of our findings, we employ the best-performing method to analyze the memorization patterns in a diffusion model.

2 Related work

2.1 Model Memorization

The phenomenon of model memorization has been extensively studied in machine learning, particularly in supervised learning contexts. Neural networks have demonstrated the capacity to memorize entire datasets, including those with random labels [15]. This memorization is not uniform across all data points; outliers and mislabeled samples are more likely to be memorized [16]. Memorization and generalization might also depend on network architecture and optimization procedure, but also on the data itself [17]. Moreover, some level of memorization in supervised learning has been shown to be important for generalization in several standard benchmarks [18]. In generative models, memorization presents unique challenges, as models that closely replicate training data may still achieve favorable scores on standard quality and diversity metrics. Recent work has demonstrated that GANs, VAEs, and diffusion models as well as vision language models are all susceptible to memorizing training data [4, 5, 6, 7, 19]. Therefore, creating a memorization metric to be monitored during training would enable a more comprehensive assessment of the generative model performance.

2.2 Memorization Detection Methods

Various approaches have been proposed to detect and quantify memorization in generative image models. Correlation-based methods, such as the structural similarity index measure (SSIM) employed by [20, 21, 22], offer a straightforward approach to assessing similarity between generated and training images. However, these methods were initially developed to measure diversity not memorization behaviour, and may be sub-optimal to detect generated samples which are mere augmented versions of the training data (e.g., rotation or flipping).

More sophisticated approaches leverage self-supervised learning and contrastive methods. In [5] the authors introduced a framework that uses contrastive learning to map images to a lower-dimensional embedding space, allowing for the detection of copies that may include rotated or flipped variants of training images. This method was further explored in [12], which investigated the effects of various hyperparameters and training setups on memorization as well as mitigation strategies.

2.3 Mitigation Strategies

Various approaches have been proposed to mitigate memorization in generative models. These include using exclusively augmented images during training [5], implementing Differentially Private Stochastic Gradient Descent (DP-SGD) [23], and applying standard regularization techniques like dropout and weight decay. Additionally, novel methods such as Privacy Distillation have been introduced [24]. This two-step approach involves training an initial diffusion model on real data, generating and refining synthetic samples to exclude identifiable information, and then using these refined samples to train a second model. This method aims to reduce re-identification risk while maintaining downstream performance.

However, these mitigation strategies often involve trade-offs. DP-SGD can compromise image quality or lead to model divergence [25], while data augmentation may complicate similarity assessments between synthetic and original images. The Privacy Distillation approach, while promising, may result in reduced quality of the final synthetic samples. Finally, factors such as over-training, dataset size, and augmentation techniques also significantly influence memorization and should be carefully addressed [5, 6, 12].

3 Methods

3.1 Problem Formulation

Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ represent a set of N training images, $\mathcal{X}_v = \{v_1, \ldots, v_K\}$ denote a set of K validation images, and $\mathcal{G} = \{g_1, \ldots, g_M\}$ be a set of M generated images. We train a Self-Supervised Contrastive Network (SSCN) to learn an embedding function $\phi : \mathcal{I} \to \mathbb{R}^d$, where $\mathcal{I}$ is the image space and d is the embedding dimension, by minimizing a contrastive loss function $L(\phi; \mathcal{X})$.

Given a similarity metric $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, we compute the similarity between training and generated images as $S(x, g) = s(\phi(x), \phi(g))$ for $x \in \mathcal{X}, g \in \mathcal{G}$, and baseline similarities between training and validation images as $S_{base}(x, v) = s(\phi(x), \phi(v))$ for $x \in \mathcal{X}, v \in \mathcal{X}_v$. To prevent memorization of synthetic data, we set a threshold τ as the p-th percentile of the S_{base} distribution.

For evaluation, we define a set of severely augmented images $\mathcal{X}_a = \{a_1, \ldots, a_L\}$, where each a_i is derived from $\mathcal{X}$ using strong augmentations. We monitor the percentage of augmented images that match their corresponding original images in $\mathcal{X}$ according to the similarity threshold τ.

3.2 Self-Supervised Contrastive Network

3.2.1 Architecture

The SSCN comprises a backbone network $f_\theta : \mathcal{I} \to \mathbb{R}^d$, followed by a projection head $g_\phi : \mathbb{R}^d \to \mathbb{R}^k$. The backbone extracts features from the input images, while the projection head maps these features to a lower-dimensional embedding space. The complete network is represented as:

$$h_{\theta,\phi}(x) = g_\phi(f_\theta(x)) \tag{1}$$

We experiment with several backbone architectures, including ResNet50 [13], DenseNet121 [26], Inception V3 [27], CLIP Image Encoder [28] , and DinoV2 [14]. The projection head is a linear layer defined as $g_\phi(z) = Wz + b$, where $W \in \mathbb{R}^{k \times d}$ and $b \in \mathbb{R}^k$.

To explore the impact of domain-specific knowledge, we use backbones pretrained on both natural images (ImageNet [29]) and medical images (RadImageNet [30]). This comparison allows us to evaluate the transfer learning benefits of using medical-domain-specific pretraining.

3.2.2 Loss Functions

To structure the embedding space, we employ and compare two popular contrastive losses: the Triplet Margin Loss [31] and InfoNCE Loss[32]. Both losses aim to pull semantically similar data points closer while pushing dissimilar points farther apart.

Triplet Margin Loss. This loss function ensures that the distance between an anchor-positive pair is smaller than the distance between the anchor-negative pair, with a margin m. Specifically, for an anchor a, a positive example p, and a negative example n, the loss is defined as:

$$L_{\text{triplet}}(a, p, n) = \max(0, m + d(a, p) - d(a, n)) \tag{2}$$

where $d(\cdot, \cdot)$ is the Euclidean distance and m is the margin parameter. This encourages positive pairs to be closer together while keeping negatives farther apart in the embedding space.

InfoNCE Loss. InfoNCE (Information Noise-Contrastive Estimation) compares each anchor representation z_i with one positive sample z_j^+ and $N-1$ negative samples $\{z_j^-\}$, using Cosine similarity between the embeddings. The objective is to maximize the probability that the positive pair is more similar than the negative ones. This probability is expressed as:

$$P(i|j) = \frac{\exp(s(z_i, z_j^+)/\tau)}{\exp(s(z_i, z_j^+)/\tau) + \sum_{z_j^-} \exp(s(z_i, z_j^-)/\tau)} \tag{3}$$

where τ is a temperature parameter that controls the smoothness of the distribution, and $s(z_i, z_j)$ is the Cosine similarity between anchor z_i and positive or negative samples.

The InfoNCE loss is computed as the negative log-likelihood of the positive pair:

$$L_{\text{InfoNCE}}(z_i, z_j^+, \{z_j^-\}) = -\log P(i|j) \tag{4}$$

3.2.3 Training Procedure

The training process is conducted over 100 epochs. For each epoch, mini-batches are sampled from the training set. Each batch undergoes a series of stochastic augmentations, including rotation, scaling, flipping, affine transformations, bias field distortion, gamma correction, noise addition, and blurring. These augmentations enhance the network's ability to learn invariant features and generalize better.

The model computes embeddings for both the original and augmented batches, then calculates the loss (either Triplet or InfoNCE) based on these embeddings. Network parameters are updated using the AdamW optimizer with an initial learning rate of 10^{-4}, which is decayed exponentially with a factor of 0.99 after each epoch.

We implemented the model using PyTorch and distributed the training across two NVIDIA RTX 4090 GPUs. A batch size of 128 was used for most experiments, except for CLIP and DinoV2 models, where it was reduced to 64 due to memory constraints. For the InfoNCE loss, we set the temperature $\tau = 0.5$, while for the triplet margin loss, we used a margin $m = 0.05$ with hard negative mining. The backbone was frozen during the first 5 epochs to ensure proper warm-up of the linear layer.

3.3 Embedding Similarity Analysis

To comprehensively evaluate the similarity between the learned embeddings, we employed and comapared the following distance and similarity metrics: Bray-Curtis distance, Canberra distance, Chebyshev distance, City Block (Manhattan) distance, Correlation distance, Cosine similarity, Dice similarity coefficient, Euclidean distance, Jensen-Shannon divergence, Mahalanobis distance, Matching distance, Minkowski distance, Standardized Euclidean distance (SEuclidean), and Squared Euclidean distance.

3.3.1 Similarity Distributions

For each trained model, we compute the similarity metrics between the training set and its adversarial (augmented) counterpart, the validation set and its adversarial counterpart, and for baseline similarity level assessment between the training and validation sets. These result in a distribution of the highest similarity score for each image enabling to test whether the contrastive model is capable of detecting augmented image copies and assess quantitatively the memorization degree by comparing with the train-val distribution. When aggregating over networks, losses, pretrainign and/or metrics we report the mean validation (augmented) detection with error bars representing 95 % confidence intervals, and significance test are calculated using two-tailed t-test.

Detection of Augmented Copies To evaluate the effectiveness of our similarity metrics in identifying augmented copies, we implement a threshold-based detection method. Let $\mathcal{X}_{aug} = \{x'_1, \ldots, x'_N\}$ and $\mathcal{X}_{v,aug} = \{v'_1, \ldots, v'_K\}$ represent the augmented versions of the training and validation sets, respectively. Given our similarity metric s and embedding function ϕ, we compute the similarity $S(x, x') = s(\phi(x), \phi(x'))$ between each original image $x \in \mathcal{X}$ and its augmented version $x' \in \mathcal{X}_{aug}$.

We flag x' as a potential copy if $S(x, x') > \tau$ for any $x \in \mathcal{X}$, where τ is set as the p-th percentile of the baseline similarity distribution $S_{base}(x, v) = s(\phi(x), \phi(v))$ for $x \in \mathcal{X}, v \in \mathcal{X}_v$.

Our benchmark aims to detect all images in $\mathcal{X}_{aug}$ and $\mathcal{X}_{v,aug}$ as copies of their original counterparts when using a τ equal to the 5-th percentile of $S_{base}(x, v)$. By comparing the detection rates between $\mathcal{X}_{aug}$ and $\mathcal{X}_{v,aug}$, the model's generalizability and robustness of our similarity metrics in identifying augmented copies can be assessed.

3.4 Dataset

Our study utilized an anonymized X-ray mammography dataset comprising 7,184 scans from 1,718 unique patients. The images were obtained and stored in DICOM format with a median shape of 2800 x 2082 pixels and median spacing of 0.065 x 0.065 mm.

The dataset includes two primary classes of mammography scans: normal scans and scans with calcification. To ensure the integrity of our evaluation, we performed a patient-aware train-validation split, ensuring that scans from the same patient were not distributed across different sets.

For preprocessing, all images were resized to square resolutions. During model training, images were further resized to match the natural input resolution of the backbone networks, typically 224 x 224 pixels. This dataset provides a robust foundation for training and evaluating our self-supervised contrastive network and conditional diffusion model for medical image synthesis.

3.5 Conditional Diffusion Model for Medical Image Synthesis

To enhance our dataset and evaluate the potential of generative models in medical imaging, we trained a class-conditional diffusion model using our medical imaging data. This model was designed to generate high-quality, synthetic medical images while preserving class-specific features.

Training Process We utilized a Diffusion Image Transformer (DiT) architecture [33], specifically the DiT XL/2 variant (670M), comprising 28 Transformer layers with a hidden size dimension of 1152 and 16 attention heads. The model, was initially pretrained on ImageNet and then fine-tuned on our medical imaging dataset for 100.000 steps with a learning rate of 1e-4, batch size of 2, with horizontal flip as the only augmentation.

Inference and Dataset Augmentation At inference time, we used the trained model to upsample our original dataset, effectively doubling its size. The resulting images were later processed via the best performing SSCN to showcase the usability of such privacy detector methods and their memorization characterization performance.

4 Results

In this study, we evaluated the performance of various deep learning models for a detection task, comparing different network architectures, pretraining datasets (ImageNet and RadImageNet), and loss functions (InfoNCE and Triplet). Our results reveal significant variations in performance across these factors, with some clear trends emerging.

4.1 Network, Pretraining and Loss

The performance of self-supervised networks varied significantly across different architectures, loss functions, and pretraining datasets (Figure 1). Consistently across all network architectures, the Triplet loss outperformed InfoNCE, often by a substantial margin. This superiority of Triplet loss over InfoNCE was found to be statistically significant ($p < 0.05$) for all tested network architectures and pretraining datasets, with many comparisons showing highly significant differences ($p < 0.001$).

When comparing the best configurations of different network architectures, several significant differences emerged. DinoV2 with ImageNet pretraining and Triplet loss achieved the highest overall performance (0.722), closely followed by DenseNet121 (0.710) and ResNet50 with RadImageNet pretraining (0.660). The differences between these top-performing models were not statistically significant ($p > 0.05$), suggesting that they perform comparably well.

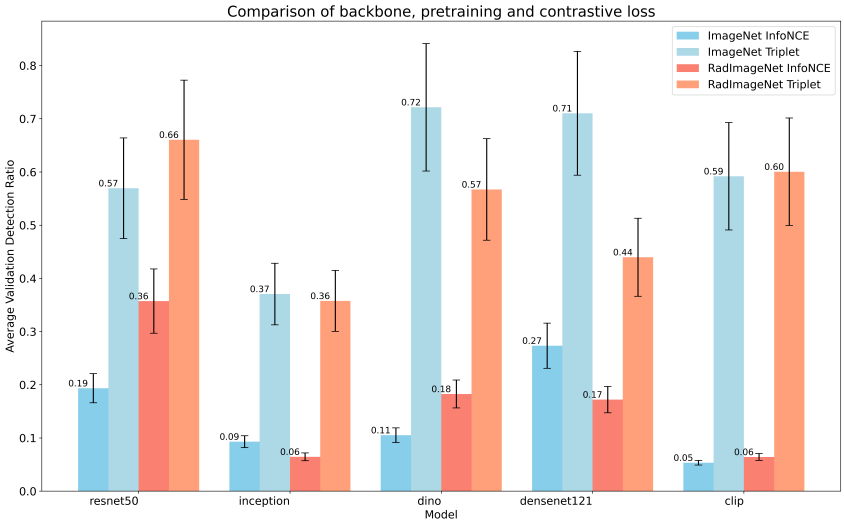

Figure 1: Comparison of network architectures performance with their best configurations.

However, significant differences were observed between the top-performing models and the Inception architecture. Inception, even in its best configuration (ImageNet, Triplet), performed significantly worse than ResNet50 ($p = 0.030$), DinoV2 ($p = 0.014$), and DenseNet121 ($p = 0.015$). The CLIP model, with its best configuration (RadImageNet, Triplet), showed intermediate performance (0.600) that was not significantly different from the top models but was marginally better than Inception ($p = 0.059$).

Interestingly, when focusing on the Triplet loss, the choice of pretraining dataset (ImageNet vs. RadImageNet) did not lead to statistically significant differences in performance for most architectures. This lack of significant difference in pretraining datasets for Triplet loss was consistent across all models, including ResNet50 ($p = 0.540$), CLIP ($p = 0.953$), Inception ($p = 0.875$), DinoV2 ($p = 0.323$), and DenseNet121 ($p = 0.060$).

These findings indicate that while the choice of network architecture and loss function (Triplet vs. InfoNCE) has a significant impact on performance, the effect of pretraining dataset is more nuanced, particularly when using Triplet loss. The top-performing models (DinoV2, DenseNet121, and ResNet50) show comparable performance, significantly outperforming Inception, with CLIP falling in between. The robustness of Triplet loss to variations in pretraining data suggests it may offer more flexibility in the choice of pretraining dataset for self-supervised learning tasks across different network architectures.

4.2 Impact of Distance Metrics on Triplet Loss Performance

In addition to comparing network architectures and pretraining datasets, we also evaluated the performance of various distance metrics when using the Triplet loss function. The results, as illustrated in Figure 2, reveal substantial variations in performance across metrics, with the mean validation detection ratios and their respective confidence intervals showing clear differences.

The Bray-Curtis distance metric demonstrated the highest mean validation detection ratio of 0.8094 (±0.1036 CI), positioning it as the best performer. It was closely followed by the Jensen-Shannon divergence (0.7882 ±0.1107 CI) and a group of Euclidean-based metrics, including Euclidean, Minkowski, and Squared Euclidean, which all achieved 0.7871 (±0.1235 CI). These metrics consistently performed well across various configurations, highlighting their robustness when applied with models trained on Triplet Margin loss.

A slightly lower performance was observed with metrics such as the City Block (Manhattan) distance (0.7813 ±0.1269 CI), the Canberra distance (0.7810 ±0.1101 CI), and the Standardized Euclidean distance (0.7708 ±0.1349 CI). Although these metrics exhibited detection ratios slightly below the top group, they still maintained strong performance, with detection ratios above 0.77. These

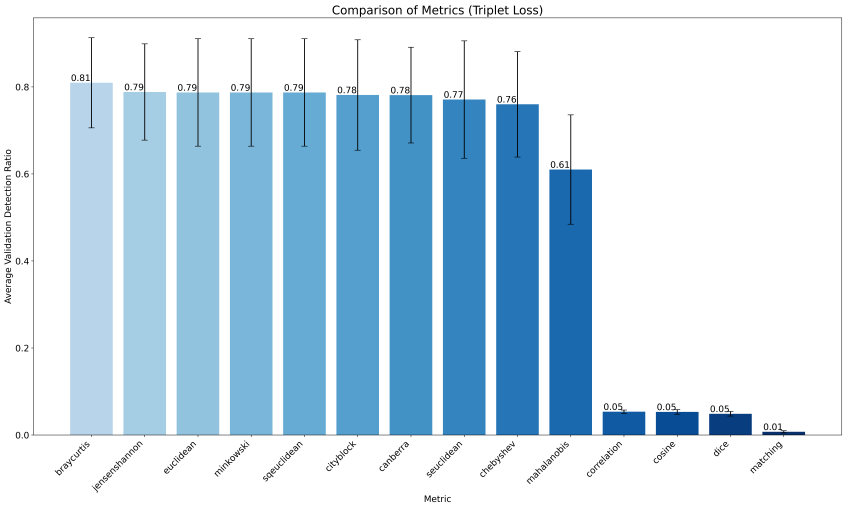

Figure 2: Comparison of distance metrics performance with Triplet loss in terms of mean validation detection ratio. Error bars represent confidence intervals.

results indicate that they are viable alternatives, particularly in situations where domain-specific considerations or computational efficiency play a role in metric selection.

On the other hand, the Chebyshev distance (0.7599 ±0.1213 CI) and the Mahalanobis distance (0.6099 ±0.1257 CI) displayed notably lower performance. The lower mean detection ratios for these metrics suggest that they may not be as effective in this task when paired with the Triplet loss function. Furthermore, the correlation-based metrics, including Correlation, Cosine, Dice, and Matching, performed significantly worse, with detection ratios falling below 0.06. Notably, the Matching distance exhibited extremely poor performance (0.0073 ±0.0025 CI), suggesting that correlation-based metrics are ill-suited for this particular detection task when using Triplet loss.

The statistical analysis of pairwise comparisons further reinforced these findings. The differences between the top-performing metrics—Bray-Curtis, Jensen-Shannon, and Euclidean-based—were not statistically significant ($p > 0.05$), indicating that their performances are comparable. However, these top-performing metrics were significantly superior to the lower-performing and poor-performing metrics, with highly significant differences observed when compared to Mahalanobis and correlation-based metrics ($p < 0.001$).

4.3 Best Combinations for Each Network Architecture

We present the best-performing combinations of network architecture, pretraining dataset, loss function and metric. Table 1 highlights the maximum validation detection achieved and the distance metric that produced this maximum value for each network configuration.

Table 1: Best combinations for each network architecture

Model	**Pretraining**	**Loss**	**Val. Detection**	**Metric**
DinoV2	ImageNet	Triplet	0.9971	Bray-Curtis
DenseNet121	ImageNet	Triplet	0.9842	SEuclidean
ResNet50	RadImageNet	Triplet	0.9568	Bray-Curtis
ResNet50	ImageNet	Triplet	0.8863	Bray-Curtis
CLIP	RadImageNet	Triplet	0.8806	City Block
CLIP	ImageNet	Triplet	0.8791	Euclidean
DinoV2	RadImageNet	Triplet	0.8604	Euclidean
DenseNet121	RadImageNet	Triplet	0.7281	Bray-Curtis
Inception	ImageNet	Triplet	0.5813	Canberra
ResNet50	RadImageNet	InfoNCE	0.5496	Euclidean

As shown in Table 1, the DinoV2 model pre-trained on ImageNet using the Triplet loss achieved the highest validation detection score (0.9971), with the Bray-Curtis distance metric. Similar trends are observed across other architectures, with DenseNet121 and ResNet50 also performing well with SEuclidean and Bray-Curtis metrics, respectively.

4.4 Memorization Characterization of Diffusion Models

Using the best-performing combinations identified for our dataset, the fine-tuned DinoV2 model was employed to analyze the memorization behavior of a DiT trained to generate synthetic mammography images (Figure 3). The augmented images are easily distinguishable from the training data, while the generated samples exhibit a slight shift towards the left of the training distribution. This shift suggests a degree of memorization, as the synthetic samples appear to be closer to the training data than the training data is to the validation images.

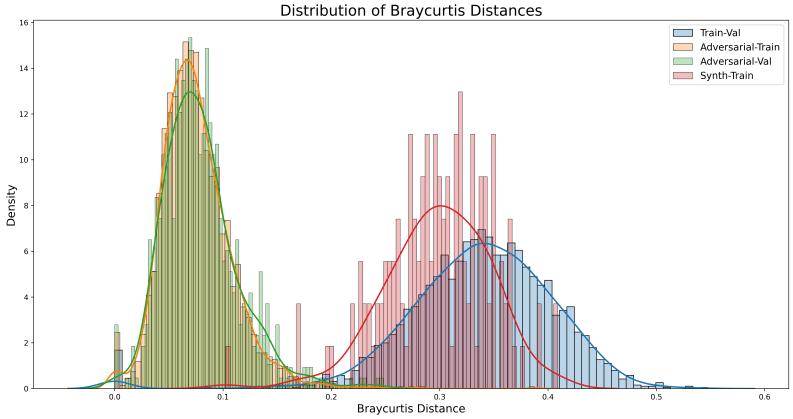

Figure 3: Memorization characterization by the two best-performing self-supervised contrastive networks, DinoV2 (left) and DenseNet121 (right), for generated samples by a DiT model.

5 Discussion

Our study presents a comprehensive, data-driven approach to evaluating and characterizing the memorization behavior of generative models in medical imaging. By systematically comparing various network architectures, loss functions, pretraining datasets, and distance metrics, we have identified optimal configurations for detecting potential privacy concerns in synthetic images. The results demonstrate that the developed method can identify all augmented images when using Triplet Margin loss with models like DinoV2, DenseNet121, and ResNet50, particularly when paired with the Bray-Curtis or Standardized Euclidean distance metrics. The ability to quantify the degree of memorization in generated images offers a method to assess the risk of patient data leakage in synthetic datasets. This approach can be integrated into the training pipeline of generative models, serving as an early warning system for memorization and potential privacy breaches.

Limitations As for limitations, our study is based on a private mammography dataset from various institutions. Although this dataset is substantial and diverse, the generalizability of our findings to other medical imaging modalities or natural image datasets remains to be validated. Future work should address these limitations by generating a foudnational model that serves for both 2D and 3D data, multi-institutional and multi-modality datasets to avoid having to fine-tune the model for each dataset. A comparative analysis of various generative model architectures and stronger conditioning forms (text or segmentation) would provide a more comprehensive understanding of memorization behavior across generative models.

References

[1] Lennart R. Koetzier, Jie Wu, Domenico Mastrodicasa, Aline Lutz, Matthew Chung, W. Adam Koszek, Jayanth Pratap, Akshay S. Chaudhari, Pranav Rajpurkar, Matthew P. Lungren, and Martin J. Willemink. Generating synthetic data for medical imaging. *Radiology*, 312(3), September 2024.

[2] Gerrit J. J. van den Burg and Christopher K. I. Williams. On memorization in probabilistic deep generative models. *CoRR*, abs/2106.03216, 2021.

[3] Ching-Yuan Bai, Hsuan-Tien Lin, Colin Raffel, and Wendy Kan. A large-scale study on training sample memorization in generative modeling, 2021.

[4] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023.

[5] Salman Ul Hassan Dar, Arman Ghanaat, Jannik Kahmann, Isabelle Ayx, Theano Papavassiliu, Stefan O. Schoenberg, and Sandy Engelhardt. Investigating data memorization in 3d latent diffusion models for medical image synthesis, 2023.

[6] Muhammad Usman Akbar, Wuhao Wang, and Anders Eklund. Beware of diffusion models for synthesizing medical images – a comparison with gans in terms of memorizing brain mri and chest x-ray images, 2023.

[7] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models, 2022.

[8] Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. An overview of diffusion models: Applications, guided generation, statistical rates and optimization, 2024.

[9] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016.

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017.

[11] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *CoRR*, abs/1802.08232, 2018.

[12] Salman Ul Hassan Dar, Marvin Seyfarth, Jannik Kahmann, Isabelle Ayx, Theano Papavassiliu, Stefan O. Schoenberg, and Sandy Engelhardt. Unconditional latent diffusion models memorize patient imaging data, 2024.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[14] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024.

[15] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016.

[16] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, volume 80 of *STOC '20*, page 954–959. ACM, June 2020.

[17] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks, 2017.

[18] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. *CoRR*, abs/2008.03703, 2020.

[19] Casey Meehan, Kamalika Chaudhuri, and Sanjoy Dasgupta. A non-parametric test to detect data-copying in generative models. *CoRR*, abs/2004.05675, 2020.

[20] Shaoyan Pan, Tonghe Wang, Richard L J Qiu, Marian Axente, Chih-Wei Chang, Junbo Peng, Ashish B Patel, Joseph Shelton, Sagar A Patel, Justin Roper, and Xiaofeng Yang. 2d medical image synthesis using transformer-based denoising diffusion probabilistic model. *Physics in Medicine amp; Biology*, 68(10):105004, May 2023.

[21] Walter H. L. Pinaya, Petru-Daniel Tudosiu, Jessica Dafflon, Pedro F da Costa, Virginia Fernandez, Parashkev Nachev, Sebastien Ourselin, and M. Jorge Cardoso. Brain imaging generation with latent diffusion models, 2022.

[22] Firas Khader, Gustav Müller-Franzes, Soroosh Tayebi Arasteh, Tianyu Han, Christoph Haarburger, Maximilian Schulze-Hagen, Philipp Schad, Sandy Engelhardt, Bettina Baeßler, Sebastian Foersch, Johannes Stegmaier, Christiane Kuhl, Sven Nebelung, Jakob Nikolas Kather, and Daniel Truhn. Denoising diffusion probabilistic models for 3d medical image generation. *Scientific Reports*, 13(1), May 2023.

[23] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS'16. ACM, October 2016.

[24] Virginia Fernandez, Pedro Sanchez, Walter Hugo Lopez Pinaya, Grzegorz Jacenków, Sotirios A. Tsaftaris, and Jorge Cardoso. Privacy distillation: Reducing re-identification risk of multimodal diffusion models, 2023.

[25] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models, 2023.

[26] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[30] Xueyan Mei, Zelong Liu, Philip M. Robson, Brett Marinelli, Mingqian Huang, Amish Doshi, Adam Jacobi, Chendi Cao, Katherine E. Link, Thomas Yang, Ying Wang, Hayit Greenspan, Timothy Deyer, Zahi A. Fayad, and Yang Yang. Radimagenet: An open radiologic deep learning research dataset for effective transfer learning. *Radiology: Artificial Intelligence*, 4(5), September 2022.

[31] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Procedings of the British Machine Vision Conference 2016*, BMVC 2016, pages 119.1–119.11. British Machine Vision Association, 2016.

[32] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[33] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

A Appendix

A.1 Backbone Comparison p-values

Table 2 shows the p-values for the comparison between ImageNet and RadImageNet pretraining across different model backbones and loss functions.

Table 2: P-values for Backbone Comparison (ImageNet vs RadImageNet)

Model	Loss	p-value
ResNet50	InfoNCE	0.0205
ResNet50	Triplet	0.5401
Inception	InfoNCE	0.0428
Inception	Triplet	0.8749
DINO	InfoNCE	0.0147
DINO	Triplet	0.3230
DenseNet121	InfoNCE	0.0498
DenseNet121	Triplet	0.0599
CLIP	InfoNCE	0.1731
CLIP	Triplet	0.9528

The results indicate varying levels of statistical significance in the performance difference between ImageNet and RadImageNet pretraining across different model architectures and loss functions. P-values below 0.05 suggest a statistically significant difference:

- ResNet50, Inception, DINO, and DenseNet121 show statistically significant differences ($p < 0.05$) when using InfoNCE loss.
- The Triplet loss generally shows no significant difference between ImageNet and RadImageNet pretraining across all models.
- CLIP shows no significant difference for either loss function.

These results suggest that the choice of pretraining dataset (ImageNet vs RadImageNet) may have a more pronounced effect when using InfoNCE loss, particularly for certain model architectures.

A.2 Loss Function Comparison Results

Table 3 presents the comparison between Triplet and InfoNCE loss functions across different model backbones and pretraining datasets.

Table 3: Comparison of Triplet and InfoNCE Loss Functions

Model	Pretrain	p-value	Triplet Mean	InfoNCE Mean
ResNet50	ImageNet	0.0008	0.5694	0.1934
ResNet50	RadImageNet	0.0249	0.6604	0.3572
Inception	ImageNet	<0.0001	0.3705	0.0929
Inception	RadImageNet	<0.0001	0.3576	0.0645
DINO	ImageNet	<0.0001	0.7216	0.1051
DINO	RadImageNet	0.0006	0.5671	0.1826
DenseNet121	ImageNet	0.0016	0.7102	0.2733
DenseNet121	RadImageNet	0.0019	0.4396	0.1720
CLIP	ImageNet	<0.0001	0.5919	0.0533
CLIP	RadImageNet	<0.0001	0.6004	0.0642

The results show a consistent and statistically significant difference between the performance of Triplet and InfoNCE loss functions across all model architectures and pretraining datasets. Key observations include:

- All comparisons show p-values well below 0.05, indicating strong statistical significance in the difference between Triplet and InfoNCE loss performance.

- Triplet loss consistently outperforms InfoNCE loss across all models and pretraining datasets, as evidenced by the higher mean values.
- The performance gap between Triplet and InfoNCE loss appears to be more pronounced for some models (e.g., DINO, CLIP) compared to others.
- The choice of pretraining dataset (ImageNet vs RadImageNet) seems to influence the magnitude of the difference between the two loss functions, though the trend of Triplet loss outperforming InfoNCE remains consistent.

These findings suggest that the choice of loss function has a substantial impact on model performance, with Triplet loss demonstrating superior results across various model architectures and pretraining scenarios. This consistent pattern underscores the importance of loss function selection in the design of contrastive learning frameworks for image analysis tasks.

A.3 Network Architecture Comparison Results

Table 4 presents the pairwise comparisons between different network architectures, considering their performance with specific pretraining datasets and loss functions.

Table 4: Pairwise Comparison of Network Architectures

Model 1	Model 2	p-value	Model 1 Mean	Model 2 Mean
ResNet50 (RadImageNet, Triplet)	Inception (ImageNet, Triplet)	0.0298	0.6604	0.3705
ResNet50 (RadImageNet, Triplet)	DINO (ImageNet, Triplet)	0.7125	0.6604	0.7216
ResNet50 (RadImageNet, Triplet)	DenseNet121 (ImageNet, Triplet)	0.7606	0.6604	0.7102
ResNet50 (RadImageNet, Triplet)	CLIP (RadImageNet, Triplet)	0.6940	0.6604	0.6004
Inception (ImageNet, Triplet)	DINO (ImageNet, Triplet)	0.0139	0.3705	0.7216
Inception (ImageNet, Triplet)	DenseNet121 (ImageNet, Triplet)	0.0147	0.3705	0.7102
Inception (ImageNet, Triplet)	CLIP (RadImageNet, Triplet)	0.0588	0.3705	0.6004
DINO (ImageNet, Triplet)	DenseNet121 (ImageNet, Triplet)	0.9461	0.7216	0.7102
DINO (ImageNet, Triplet)	CLIP (RadImageNet, Triplet)	0.4465	0.7216	0.6004
DenseNet121 (ImageNet, Triplet)	CLIP (RadImageNet, Triplet)	0.4825	0.7102	0.6004

The results reveal interesting patterns in the performance of different network architectures:

- ResNet50 (RadImageNet, Triplet) shows significantly better performance than Inception (ImageNet, Triplet) with a p-value of 0.0298.
- There is no statistically significant difference between ResNet50 (RadImageNet, Triplet) and DINO, DenseNet121, or CLIP, as evidenced by high p-values (>0.05).
- Inception (ImageNet, Triplet) consistently underperforms compared to other architectures, with statistically significant differences against DINO and DenseNet121 (p-values < 0.05).
- DinoV2, DenseNet121, and CLIP show comparable performance, with no statistically significant differences among them (p-values > 0.05).
- The choice of pretraining dataset (ImageNet vs RadImageNet) appears to influence performance, but the effect varies across architectures.

These findings suggest that:

1. ResNet50, DinoV2, DenseNet121, and CLIP demonstrate robust performance across different pretraining scenarios when using Triplet loss.
2. Inception architecture may not be optimal for this particular task, consistently showing lower performance.
3. The impact of pretraining dataset choice (ImageNet vs RadImageNet) may be architecture-dependent and warrants further investigation.

Overall, these results underscore the importance of carefully selecting network architectures and pretraining strategies in contrastive learning frameworks for image analysis tasks. The comparable performance of several architectures (ResNet50, DINO, DenseNet121, CLIP) suggests that factors beyond architecture, such as loss function and pretraining data, play crucial roles in determining overall system performance.

A.4 InfoNCE Loss Metric Results

In this section we show the results on metric comparison for the models trained with InfoNCE loss (Table 4). We observe that on average the reults are much lower than when using Triplet Margin loss, and distance metrics like Euclidean are preferred over correlation, Mahalanobis or Cosine similarity.

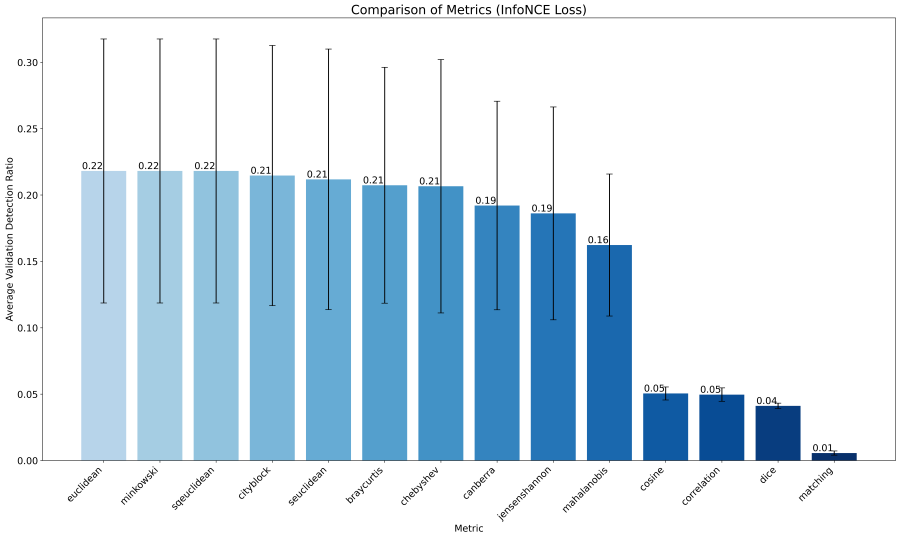

Figure 4: Comparison of distance metrics performance with InfoNCE loss in terms of mean validation detection ratio. Error bars represent confidence intervals.

A.5 Class conditional Performance Metrics DiT vs StyleGAN2

The generative imaging results, shown in Figure **??**, indicate that the class-conditional DiT model performs better or at least comparably across all relevant metrics to the unconditional StyleGAN2s. DiT models learns more comprehensively the real image distribution and is less affected by mode-collapse. Both models exhibit a tendency for memorization, as the generated data closely resembles the training data more than the training data resembles the validation data. However, the degree of memorization observed is not excessive after manually inspection.

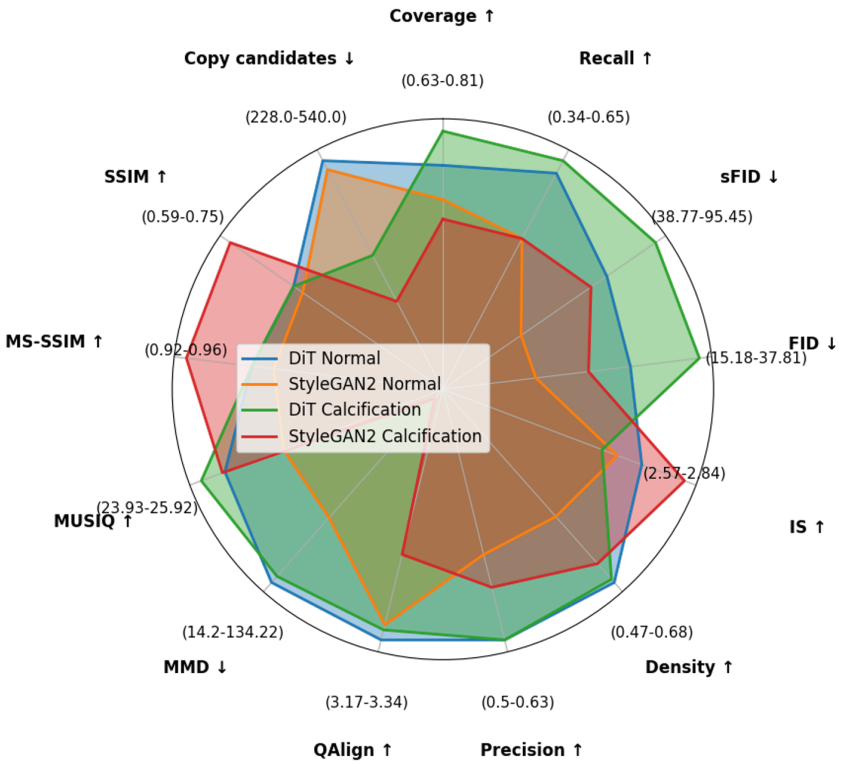

Figure 5: High-Resolution Image Synthesis Comparison Between StyleGAN and Diffusion Models (2048 x 2048 pixels).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main goals and scope are specified at the abstract and introduction by acknowledging we build upon a previous work on the memorization topic and we focus on the optimization of it via a thorough benchmark to ensure that non-trivial decisions about the method configuration (Backbone, Loss, Pretraining, Similarity metric) are backed with experiments.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: The paper comment on the limitations of our work in the last section of the paper where we acknowledge the generalization aspect of our method remains unclear and fine-tuning to each individual dataset might be required. And we propose as future work to create a multi-modality medical imaging dataset to develop a contrastive foundational model capable of generalize to unseen data and avoid the fine-tuning step.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We do define the problem in a closed format so that all assumptions and criteria are clear, specifically for our detection benchmark via adversarial augmentations.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides a complete definition of the attained problem, a definition of the networks, losses, pretraining, hyperparameters and metrics employed as well as the resources needed to run all the experiments (Methods section).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification We do provide access to the code that enables to reproduce the results with any dataset in a flexible manner. However, we do not disclose the dataset employed due to the sensitivity of the medical images which although anonymized we are not allowed to share openly.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not

including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We do describe all the hyper-parameters employed and how they were selected as well as the optimizer and other training details selected via manual tunning in the methods section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We describe the statistical tests employed the meaning of the error bars which support the main claims of our paper (Methods and results section, figure 1 2 and Tables in supplementary)

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We specify the hardware used to run all the experiments in the implementation details of the methods section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read and checked that we conform with the code of ethics of NeurIPS and we preserve anonymity in the current submission for a fair and unbiased review process.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the different aspects to consider about generative models memorization in medical imaging and how by not implementing aafety measure patient data could be leaked into the generated datasets which would violate privacy regulations (Introduction, Related work and Discussion)

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We are the creators of all the models, code and data employed with the usage of open source fully complying with the licensing of each pretrained backbone.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets, the code, are properly documented in the GitHub repository specified on the footnote of the first page of the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Human subjects were not use in an instructive manner, no instructions to specify.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: The study contains retropecitve fully anonymized data based and all subjects gave their consent to use their health data (images) for further research and was conducted based on a IRB approval

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.