UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA DELL'INFORMAZIONE

Dataset inference per reti neurali generative

Relatore:
Prof. Simone Milani

Laureando:
Matteo De Gobbi

ANNO ACCADEMICO 2022 / 2023.

Data di laurea 16/11/2023

Indice:

1 Introduzione **3**

2 Membership Inference Attacks **4**
2.1 M.I.A e I.I.A 4
2.2 Differential privacy nei modelli di machine learning 4
2.3 Black Box e White Box 5
2.4 M.I.A. su classificatori 5
2.5 GAN 7
2.5.1 Value function della GAN 8
2.6 M.I.A su GAN 8
2.6.1 Attacco white box 9
2.6.2 Attacco black box 9
2.6.3 Attacco black box con informazioni addizionali 9
2.7 Precisione e Recall nei Membership inference attacks 10

3 Setting sperimentale **12**
3.1 Struttura dell'attacco white box 13
3.2 Struttura dell'attacco black box 16

4 Analisi dei risultati **18**
4.1 Qualità delle immagini generate 18
4.2 Performance dell'attacco white box 21
4.2.1 Dataset con 128 immagini 21
4.2.1.1 Separazione delle likelihood tra immagini di train e test 21
4.2.1.2 Precisione nelle top k immagini per likelihood 23
4.2.2 Dataset con un maggior numero di immagini 25
4.2.3 Precisione in funzione della FID 27
4.3 Performance attacco black box 29
4.3.1 Shadow training set con la disponibilità di 6000 generate e 1000 epoche di training della shadow GAN 29
4.3.1.1 Separazione delle likelihood tra training e test set e precisione nelle top k 29
4.3.2 Effetto del numero di epoche di training della shadow GAN e numero di immagini generate disponibili 36
4.3.2.1 Numero di epoche shadow GAN 37

4.3.2.2 Shadow training set con diversi numeri di immagini generate disponibili 38
4.3.3 Precisione in funzione della FID degli attacchi black box 42
4.4 Attacchi sul dataset Anime Faces 43
4.4.1 Qualità delle facce generate 45
4.4.1.1 FID in funzione del numero di immagini nel dataset 47
4.4.2 Attacco whitebox 48
4.4.2.1 GAN 128, 1200 48
4.4.2.2 GAN 4096, 16000 e 45789 49

5 Conclusioni e sviluppi futuri **53**

Bibliografia **55**

1 Introduzione

La gestione della privacy [1] nell'intelligenza artificiale è una delle tematiche più calde degli ultimi anni [2]. Infatti l'allenamento di un modello di machine learning che utilizza dei dati sensibili presenta alcuni rischi qualora sia possibile recuperare o stimare i dati di training partendo dal modello stesso.

Le due principali tipologie [2] di attacchi su modelli di machine learning sono:

- Model Inversion, cioè un tipo di attacco che a partire da l'output di un modello di machine learning cerca di costruire un dato di input artificiale che possa corrispondere a quel determinato output. Questo tipo di attacco permette di ricostruire la distribuzione dei dati di training del modello di machine learning attaccato.
- Membership Inference, il tipo di attacco che analizzeremo, consiste nel determinare se un certo dato fosse presente nel dataset del modello di machine learning attaccato.

L'importanza di studiare questo tipo di attacchi è chiara nel campo medico: certi dataset non possono essere divulgati pubblicamente perché contengono informazioni sensibili sui pazienti. Normalmente questi dati non sarebbero disponibili al pubblico, ma nel caso vengano addestrati dei modelli di machine learning usando questi dati potrebbe risultare che si possano ricavare informazioni sui dati di training usando il modello di machine learning come metodo di attacco.

Ad esempio un modello di machine learning che genera immagini di impronte digitali artificiali potrebbe essere sfruttato per inferire se una persona fosse presente nel dataset di training [3].

L'appartenenza al dataset può essere una informazione sensibile in sè in alcuni casi. Ad esempio l'appartenenza al training set di una GAN addestrata su impronte digitali di persone arrestate implica che la persona di cui si è determinata la membership sia stata arrestata o schedata in passato.

2 Membership Inference Attacks

2.1 M.I.A e I.I.A

Si parla di Membership Inference Attack (o M.I.A.) se è possibile con una certa accuratezza determinare se un certo dato fosse presente nel training set di un modello. Si parla invece di Identity Inference Attack (o I.I.A.) se è possibile determinare se nel training dataset fosse presente qualche dato personale corrispondente ad una certa persona avendo a disposizione un altro dato della stessa (ad esempio a partire da una foto di un individuo determinare se nel training dataset fosse presente un'altra foto dello stesso individuo). A tal proposito, il capitolo corrente presenterà le tecniche di inferenza sui modelli classificazione e sulle Generative Adversarial Networks e le loro implicazioni sulla privacy.

2.2 Differential privacy nei modelli di machine learning

Nel machine learning viene usata una definizione di privacy chiamata "differential privacy". Un modello si definisce differentially private se un attaccante non può utilizzarlo per ottenere informazioni sul training dataset che non siano ricavabili da altri dataset provenienti dalla stessa distribuzione.

Questo quindi non significa che non si possa dedurre alcuna informazione sulla distribuzione dei dati di training. Infatti sarebbe una richiesta troppo forte in quanto un modello per essere utile deve stimare in maniera più o meno accurata la distribuzione che ha generato il dataset. Richiedere che dal modello non si possa dedurre alcuna informazione sulla distribuzione del dataset equivale a chiedere che il modello non funzioni. Differential privacy quindi è la richiesta che dal modello non si possa risalire a informazioni specifiche dei dati della distribuzione che erano presenti nel training dataset.

Ad esempio un classificatore che, ricevendo in input una serie di dati fisiologici misurati su una popolazione di individui è in grado di stimare se l'individuo in esame è predisposto ad una certa malattia, non viola la differential privacy se:

1. È possibile costruire un altro modello tale che addestrandolo sulla stessa distribuzione troverà la stessa relazione tra categoria di persone e malattia.

2. Non è possibile ricavare informazioni sulle persone presenti nel training dataset, quindi il modello deve essere resistente a M.I.A. e I.I.A.

In questo esempio si nota che violare la differential privacy è particolarmente grave in quanto permette di risalire allo stato di salute di una specifica persona se presente nel dataset di addestramento.

2.3 Black Box e White Box

Per gli inference attacks possiamo definire due scenari o condizioni sperimentali diverse:

- Attacco white box (scatola bianca) ovvero l'attaccante conosce informazioni sul modello da attaccare come: tipo di modello utilizzato, numero di layer, parametri.
- Attacco black box (scatola nera) ovvero l'attaccante non ha alcuna informazione sul funzionamento interno del modello e può solo utilizzarlo osservando gli output forniti.

Nell'attacco black box, il modello di machine learning da attaccare può essere visto come una API a cui è possibile fare delle richieste con degli input scelti dall'attaccante e ottenere le risposte da analizzare per l'attacco. Ci concentreremo principalmente su questo secondo modello in quanto è più simile ad un caso reale in cui un attaccante effettua un M.I.A. su un modello target a cui non ha accesso direttamente.

2.4 M.I.A. su classificatori

Chiamato $\mathcal{T}$ il modello target su cui vogliamo svolgere l'attacco $\mathcal{D}_{train,\mathcal{T}}$ è il dataset di training di $\mathcal{T}$.

Il modello target determina a quale tra k classi è più probabile appartenga l'input.

Il classificatore dà in output un vettore lungo k dove ogni componente rappresenta la probabilità che l'input appartenga alla corrispondente classe.

Ad esempio:

$$\begin{bmatrix} cane \\ gatto \\ orso \\ volpe \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.1 \\ 0.1 \\ 0.2 \end{bmatrix} \tag{1}$$

L'intuizione su cui ci basiamo è che il modello classificherà gli input che erano già presenti nel training set ($\mathcal{D}_{train,\mathcal{T}}$) con una confidenza maggiore nel vettore di predizione. Ad esempio un cane presente nel training set viene classificato con alta confidenza:

$$\begin{bmatrix} cane \\ gatto \\ orso \\ volpe \end{bmatrix} = \begin{bmatrix} 0.9 \\ 0.025 \\ 0.025 \\ 0.05 \end{bmatrix} \tag{2}$$

Come analizzato in [4] l'overfitting del modello target $\mathcal{T}$ rende maggiori le differenze nella confidenza della classificazione tra dati nuovi e dati già visti dal modello durante il training. Quindi l'overfitting di $\mathcal{T}$ facilita attacchi di inferenza.

Possiamo quindi addestrare un nuovo modello di machine learning $M_{inference}$, un classificatore binario che a partire da queste differenze nell'output tra i dati in $\mathcal{D}_{train,\mathcal{T}}$ e quelli in $\overline{\mathcal{D}_{train,\mathcal{T}}}$ (il complemento) determini se l'input $\in \mathcal{D}_{train,\mathcal{T}}$ o no.

Per poter addestrare $\mathcal{M}_{inference}$ avremmo bisogno dei vettori di predizione con la corrispondente label `in` o `out` in base all'appartenenza a $\mathcal{D}_{train,\mathcal{T}}$.

Non abbiamo a disposizione questi dati per il modello $\mathcal{T}$, quindi creiamo una serie di "shadow models" $\mathcal{S}_i$ il cui scopo è imitare $\mathcal{T}$. Gli shadow models $\mathcal{S}_i$ sono creati dall'attaccante e quindi è possibile controllarne il training set $\mathcal{D}_{train,\mathcal{S}_i}$ e quindi l'attaccante ha a disposizione dei vettori di predizione con la corrispondente label `in` e `out`.

Possiamo quindi addestrare il classificatore binario $\mathcal{M}_{inference}$ in modo che determini se un certo input appartenga a $\mathcal{D}_{train,\mathcal{S}_i}$ oppure no in base al vettore di predizione corrispondente. L'idea è che se gli shadow models $\mathcal{S}_i$ si comportano in maniera abbastanza simile a $\mathcal{T}$ la capacità di $\mathcal{M}_{inference}$ di discriminare `in` e `out` su dati in $\mathcal{D}_{train,\mathcal{S}_i}$ si tradurrà nella capacità di discriminare `in` e `out` su dati in $\mathcal{D}_{train,\mathcal{T}}$.

In questo caso avere informazioni aggiuntive su $\mathcal{T}$ (ad esempio il tipo del modello) permetterebbe di creare dei $\mathcal{S}_i$ più simili aumentando l'accuratezza del nostro attacco.

Per l'addestramento dei $\mathcal{S}_i$ avremo che $\mathcal{D}_{train,\mathcal{S}_i}$ contiene tutti dati con una determinata classe i (ad esempio $\mathcal{D}_{train,\mathcal{S}_i}$ ha tutti cani,$\mathcal{D}_{train,\mathcal{S}_2}$ ha tutti gatti ecc...), questo perché la distribuzione del vettore di predizione può dipendere dalla classe dell'input (ad esempio se è più facile per un modello classificare certi animali rispetto ad altri). Quindi si addestrano k shadow models uno per ogni classe in modo da catturare la distribuzione dei vettori di predizione condizionata dalla appartenenza ad una classe.

Per il modello $\mathcal{M}_{inference}$ si può utilizzare qualsiasi modello che permetta la classificazione binaria.

2.5 GAN

Le generative neural network [5] sono una classe di modelli di deep learning utilizzati per generare dei dati dalla stessa distribuzione dei dati di training. A differenza di altri modelli generativi le GAN non approssimano esplicitamente la densità di distribuzione dei dati di training ma trasformano del rumore in input in dei dati con una alta likelihood di appartenere alla distribuzione dei dati di training.

Una GAN è composta da due reti neurali: una rete generatrice che prende in input rumore e genera dei dati e una rete discriminatrice che prende in input i dati provenienti dalla distribuzione originale e quelli generati dalla rete generatrice e cerca di discriminare quali i dati originali da quelli generati.

La rete discriminatrice viene addestrata in modo da raggiungere la massima probabilità che riesca a distinguere dati originali da quelli generati. La rete generatrice viene addestrata in modo da massimizzare la probabilità che la rete discriminatrice classifichi i dati generati come dati originali.

Le due reti vengono addestrate alternandosi cercando di tenere la rete discriminatrice vicino all'ottimalità in modo da forzare la rete generatrice a generare dati più simili a quelli originali.

2.5.1 Value function della GAN

Nella progettazione di una GAN, i valori numerici generati dalla rete e fondamentali nel processo di training sono:

- $D(x)$, cioè la probabilità che il discriminatore classifichi correttamente un dato proveniente dalla distribuzione originale;
- $D(G(z))$, cioè la probabilità che il discriminatore classifichi erroneamente un dato generato come provieniente dalla distribuzione originale.

Una GAN può essere modellata come un gioco minimax dove il generatore G e il discriminatore D competono con la seguente value function (3):

$$\min_G \max_D \{\underbrace{\mathbb{E}_x[\log D(x)]}_{(A)} + \underbrace{\mathbb{E}_z[1 - \log D(G(z))]}_{(B)}\} \tag{3}$$

Il generatore può intervenire solo su (B). Minimizzando (B) il generatore sta diminuendo la likelihood che D riesca a distinguere gli output di G da dati reali. Quindi G cerca di aumentare i falsi positivi di D.

Il discriminatore cerca di massimizzare (B) ovvero la likelihood di identificare come falsi i dati generati da G. Quindi D cerca di diminuire i falsi positivi. Inoltre D massimizza (A) ovvero la likelihood che classifichi come veri i dati provenienti dalla distribuzione originale. Quindi D cerca di diminuire i falsi negativi.

Il termine (A) nella value function è necessario per evitare che D classifichi come falso ogni input, ottenendo precisione del 100% ma recall del 0%.

È stato dimostrato in [5] che alternando il gradient descent di D e G la GAN converge a $D(x) = \frac{1}{2}$ ovvero il generatore crea dei dati che D non riesce a distinguere da quelli originali.

2.6 M.I.A su GAN

Un M.I.A. su una GAN [2] presenta delle difficoltà ulteriori rispetto a quello su un modello classificatore. Infatti per l'attacco su un classificatore si hanno a disposizione i vettori di predizione. L'attaccante può sfruttare i diversi

livelli di confidenza nei vettori di predizione su input appartenenti ai dati di training rispetto a dati mai visti dal classificatore per determinare se un certo dato appartenente al training set. Questo non è possibile nelle GAN quindi sono stati trovati paradigmi alternativi[6]:

2.6.1 Attacco white box

Nel paradigma white box contro una GAN si ha disposizione la rete neurale discriminatrice D utilizzata durante il training. Per l'attacco è sufficiente utilizzare la rete discriminatrice e, nel caso di overfitting, gli input appartenenti al training dataset avranno una confidenza più alta quando vengono classificati. Questo procedimento è simile al M.I.A. su classificatori che utilizza la confidenza del vettore di predizione.

2.6.2 Attacco black box

Nell' attacco black box senza informazioni addizionali su una GAN non abbiamo a disposizione gli output delle rete neurale discriminatrice, abbiamo solo a disposizione i dati generati dalla GAN target.

L'idea per poter effettuare un attacco è di addestrare localmente una GAN detta shadow GAN o GAN ombra, che riproduca in parte le operazioni sviluppate dalla GAN originale. Si può sfruttare l'attacco white box sulla shadow GAN locale di cui abbiamo a disposizione la rete discriminatrice.

Se la GAN locale è abbastanza simile a quella da attaccare e quest'ultima presenta overfitting possiamo con successo determinare se un dato era presente nel training set. Questa idea è simile a quella degli shadow models utilizzati per il M.I.A. nei classificatori: creare un modello locale di cui l'attaccante ha il controllo per simulare il modello target che vuole attaccare. Per addestrare la GAN locale non si hanno a disposizione membri del traning dataset della GAN attaccata quindi si usano i dati generati da quest'ultima.

2.6.3 Attacco black box con informazioni addizionali

Il tipo di attacco precedente dove l'attaccante non ha nessuna informazione sul modello da attaccare è molto restrittivo, a volte l'attaccante ha a disposizione una parte del dataset originale (una parte del dataset è pubblica, data breach, foto o testi presi da internet ecc.). Ad esempio per una GAN che genera testi l'attaccante può sapere che nel training dataset fosse presente un certo

romanzo e voler inferire altri testi usati nell'addestramento. L'attaccante può sfruttare queste informazioni aggiuntive sul training dataset per migliorare le prestazioni dell'attacco in due modi: attacco discriminativo e attacco generativo.

Nell'attacco discriminativo è richiesto che l'attaccante abbia a disposizione alcuni dati che non siano stati utilizzati nel training della GAN, che chiamiamo $\mathcal{A}_{\text{not train}}$, ed eventualmente alcuni dati appartenenti al training set, chiamati $\mathcal{A}_{\text{train}}$.

Si addestra una rete discriminatrice locale dove come input falsi vengono dati $\mathcal{A}_{\text{not train}}$ e come input veri dei dati generati dalla GAN target più, se sono a disposizione dell'attaccante, anche i dati di $\mathcal{A}_{\text{train}}$. In questo modo il discriminatore locale impara a differenziare dati non presenti nel training set da quelli presenti nel training set/generati dalla GAN target. Una volta addestrato il discriminatore locale si può procedere con un attacco uguale a quello white box.

Nell'attacco generativo è richiesto che l'attaccante abbia a disposizione alcuni dati che siano stati utilizzati nel training della GAN, che chiamiamo $\mathcal{A}_{\text{train}}$, ed eventualmente alcuni dati appartenenti al test set, chiamati $\mathcal{A}_{\text{test}}$.

Si addestra una shadow GAN locale dove come input veri vengono usati $\mathcal{A}_{\text{train}}$ e dati generati dalla GAN target mentre come input falsi vengono usati dati generati dalla GAN shadow locale e, se sono a disposizione, anche i dati di $\mathcal{A}_{\text{test}}$. Di nuovo si può usare il discriminatore della GAN locale per procedere con un attacco white box.

È da notare come nell'approccio discriminativo sia necessario avere $\mathcal{A}_{\text{not train}}$ ma $\mathcal{A}_{\text{train}}$ potrebbe essere vuoto, in questo caso si possono anche usare solo i dati generati dalla GAN target. Nell'approccio generativo invece è $\mathcal{A}_{\text{test}}$ a non essere necessario. In ogni caso avere più informazioni possibili sia sul training set che sul test set originali migliorerà le prestazioni dell'attacco.

2.7 Precisione e Recall nei Membership inference attacks

Nei modelli di machine learning utilizziamo due metriche per valutare il successo di un discriminatore:

$$\text{Precision} = \frac{\text{truePositives}}{\text{truePositives} + \text{falsePositives}} \tag{4}$$

$$\text{Recall} = \frac{\text{truePositives}}{\text{truePositives} + \text{falseNegatives}} \tag{5}$$

La precision (4) è la frazione degli input considerati reali dal discriminatore che sia veramente un input reale.

Il recall (5) la frazione degli input reali che il discriminatore identifica correttamente come reali.

In pratica la precision rappresenta la probabilità che un input classificato come reale sia veramente reale e il recall rappresenta la probabilità che un input reale venga riconosciuto come reale. (Queste metriche non sono definite solo per i discriminatori delle GAN ma in generale per qualsiasi classificatore binario)

Negli attacchi di membership inference in letteratura viene utilizzata la precision come metrica per valutare il successo dell'attacco. Anche negli attacchi presentati in seguito viene usata la precision come metrica di performance.

---

3 Setting sperimentale

Per gli esperimenti è stato utilizzato Python 3.9.18 con la libreria Tensorflow 2.10.0 usando una NVidia RTX 4070 (Inoltre sono state utilizzate le librerie numpy per matrici e array, matplotlib per i grafici e Pillow per elaborare e salvare velocemente le immagini generate su disco). I dataset utilizzati sono stati:

- MNIST, 70000 immagini 28x28 di cifre da 0 a 9 scritte a mano, è un dataset molto comune nel machine learning in quanto la dimensione ridotta delle immagini permette di addestrare velocemente i modelli
- Dataset Anime Faces da Kaggle, dal dataset originale sono state rimosse tutte le immagini di dimensione inferiore a 64x64 perché sono state giudicate di qualità troppo bassa il dataset ridotto è composto da 57237 immagini. Il dataset è composto da facce di personaggi di cartoni animati giapponesi.

Le GAN target e shadow hanno la stessa architettura e non sono state utilizzate informazioni sulla architettura nell'attacco, riprendendo il modello in [4] in cui l'attaccante usa una API per il machine learning come quella di Google o di Amazon. Il generatore e discriminatore delle GAN sono Convolutional Neural Networks.

Le grandezze dei layer delle GAN sono state adattate in base alle differenti dimensioni delle immagini dei diversi dataset ma per il resto l'architettura usata è stata la stessa per tutti i dataset.

Negli attacchi presentati in seguito gli obiettivi sono:

- Mostrare le differenze nelle performance di attacchi white box e black box
- Studiare l'effetto della grandezza del dataset della GAN target sugli attacchi (sia white che black box)
- Studiare l'effetto del numero di epochs del training della GAN target sugli attacchi (sia white che black box)
- Studiare l'effetto del numero di immagini generate dalla GAN target a disposizione della GAN shadow sugli attacchi black box

- Studiare l'effetto del numero di epochs del training della GAN shadow sugli attacchi black box
- Capire se l'attacco black box può avere successo avendo a disposizione meno immagini ma compensando con più epoch di training della shadow
- Studiare il collegamento tra la qualità delle immagini generate dalla GAN target (misurata con la Frechet Inception Distance dal dataset originale) e il successo degli attacchi (sia white che black box)
- Studiare se è possibile addestrare una GAN che sia resistente agli attacchi black box e white box che produca immagini di qualità sufficiente

(In questa sezione per attacco black box si intende quello senza informazioni addizionali sul training o test dataset presentato nella sezione precedente).

3.1 Struttura dell'attacco white box

Possiamo descrivere le varie fasi dell'attacco white box come segue:

- Definizione dell'architettura di generatore e discriminatore della GAN target tramite keras (API ad alto livello per tensorflow) e delle loro training loss e training step.

- Importazione del dataset (da disco o download tramite `tensorflow.keras.datasets` nel caso di MNIST) e divisione in train e test set. La grandezza del training set è definita tramite il parametro `DATASET_SIZE` in modo da poter studiare l'effetto di diverse grandezze del dataset.

- Vengono definiti i parametri `MAX_EPOCHS`, che specifica il numero totale di epoche di training della GAN, e `epoch_per_iter` che determina ogni quante epoche verrà eseguito un tentativo di attacco white box, questo serve a determinare l'impatto del numero di epoche sul successo dell'attacco.

- Poi procede con un ciclo nel quale la GAN viene addestrata per `epochs_per_iter` epoche e poi viene attaccata, si utilizza il discriminatore appena addestrato per determinare le likelihood delle immagini nel training set e di quelle nel test set. Inoltre alla fine di ogni iterazione viene salvato un esempio di 16 immagini generate in modo da vedere dopo quante epoche la qualità sia sufficiente.

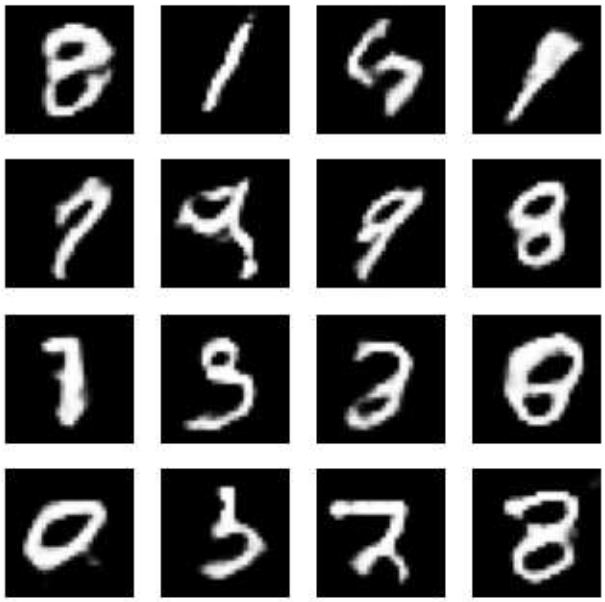


Fig 1: esempio di 16 immagini generate dopo 5000 epochs con 1024 sample nel training set

- Vengono poi fatti gli istogrammi delle likelihood ottenute: se le due distribuzioni di train e test set sono abbastanza separate è possibile distinguere se un dato è presente nel dataset di training con un'accuratezza elevata.

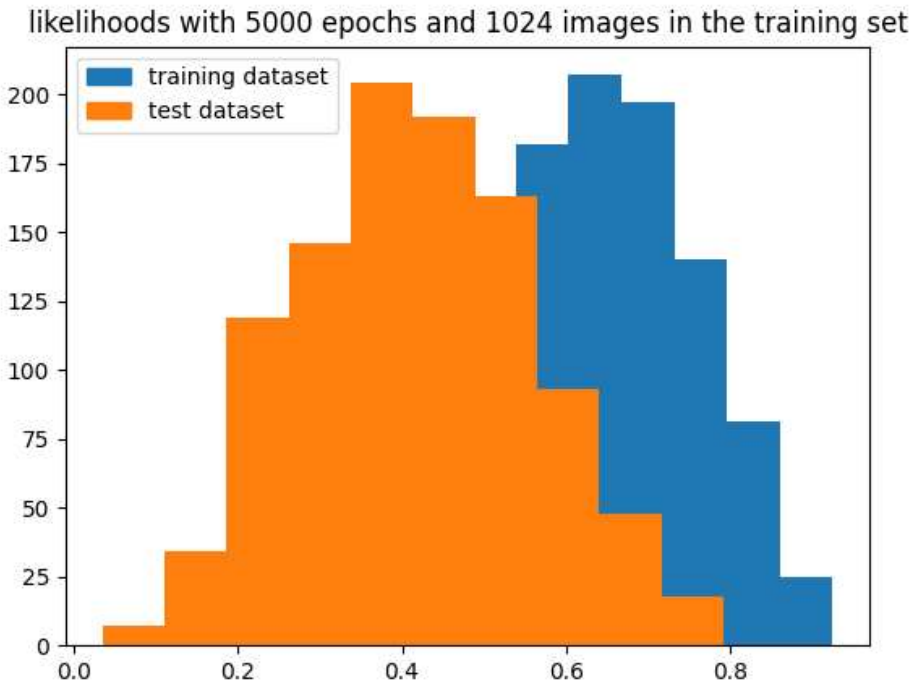


Fig 2: istogrammi delle likelihood del train e test set

- I valori delle likelihood vengono poi ordinati in modo decrescente e per vari valori di k si conta quante immagini erano nel training set nelle prime top k. Calcolando il rapporto

$$\frac{\#\{\text{immagine } \in \text{training set} \mid \text{immagine nelle prime } k \text{ likelihood}\}}{k} \tag{6}$$

  otteniamo la precision dell'attacco per un determinato k.

  Avendo usato un ugual numero di immagini di test e train nel calcolo delle likelihood per avere successo deve essere che la precision > 0.50.

- Poi si procede con la prossima iterazione dove si ripeterà questo attacco dopo aver addestrato la GAN per altre `epochs_per_iter` epoche, il ciclo si fermerà dopo aver addestrato per `MAX_EPOCHS` la GAN.

- Finito il ciclo di training e attacco si crea e salva un grafico della precision per i diversi k e per il diverso numero di epoche in modo da poterne studiare l'effetto. Poi confronteremo questi grafici per valori diversi della grandezza del dataset. (la linea tratteggiata rappresenta l'accuratezza stimata da un random guessing ovvero nell'indovinare casualmente la membership)

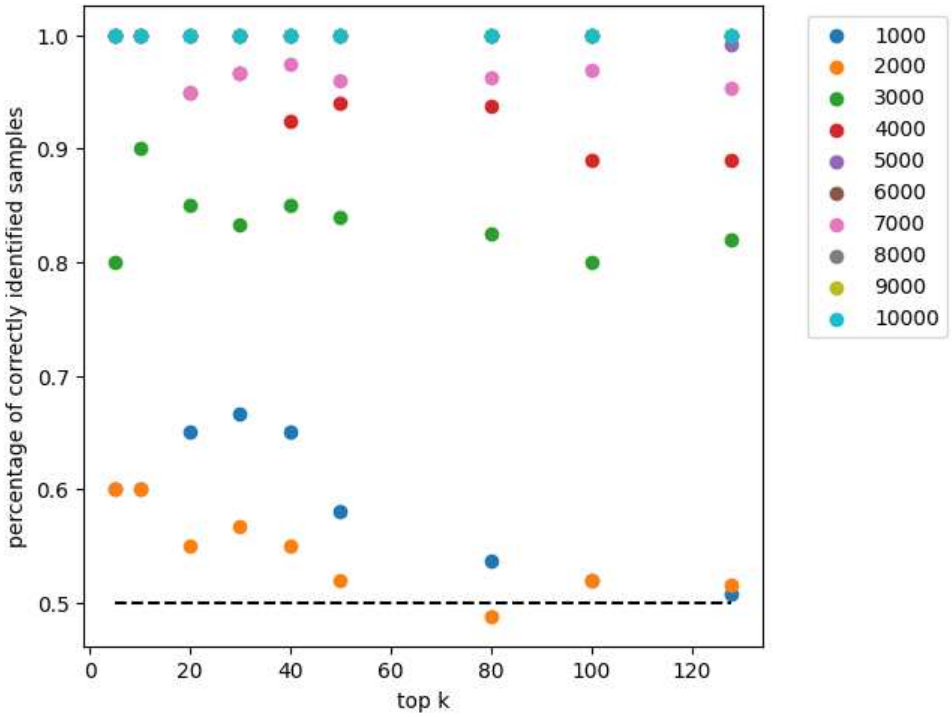


Fig 3: Precision in funzione di k

- Infine vengono generate 20000 immagini usate per calcolare la Frechet Inception Distance da un set di 20000 immagini reali. Una parte di queste immagini verrà utilizzata anche per il training della GAN shadow nell'attacco blackbox.

3.2 Struttura dell'attacco black box

Possiamo descrivere le varie fasi dell'attacco black box come segue:

- Definizione dell'architettura di generatore e discriminatore della GAN shadow tramite keras (API ad alto livello per tensorflow) e delle loro

training loss e training step (negli attacchi svolti è stata utilizzata la stessa architettura della GAN target)

- Importazione del dataset di training della GAN shadow ovvero le immagini generate dalla GAN target, se necessario viene ridotto il numero di immagini importate in base ai parametri dell'esperimento. La riduzione del numero di immagini generate importante è utile per determinare l'impatto del numero di immagini a disposizione sulla precisione dell'attacco.
- Importazione del dataset di training della GAN target, questo non viene utilizzato per il training della GAN shadow ma solo per valutare la precisione dell'attacco.
- Vengono definiti i parametri `MAX_EPOCHS`, che specifica il numero totale di epoche di training della GAN shadow , e `epochs_per_iter` che determina ogni quante epoche verrà eseguito un tentativo di attacco black box, questo serve a determinare l'impatto del numero di epoche di training della GAN shadow sul successo dell'attacco.
- Poi procede con un ciclo nel quale la GAN shadow viene addestrata per `epochs_per_iter` epoche e poi viene attaccata, si utilizza il discriminatore della GAN shadow appena addestrato per determinare le likelihood delle immagini nel training set e di quelle nel test set. Questo passaggio differenzia l'attacco black box da quello white box, infatti il discriminatore usato per determinare le likelihood non è quello originale ma quello locale addestrato dall'attaccante.
- Vengono poi fatti gli istogrammi delle likelihood ottenute come per l'attacco white box.
- Si ordinano i valori delle likelihood in modo decrescente e si calcola il rapporto (6).

---

4 Analisi dei risultati

4.1 Qualità delle immagini generate

Per prima cosa valutiamo la qualità delle immagini generate dalla GAN target riportiamo alcuni esempi di immagini:

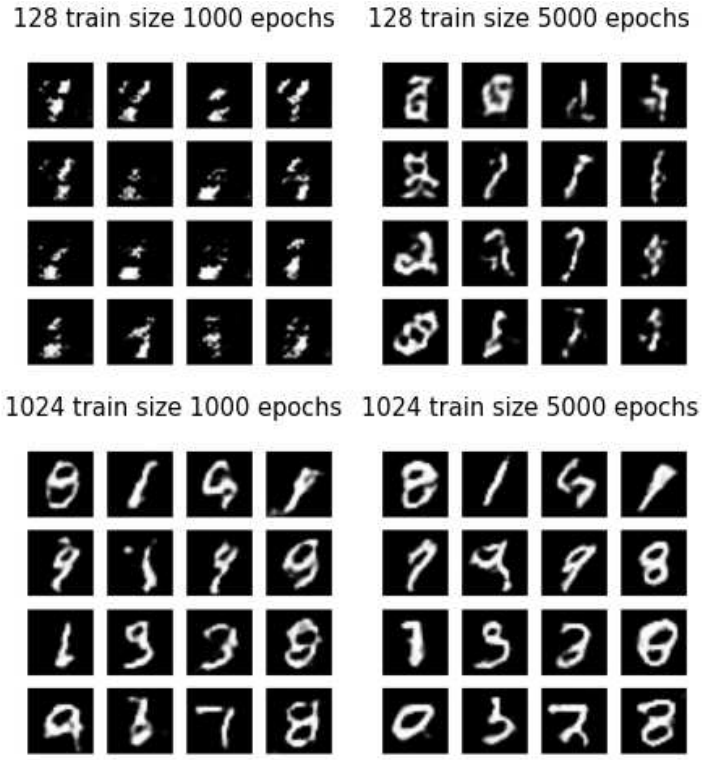


Fig 4: esempi di 64 immagini generate dalla GAN target con diverse grandezze del dataset e epoche di training

Da questi esempi si può notare che sia il numero di immagini nel train set che il numero di epoche di addestramento hanno un effetto sulla qualità dell'immagine. Si vede anche che nel caso si abbiano poche immagini è possibile ottenere una qualità accettabile aumentando il numero di epoche di training, come vedremo in seguito però questo porta la GAN target a essere molto vulnerabile ai M.I.A. perché le immagini generate sono troppo simili a quelle di training. Avere poche immagini di training è una situazione molto comune nelle GAN che producono immagini mediche, che desireremmo fossero

particolarmente resisenti a M.I.A. per la privacy dei pazienti.

Un modo oggettivo per valutare la qualità delle immagini generate è la Fréchet's Inception Distance (FID) che compara la distribuzione delle immagini generate a quella delle immagini reali e dà una "distanza" tra le due distribuzioni. Per essere calcolata si utilizza il modello InceptionV3 preaddestrato (i cui pesi sono disponibili da keras) che è un classificatore con 1000 classi a cui viene tolto l'ultimo layer, si prendono i valori dell'ultimo layer per le immagini generate e quelle reali, di cui si calcolano media e matrice di covarianza. La FID è infine determinata utilizzando la formula (7):

$$d_F(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu',\Sigma'))^2 = \|\mu'-\mu\|_2^2 + \mathrm{tr}(\Sigma+\Sigma'-2(\Sigma^{\frac{1}{2}}\cdot\Sigma'\cdot\Sigma^{\frac{1}{2}})^{\frac{1}{2}}) \quad (7)$$

dove μ, μ' sono i vettori media, Σ, Σ' le matrici di covarianza, tr la traccia della matrice e $\frac{1}{2}$ la radice quadrata estesa alle matrici.

Una FID bassa significa che i due set di immagini hanno una distribuzione simile ovvero che le immagini generate sono realistiche. In tutti le FID riportate in seguito sono state confrontate le distribuzioni delle immagini generate con quelle del test set in modo da non usare immagini già viste nel training dalla GAN, ciò infatti andrebbe a falsare i dati perché una GAN con overfitting otterrebbe una FID bassa.

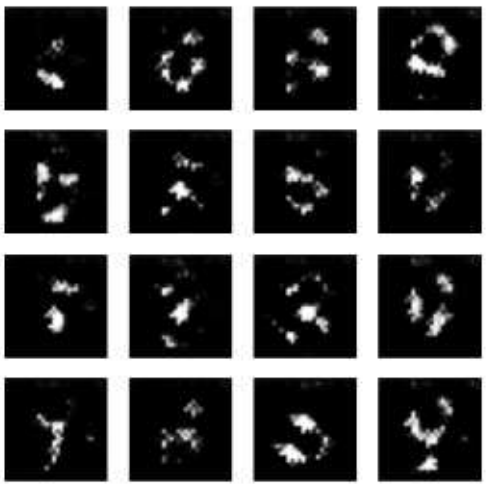


Fig 5: Immagine generata con FID superiore a 10^6, la qualità è molto bassa

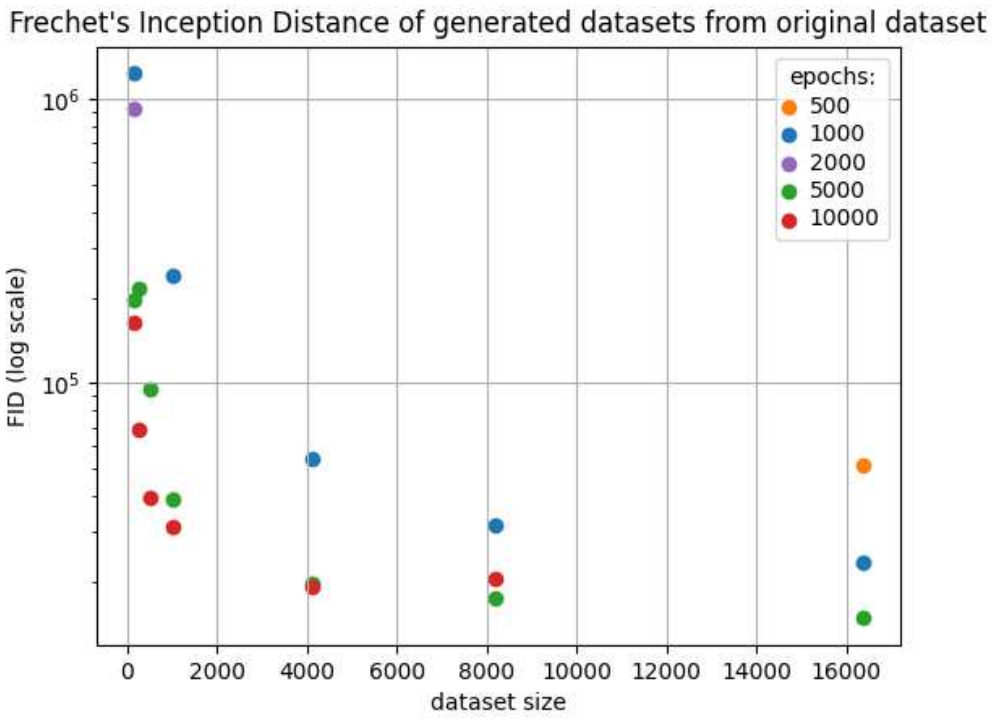


Fig 6: FIDs in funzione della training size

Dalla Fig 6 si nota come la grandezza del dataset abbia un grande effetto sulla FID (la scala delle ordinate è logaritmica) questo conferma ciò che vediamo dalla Fig 4 ovvero ingrandire la grandezza del training set migliora la qualità delle immagini. Anche aumentare il numero di epoche a parità di grandezza del training set abbassa la FID, ma per dataset troppo piccoli aumentare il numero di epoche dà un miglioramento minore rispetto ad aumentare la grandezza del dataset. Infatti dalla Fig 7 notiamo come passare da 128 immagini a 4096 migliori la qualità molto di più di quanto lo faccia passare da 1000 epoche a 5000 epoche di training (si vede anche dal grafico della FID in Fig 6).

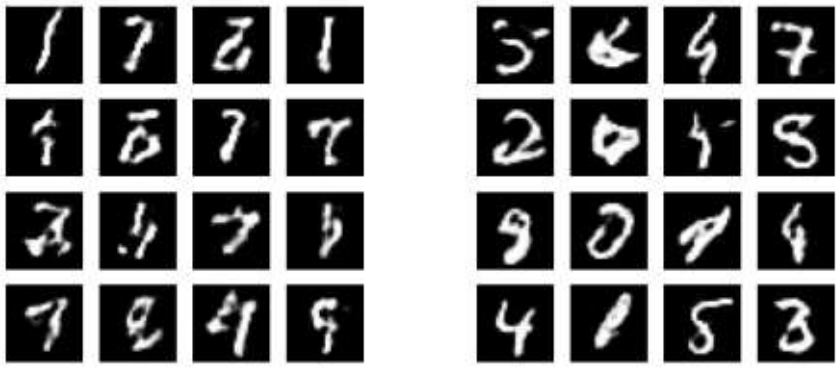


Fig 7: Immagini generate con 5000 epoche e 128 training size (a sinistra) e con 1000 epoche e 4096 training size (a destra)

4.2 Performance dell'attacco white box

4.2.1 Dataset con 128 immagini

4.2.1.1 Separazione delle likelihood tra immagini di train e test

Come accennato in precedenza nella sezione sulla struttura dell'attacco white box la separazione tra le distribuzioni delle likelihood delle immagini di test e di train fornisce informazioni sull'overfit del discriminatore della GAN. In questo caso le likelihood sono determinate dal discriminatore della GAN target, nell'attacco black box viene utilizzato il discriminatore della GAN shadow. Più il discriminatore presenta overfitting (e quindi più le due distribuzioni delle likelihood sono separate) più avrà successo l'attacco. Ora analizziamo alcuni istogrammi delle likelihood per il dataset con 128 immagini:

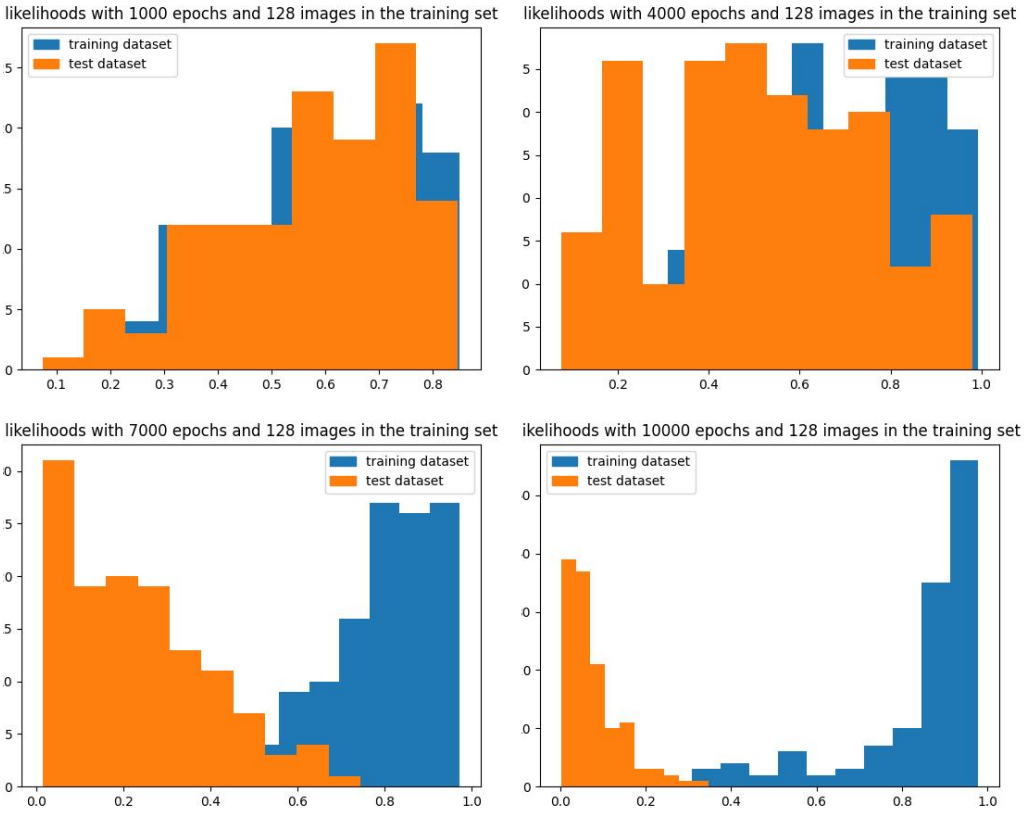


Fig 8: distribuzioni delle likelihood per diverse epoche di training con 128 immagini nel dataset

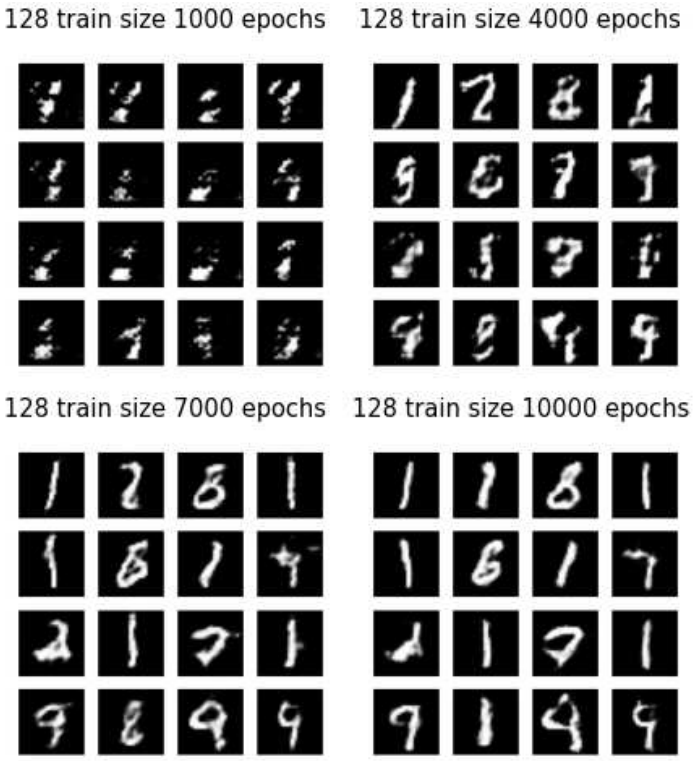


Fig 9: immagini generate corrispondenti alle likelihood della Fig 8

Dalla Fig 8 notiamo come aumentare il numero di epoche separi sempre di più le distribuzioni delle likelihood del training set da quelle del test set. Dalla Fig 9 si vede che aumentando il numero di epoche migliora la qualità dell'immagine, come abbiamo mostrato in precedenza anche con i valori della FID in funzione del numero di epoche. Quindi fissando il training set è presente un tradeoff tra la qualità delle immagini e quanto separate sono le likelihood di train e test set (ovvero l'overfitting del discriminatore sul train set).

4.2.1.2 Precisione nelle top k immagini per likelihood

Ora andiamo ad analizzare il valore della precisione dell'attacco quando prendiamo le prime top k immagini per likelihood (in un set con metà immagini presenti nel training set e metà presenti nel test set) e vediamo quante di queste appartengano al training set. Se la maggior parte delle immagini nelle top k era presente nel dataset l'attacco ha successo.

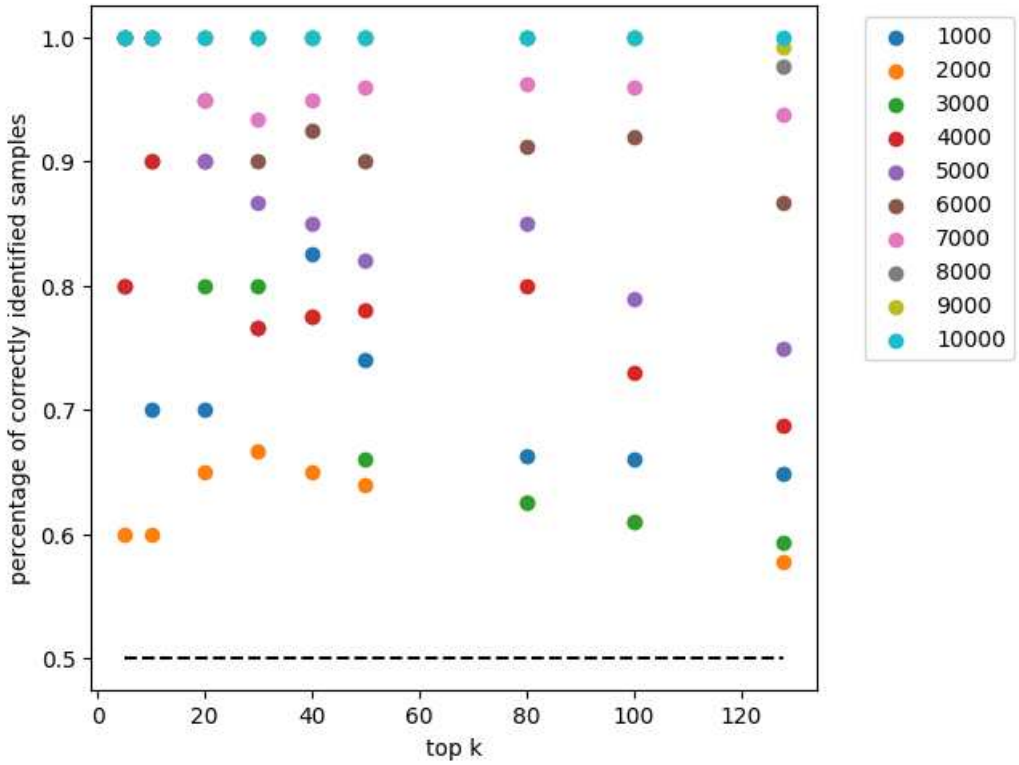


Fig 10: Precision dell'attacco in diverse top k immagini per likelihood e per diverso numero di epoche di training con un dataset di 128 immagini

Dalla Fig 10 vediamo come per il training set di 128 immagini la precisione dell'attacco white box sia sopra al 60% fin dalle prime 1000 epoche, e aumenti con l'aumentare delle epoche di training. Questo è dovuto al fatto che la GAN ha troppe poche immagini per generalizzare e quindi con l'aumentare del numero di epoche presenta overfit sempre maggiore sul training set, come si vede dalla separazione della distribuzione delle likelihood in Fig 8.

Inoltre si può notare in Fig 9 che le immagini generate all'epoca 1000 presentino ancora qualità molto bassa, quindi l'attacco ha successo prima ancora che le immagini diventino di qualità accettabile.

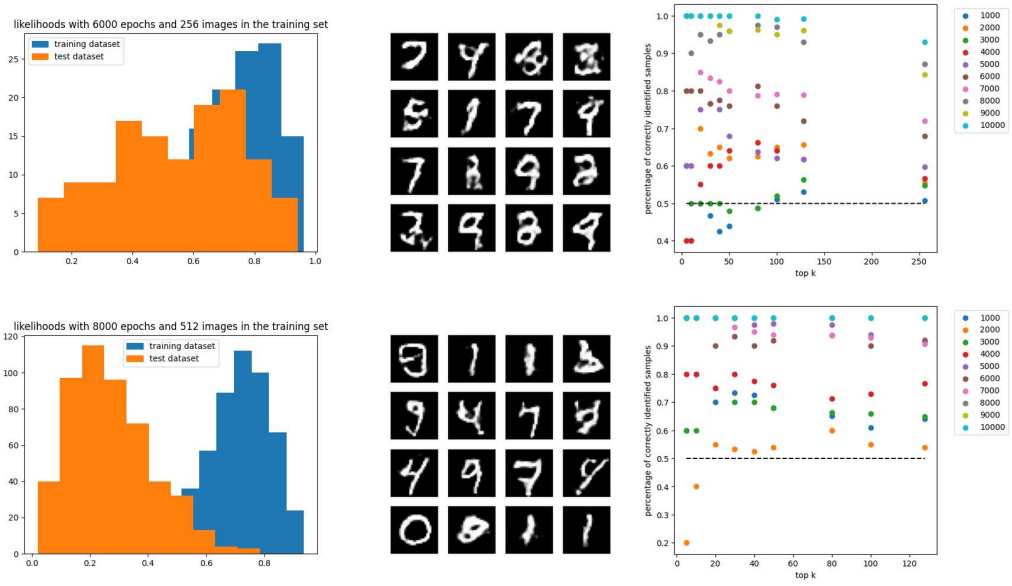
Osservando l'istogramma delle likelihood in Fig 8 vediamo come per l'epoch 1000 le due distribuzioni siano in gran parte sovrapposte ma quella di train sia leggermente spostata a destra. Questo implica che prendendo le top k stiamo selezionando la coda a destra della distribuzione di probabilità che è

in gran parte formata da immagini di training. Da ciò si capisce che per il successo dell'attaccco basta che la distribuzione delle likelihood del training set sia leggermente spostata a destra e non serve che sia completamente separata. (Una likelihood completamente separata porta ad avere una precisione del 100% per ogni k)

Infine si può notare dal grafico in Fig 10 come per k più grandi, a parità di epoche, la precisione tenda a scendere. Questo è dovuto al fatto che con un k più grande stiamo considerando una porzione maggiore della coda destra della distribuzione delle likelihood, quindi a parità di separazione tra gli istogrammi stiamo includendo un maggior numero di immagini del test set nelle top k.

4.2.2 Dataset con un maggior numero di immagini

Aumentando il training set a 256, 512, 1024, 4096, 8192 immagini possiamo confermare le stesse osservazioni fatte per il training set di 128, ovvero il numero di epoche richiesto per ottenere immagini di qualità sufficiente porta ad overfitting del discriminatore. Riporto in seguito per queste grandezze del training set la prima epoca nella quale si possono distinguere i numeri visivamente, un immagine di 16 esempi generati in quell'epoca, la separazione delle likelihood per quell'epoca e il grafico della precisione in funzione di k.



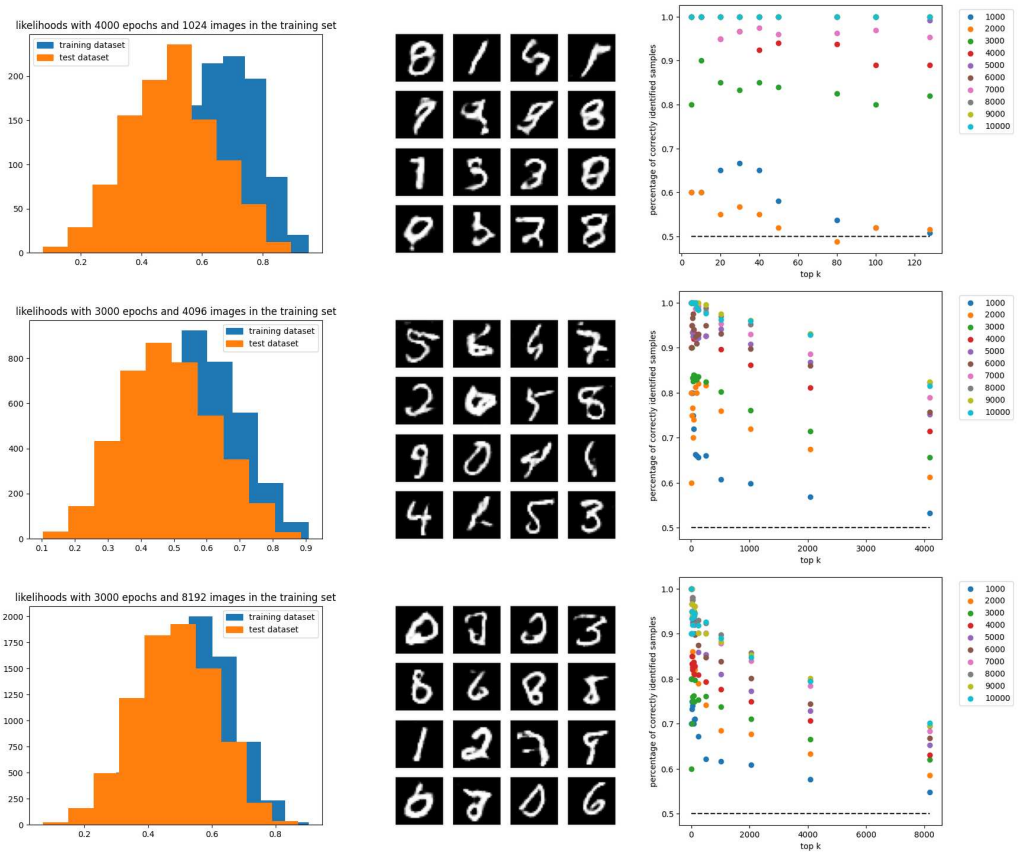


Fig 11: Grandezze del training set per cui non è possibile addestrare una GAN che resista agli attacchi whitebox generando immagini di qualità accettabile

Per poter ottenere una GAN resistente all'attacco white box che produca immagini distinguibili è necessario avere un dataset di almeno 16384 immagini infatti:

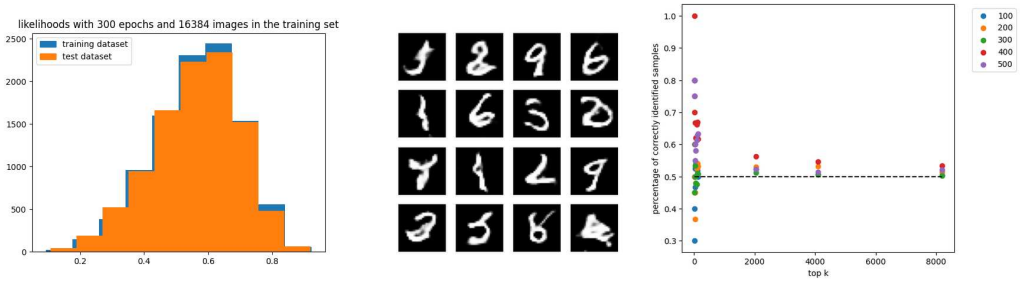


Fig 12: GAN che resiste agli attacchi whitebox generando immagini di qualità accettabile

In Fig 12 vediamo come la GAN con training size di 16384 dopo 300 epochs di training riesca a produrre delle cifre distinguibili ma le distribuzioni delle likelihood sono sovrapposte e la precisione per la serie 300 epochs (in verde) è per la maggior parte delle k al di sotto del 50%.

Aumentando il numero di epoche si può migliorare la qualità delle immagini di questa GAN ma di nuovo tornerebbe ad essere vulnerabile agli attacchi white box.

4.2.3 Precisione in funzione della FID

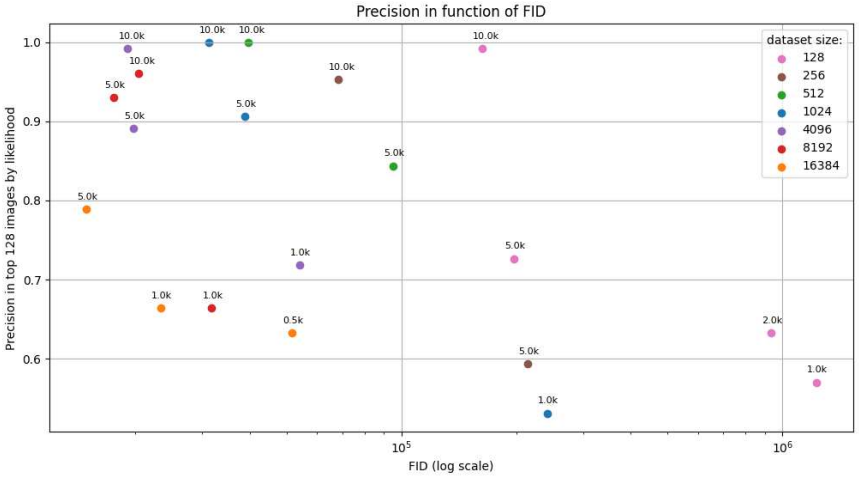


Fig 13: Precisione dei vari attacchi per k=128 in funzione della FID, annotate di fianco ai data points ci sono il numero di epoche di training. L'asse della FID è in scala logaritmica.

Dalla Fig 13 si vede come fissato il numero di immagini nel training set aumentare le epoche diminuisca la FID (migliora la qualità delle immagini) ma aumenti la precisione dell'attacco.

Questo significa che avendo a disposizione un certo numero (fissato) di immagini per il training di una GAN non è possibile migliorare la qualità delle immagini generate senza renderla più vulnerabile a M.I.A.

Nella pratica chi addestra la GAN target desidererebbe che si trovasse

in basso a sinistra in questo grafico Precisione/FID ovvero resistente agli attacchi e con immagini di alta qualità.

Osservando le immagini generate è possibile decidere una soglia massima della FID, ad esempio selezionando una FID di soglia di ≈ 170000 abbiamo che le immagini in Fig 14 nella riga superiore sono considerate di qualità accetabile quelle in basso superano la soglia della FID massima per essere considerate buone.

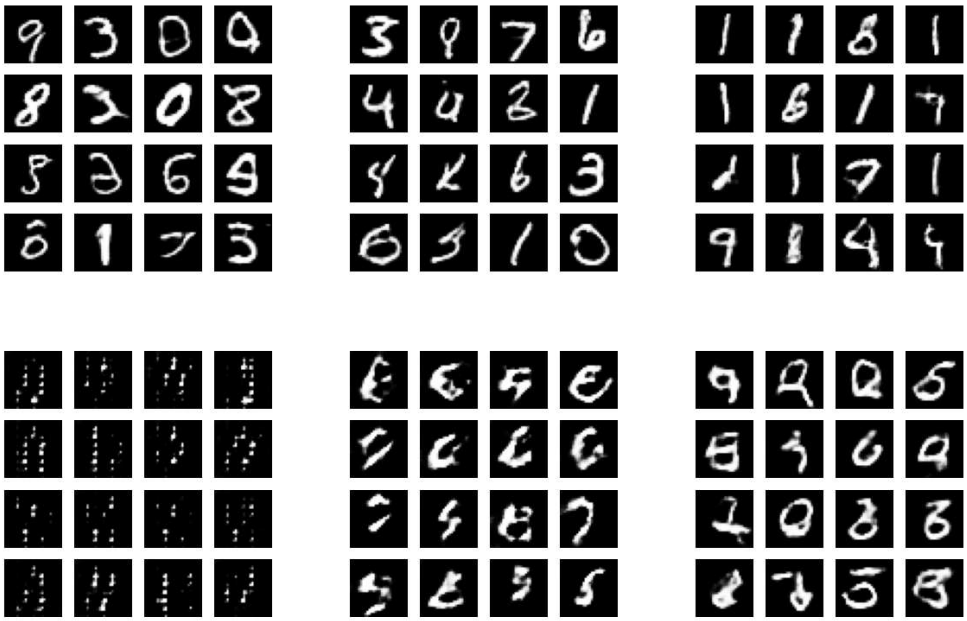


Fig 14: Usando una soglia di circa 170000 le immagini nella riga superiore sono giudicate di qualità accettabile quelle sotto di qualità insufficiente

Dalla Fig 13 si comprende anche l'importanza di aver un sufficiente numero di immagini nel dataset di training.

Infatti a parità di FID le GAN con un dataset più grande sono più resistenti ai M.I.A. Ad esempio per $FID \in [20000, 35000]$ gli attacchi contro le GAN 8192 e 16384 hanno precisione di circa 67% mentre per i dataset più piccoli la precisione è oltre il 90%.

4.3 Performance attacco black box

4.3.1 Shadow training set con la disponibilità di 6000 generate e 1000 epoche di training della shadow GAN

Inizialmente fissiamo il numero di immagini generate dalla GAN target disponibili per l'attacco black box a 6000 e il numero di epoche della shadow GAN a 1000. In seguito studieremo l'effetto del numero di immagini generate e delle epoche della shadow GAN sulla precisione dell'attacco.

4.3.1.1 Separazione delle likelihood tra training e test set e precisione nelle top k Come nell'attacco white box si può utilizzare la separazione delle distribuzioni delle likelihood per visualizzare la facilità dell'attacco. Nel caso di attacco black box però le likelihood non vengono determinate dal discriminatore della GAN target ma da quello della GAN shadow.

4.3.1.1.1 Effetto della grandezza del training set della GAN target Fissato il numero di epoche di training della GAN target a 10000 andiamo ad analizzare come la precisione M.I.A. black box sia influenzata dal numero di immagini nel dataset della GAN target.

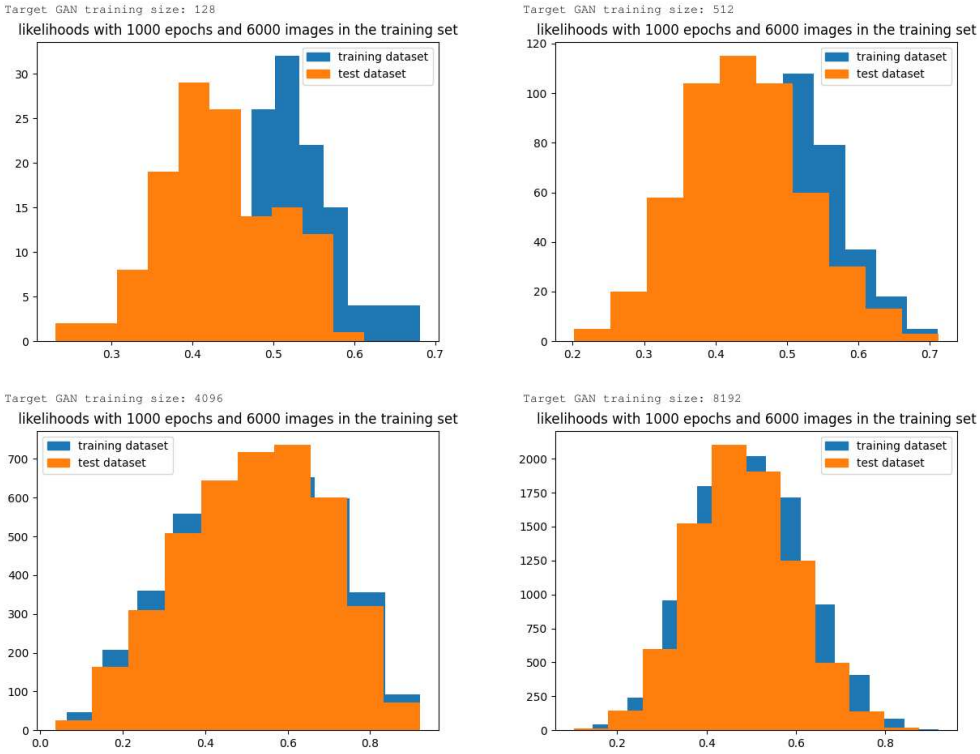


Fig 15: Istogrammi delle likelihoods dell'attacco black box per training set della GAN target di 128, 512, 4096, 8192 immagini

Con l'aumentare del numero di immagini nel training set della GAN target vediamo dalla Fig 15 come le likelihood del training e test set determinate dal discriminatore shadow siano sempre più sovrapposte. Quindi aumentare il numero di immagini nel dataset originale non solo migliora la qualità delle immagini generate ma anche rende più resistente la GAN target a attacchi sia white box che black box.

Quest'ultima affermazione è confermata anche dal grafico in Fig 16 in cui si vede che aumentando la grandezza del dataset di training della target GAN la precisione dell'attacco black box tende a diminuire come succede per gli attacchi white box.

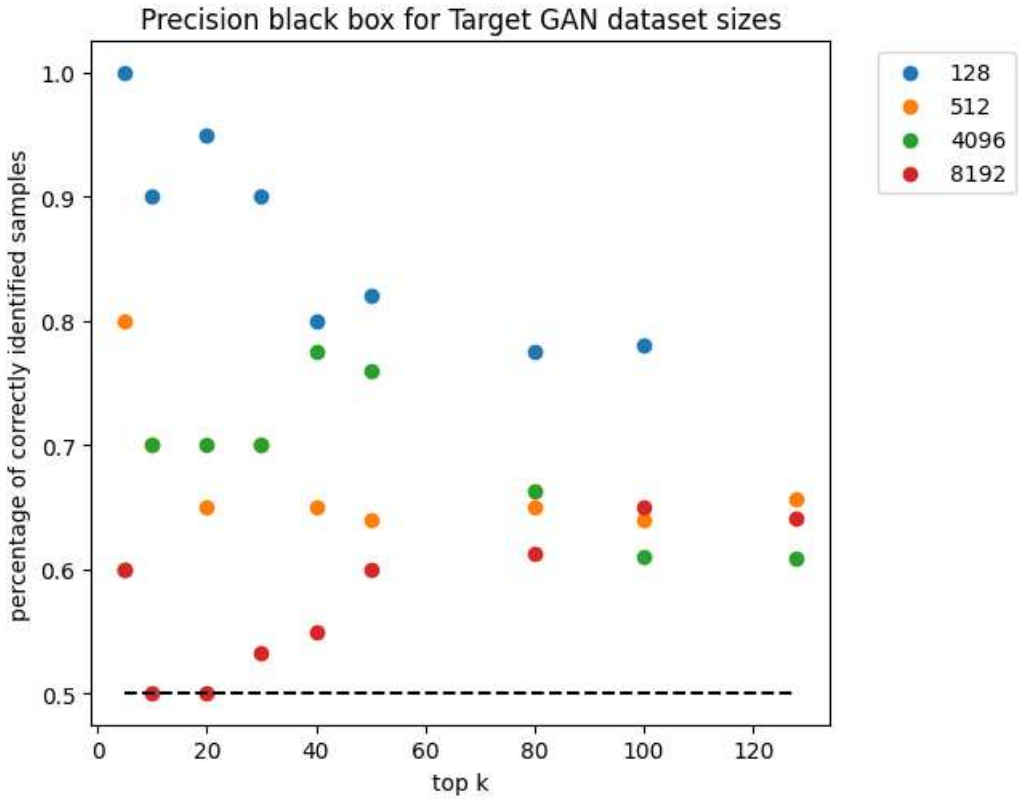


Fig 16: Precisione di alcuni attacchi black box tenendo fisso: numero di immagini generate (6000), numero di epoche di training di target (10000) e shadow (1000) GAN

4.3.1.1.2 Effetto del numero di epoche di training della GAN target Un'altra variabile che gioca un ruolo importante nella precisione degli attacchi white box è il numero di epoche di training della GAN target, andiamo ad analizzare se questo vale anche per gli attacchi black box.

GAN 16384

Fissati il numero di immagini nel training set della GAN target a 16384, il numero di immagini generate a 6000 e numero di epoche della GAN shadow a 1000 possiamo procedere con la analisi delle likelihood quando la GAN target viene addestrata per 500, 1000 o 5000 epoche.

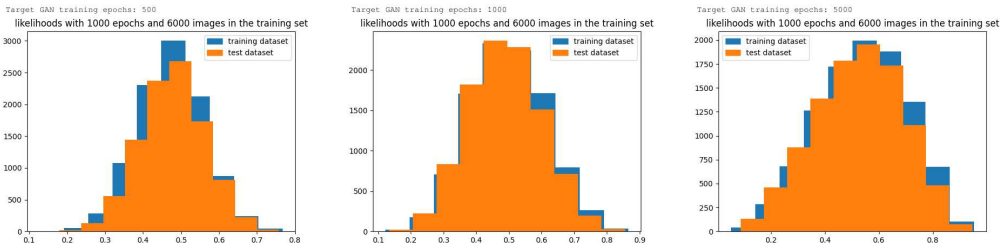


Fig 17: Istogrammi delle likelihoods dell'attacco black box con numero di epoche della GAN target uguali a 500, 1000, 5000

Dalla Fig 17 è possibile vedere come aumentando il numero di epoche di training della target la likelihood del training set (in blu) si sposti più a destra rispetto a quella del test set (in arancione), visivamente non sembra un grande spostamento ma osservando anche l'andamento della precisione in Fig 18 si vede come per k=128 si passi da una precisione del 50% per le 500 epochs (ovvero uguale al random guessing) a quella del 66% delle 5000 epochs.

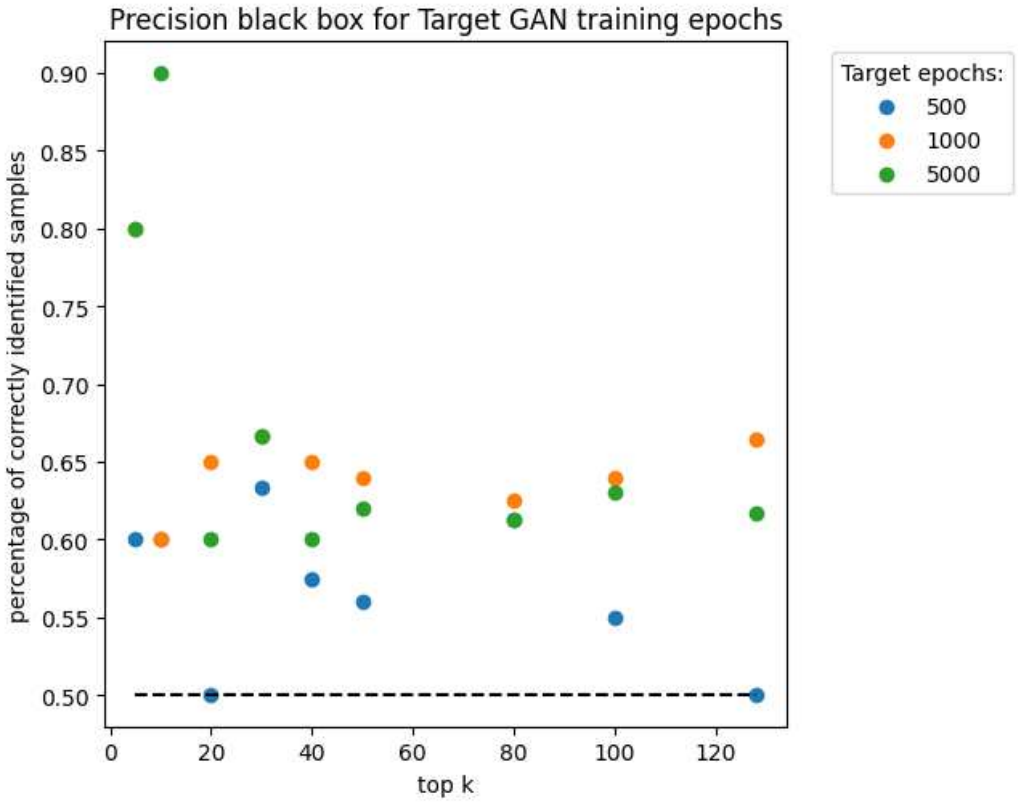


Fig 18: Precisione di alcuni attacchi black box tenendo fisso: numero di immagini generate (6000), training set target (16384) e epoche della shadow (1000) GAN

Quindi per la maggior parte dei valori di k gli attacchi sulle GAN con 1000 e 5000 epoche hanno precisione tra il $60\% - 65\%$ (a parte qualche picco sopra 80% per k bassi) mentre per la GAN con 500 epoche l'attacco black box fallisce in quanto è al 50% di precisione o poco più per quasi tutti i k. Questo non ci sorprende in quanto la GAN con 500 epoche di training e 16384 immagini nel dataset era quella che riusciva in parte a resistere ad attacchi white box (producendo immagini di qualità accettabile).

GAN 128

Andiamo ad analizzare anche i risultati di attacco black box in funzione delle epoche della GAN target per una dataset size fissata stavolta a 128. Questa volta le epoche di training sono: 1000, 2000, 5000, 10000.

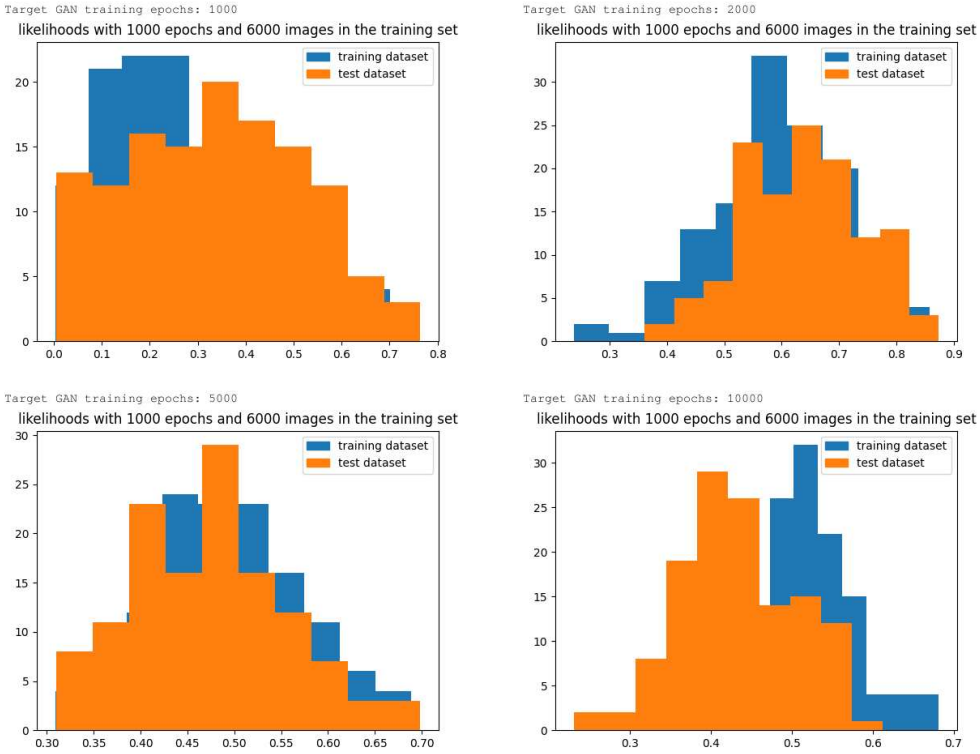


Fig 19: Istogrammi delle likelihoods dell'attacco black box con numero di epoche della GAN target uguali a 1000, 2000, 5000, 10000

Dalla Fig 19 è possibile vedere come aumentando il numero di epoche di training della target la likelihood del training set (in blu) si sposti più a destra rispetto a quella del test set (in arancione), lo spostamento per la GAN 128 è visibile più che nella GAN 16384, questo probabilmente è dovuto al fatto che avendo poche immagini l'overfitting sul training set avviene più velocemente.

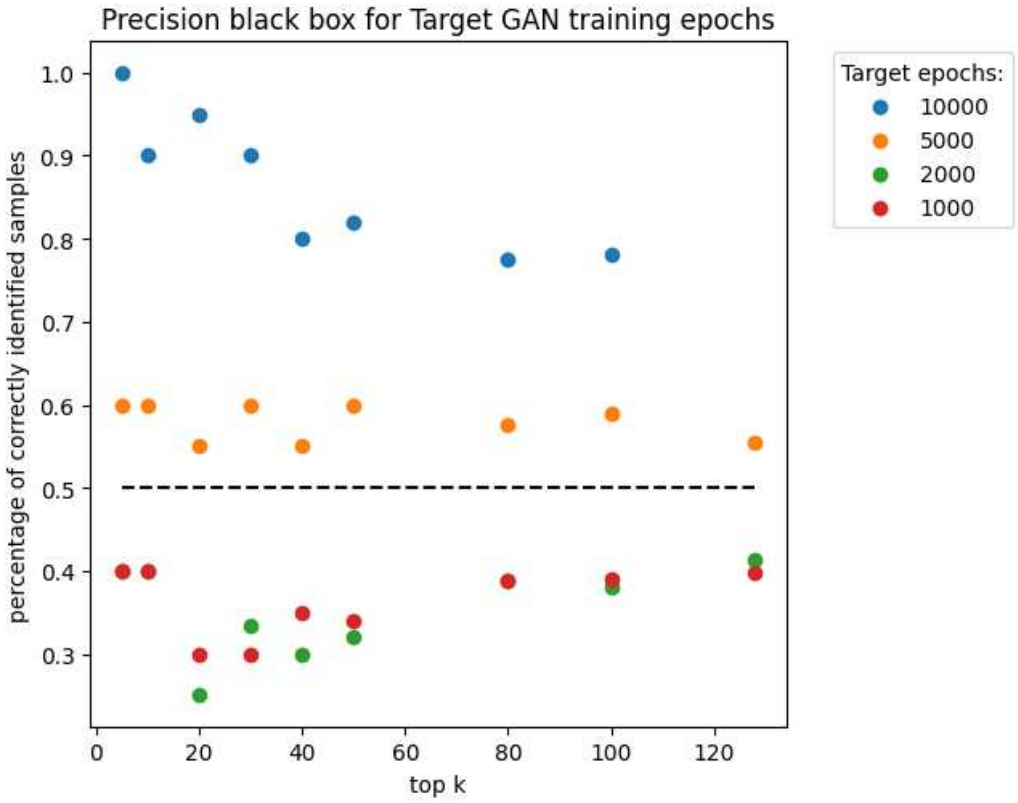


Fig 20: Precisione di alcuni attacchi black box tenendo fisso: numero di immagini generate (6000), training set target (128) e epoche della shadow (1000) GAN

Dalla Fig 20 vediamo inoltre come per 10000 epochs di training della target l'attacco black box ha successo con precisione che va dal 80% per k maggiori di 32 a fin sopra del 90% per k più bassi. Per 5000 epochs la precisione si aggira intorno al 60% per tutti i kmentre per 2000 e 1000 epochs è minore del 40% (quindi è peggio di random guessing). Questo sembrerebbe in contraddizione con quanto sostenuto in precedenza, ovvero che per avere una GAN resistente a M.I.A. serve un dataset grande (come la GAN 16384), ma bisogna ricordare che nella Fig 13 queste due GAN con 1000 e 2000 epoche e 128 immagini di training avevano una $FID \approx 10^6$.

Infatti andando a vedere le immagini generate dalle GAN in Fig 21 vediamo come la qualità sia molto bassa e non si possa distinguere nessuna cifra.

Da questa osservazione possiamo aggiungere un'altra interpretazione al grafico in Fig 13 e dividere il grafico in 3 zone:

- Nell'angolo in basso a sinistra del grafico abbiamo le GAN che generalizzano bene e quindi creano buone immagini e resistono ai M.I.A.
- Nell'angolo in basso a destra si hanno le GAN che resistono agli M.I.A. perché generano immagini di qualità troppo bassa per poter fare un attacco di inferenza. Queste GAN non sono utili nella pratica.
- Nella zona superiore del grafico si trovano tutte le GAN su cui funzionano gli attacchi di inferenza, sia quelle che generano immagini ad alta qualità che quelle mediocri.

Negli esperimenti eseguiti non sono state trovate GAN tali che siano sia vulnerabili a M.I.A. e che generino immagini di qualità molto bassa ($FID \approx 10^6$) quindi l'angolo in alto a sinistra della Fig 13 è vuoto.

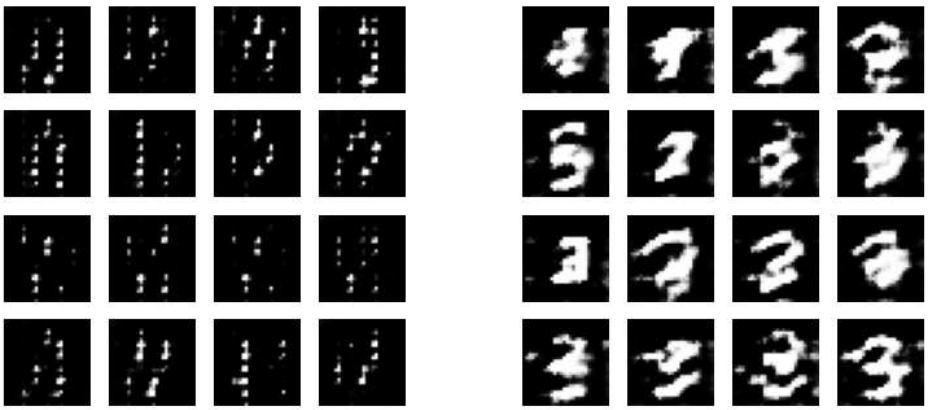


Fig 21: Immagini generate dalla GAN target con 128 immagini nel training set e 1000 epoche (sinistra) 2000 epoche (destra)

4.3.2 Effetto del numero di epoche di training della shadow GAN e numero di immagini generate disponibili

Le variabili analizzate in precedenza (numero di epoche e training size della GAN target) sono sotto il controllo di chi progetta la GAN target e quindi

l'attacante non può intervenire per cambiarne il valore.

Le due principali risorse dell'attaccante sono il numero di immagini generate dalla GAN target per il training della GAN shadow e per quante epoche quest'ultima viene addestrata. Andiamo ad analizzare come queste due varibili possano venire sfruttate e l'effetto che hanno sulla precisione.

4.3.2.1 Numero di epoche shadow GAN Fissiamo il numero di immagini generate dalla target GAN disponibili a 6000, e GAN target con 4096 immagini di training e 10000 epochs in modo da analizzare solo l'effetto del numero di epoche della shadow GAN.

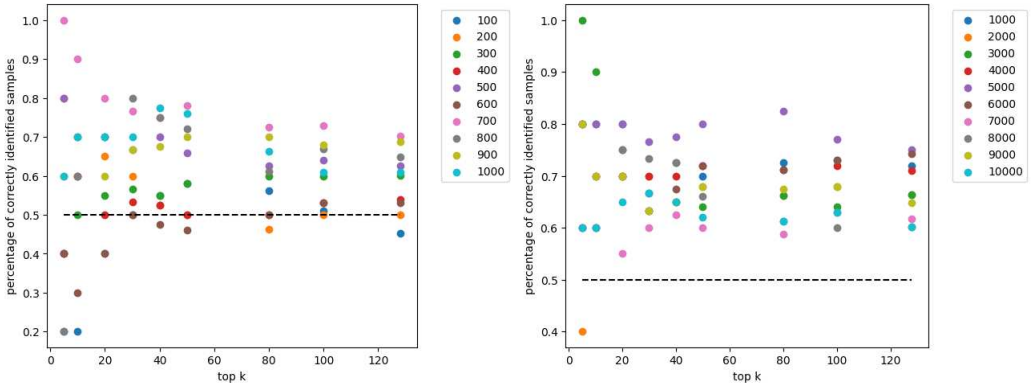


Fig 22: Precisione degli attacchi sulla GAN target 4096 con 10000 epochs di training, 6000 immagini generate e varie epoche di training della shadow GAN

Dalla Fig 22 vediamo come aumentare il numero di epoche della shadow GAN da 100 a 700 migliori molto la precisione dell'attacco, da una precisione minore del 50% a circa 80% ma andare oltre le 700 epoche non porta alcun miglioramento in quanto la precisione rimane nell'intervallo tra circa $[60\%, 80\%]$ per le epoche da 800 a 10000. Una interpretazione di questi risultati è che a 700 epoche la shadow GAN ha estratto tutte le informazioni sul training set della target presenti nelle immagini generate e le oscillazioni successive tra 60% e 80% sono dovute all'aleatorità del training.

Quindi si può concludere che il numero di epoche di training della shadow GAN è una risorsa per l'attaccante finché non raggiunge una certa precisione, che non può migliorare continuando ad aumentare le epoche di training.

Il numero di epoche per ottenere questa precisione massima però non è

conosciuto all'attaccante quindi una scelta intelligente può essere quella di impostare un numero di epoche alto per il training della shadow (ad esempio 10000) per essere sicuri che abbia raggiunto la soglia di epoche dopo la quale la precisione non aumenta più. Utilizzando questa tecnica una parte delle epoche di training potrebbe essere inutile ma l'attaccante può essere ragionevolmente sicuro di non aver fermato il training troppo presto.

In uno sviluppo futuro potrebbe essere utile cercare se sia possibile utilizzare FID tra le immagini generate dalla target e quelle generate dalla shadow per determinare dopo quante epoche fermare il train della shadow GAN.

4.3.2.2 Shadow training set con diversi numeri di immagini generate disponibili

Fissiamo la GAN target a 4096 immagini di training e 10000 epochs, il numero di epoche di training della shadow GAN a 1000. Analizziamo l'effetto di avere a disposizione un maggior numero di immagini generate dalla GAN target:

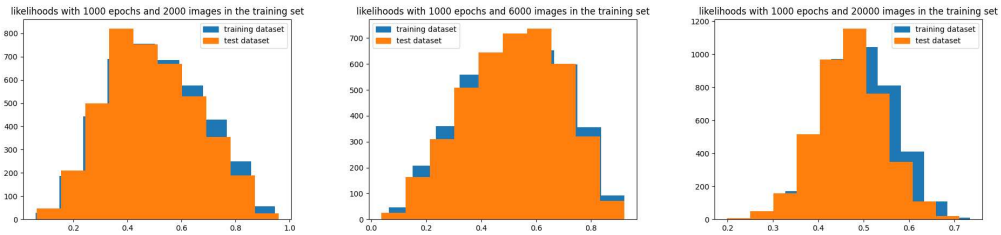


Fig 23: Separazione delle likelihoods del discriminatore della shadow GAN con: 4096 immagini nel training set della GAN target addestrata per 10000 epoche, 1000 epoche di training della shadow GAN. Da sinistra a destra il numero immagini generate è 2000, 6000 e 20000

Dalla Fig 23 vediamo come aumentare il numero di immagini generate utilizzate nel training della GAN shadow permetta di ottenere una separazione maggiore delle likelihood. Infatti per 2000 immagini la coda destra della likelihood di test è più grande di quella di training, per 6000 immagini si vede che la coda destra ha più immagini del training set che del test set, mentre per 20000 immagini le due distribuzioni sono chiaramente separate.

Guardando anche i dati sulla precisione in Fig 24 si nota come la precisione migliora avendo a disposizione più immagini generate, infatti l'attacco con 2000 immagini a disposizione fallisce, la sovrapposizione delle likelihood fa sì

che molte delle immagini con likelihood alta siano del test set. Per l'attacco con 6000 immagini la precisione è circa nell'intervallo $[60\%, 75\%]$ mentre con 20000 immagini migliora ulteriormente ed è sopra il 70% per tutti i valori di k.

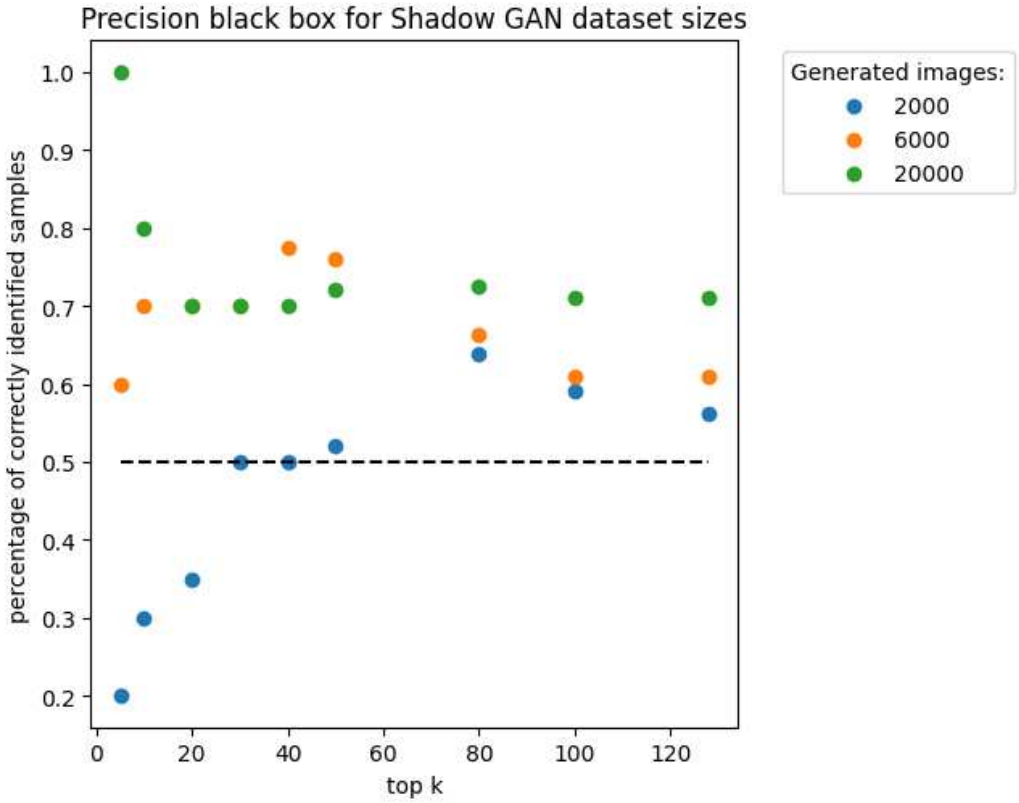


Fig 24: Precisione degli attacchi black box con: 4096 immagini nel training set della GAN target addestrata per 10000 epoche, 1000 epoche di training della shadow GAN. Il numero immagini generate per le tre serie nel grafico è 2000, 6000 e 20000

4.3.2.2.1 È possibile compensare un basso numero di immagini generate con un maggior numero di epoche? Abbiamo visto come per il caso in cui l'attacante abbia a disposizione solo 2000 immagini generate l'attacco fallisce, ora analizziamo se sia possibile addestrare la shadow GAN per un maggior numero di epoche e raggiungere una precisione maggior anche con questo attacco.

Questa situazione potrebbe presentarsi nella pratica perché le GAN target online a cui si può accedere tramite API spesso hanno un limite di richieste massimo oppure un prezzo associato ad una richiesta. Quindi l'attaccante può non avere accesso ad un gran numero di immagini generate (oppure potrebbe non essere economicamente sostenibile).

Prendendo la stessa GAN4096 analizzata con 2000 immagini generate e addestrando la GAN shadow corrispondente per 10000 epoche al posto di fermarsi dopo 1000 vediamo se la precisione dell'attacco migliora.

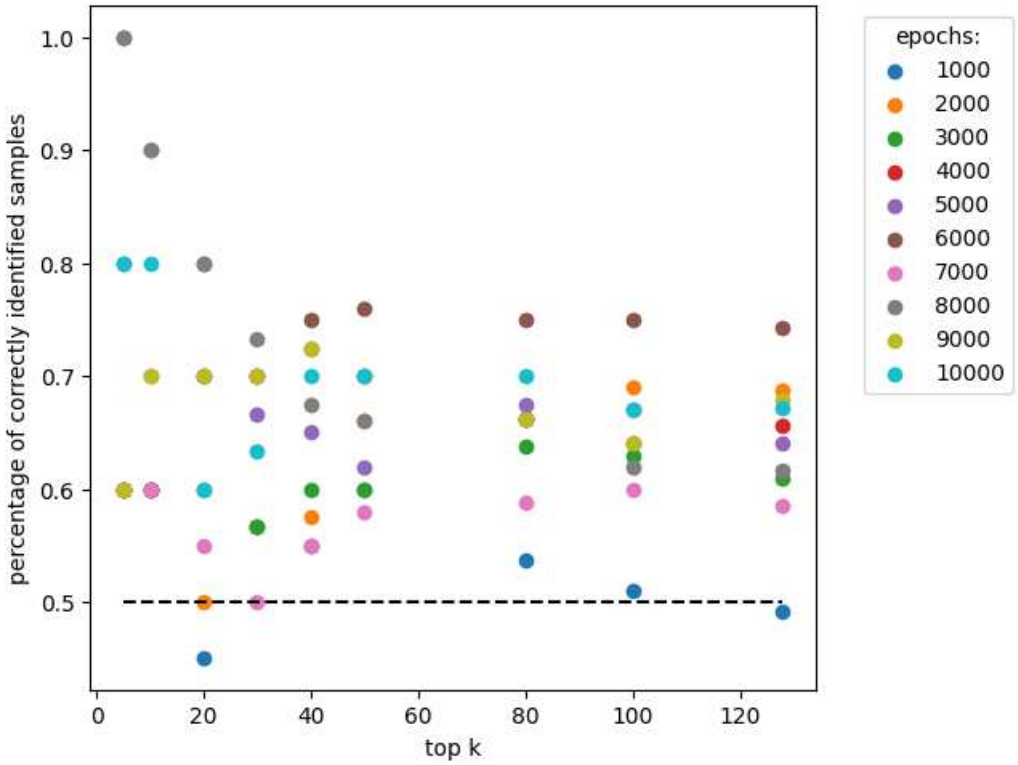


Fig 25: Precisione degli attacchi black box con: 4096 immagini nel training set della GAN target addestrata per 10000 epoche, e epoche di training della shadow GAN da 1000 a 10000

Dalla Fig 25 si vede che la precisione dell'attacco migliora aumentando il numero di epoche di training, a partire da 8000 epochs in poi la precisione è sempre superiore al 60%, quindi è possibile rendere un successo l'attacco che con 1000 epoche era fallito. Comunque non è possibile raggiungere una

precisione maggiore del 70% per ogni k come con 20000 immagini e 1000 epoche.

Da questi risultati si può concludere che sia il numero di immagini generate disponibili che il numero di epoche di training della GAN shadow sono una risorsa a disposizone dell'attaccante. Il numero di epoche di training della GAN shadow può essere aumentato fino ad un certo valore prima che la precisione saturi e rimanga nello stesso intervallo per ulteriori epoche di training. Aumentare il numero di immagini generate permette di migliorare molto la precisione ma potrebbe essere difficile per l'attaccante in caso di limiti sul numero massimo di richieste o un costo per richiesta (nel caso la GAN target sia accessibile tramite una API). Inoltre si può compensare un basso numero di immagini generate disponibili con un maggior numero di epoche ma la precisione ottenuta sarà più bassa rispetto a quella con un dataset a disposizione più grande.

4.3.3 Precisione in funzione della FID degli attacchi black box

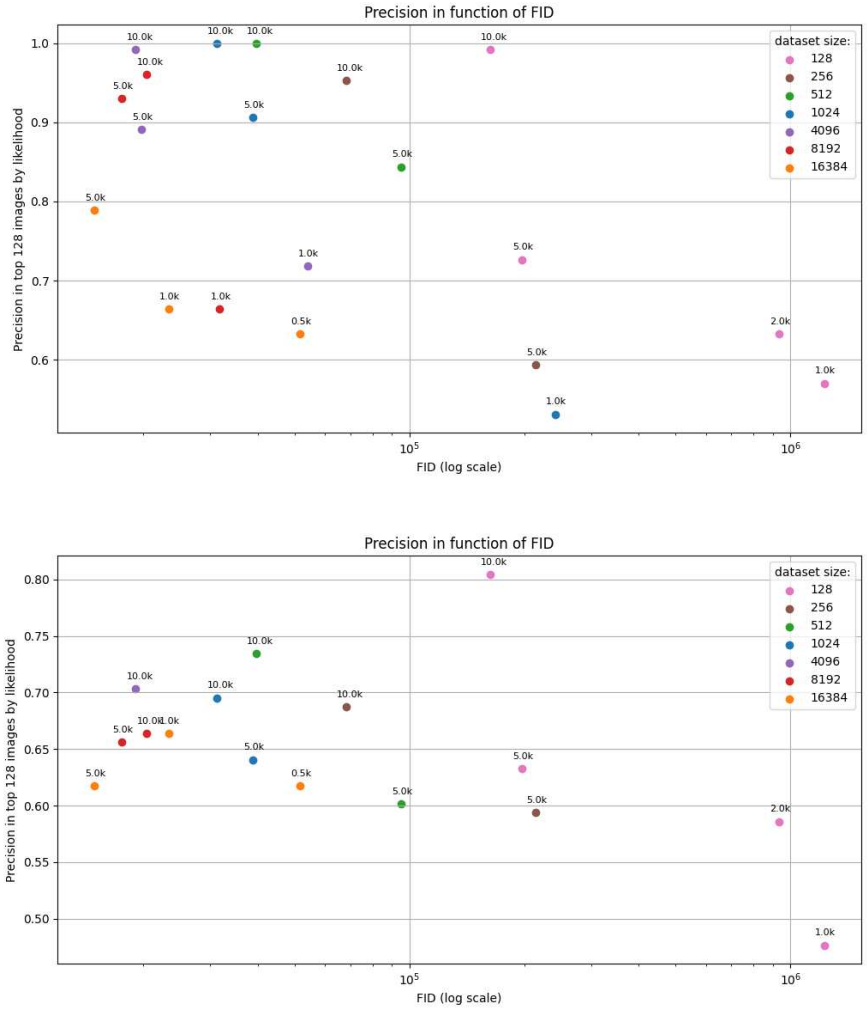


Fig 26: Precisione dei vari attacchi white box (sopra) e black box (sotto) per k=128 in funzione della FID, annotate di fianco ai data points ci sono il numero di epoche di training della GAN target. L'asse della FID è in scala logaritmica.

Dalla Fig 26 vediamo come il grafico della precisione degli attacchi black box in funzione della FID sia molto simile a quello della precisione degli attacchi white box, con la differenza principale che tutti i punti sono traslati

in basso, ovvero per la stessa GAN target l'attacco black box è meno preciso che quello white box.

Questa osservazione è consona con l'ipotesi l'attacco abbia successo grazie all'overfitting del discriminatore della target GAN per effettuare l'attacco infatti: nell'attacco white box si ha direttamente accesso al discriminatore mentre in quello black box si ottengono delle informazioni indirettamente tramite le immagini generate.

Le osservazioni fatte per la precisione dell'attacco white box in funzione della FID valgono ancora:

- Avendo a disposizione un numero fissato di immagini per il training della GAN target non è possibile aumentare la qualità delle immagini senza renderla più vulnerabile a attacchi black box. Questo vale perché aumentando il numero di epoche di training si abbassa la FID ma si migliora la precisione degli attacchi black box.
- A parità di FID le GAN con training dataset più grandi sono più resistenti agli attacchi black box.

Da queste osservazioni possiamo concludere che una GAN resistente ad attacchi white box sarà anche resistente ad attacchi black box.

4.4 Attacchi sul dataset Anime Faces

Alcuni degli attacchi effettuati sul dataset MNIST sono stati ripetuti sul dataset Anime Faces di Kaggle [7] per poter studiare la differenza nai M.I.A. su un dataset diverso con immagini di dimensione maggiore.

Per poter utilizzare la GAN con immagini 64x64 è stato adattato il numero di neuroni per layer sia per il generatore che per il discriminatore della GAN:

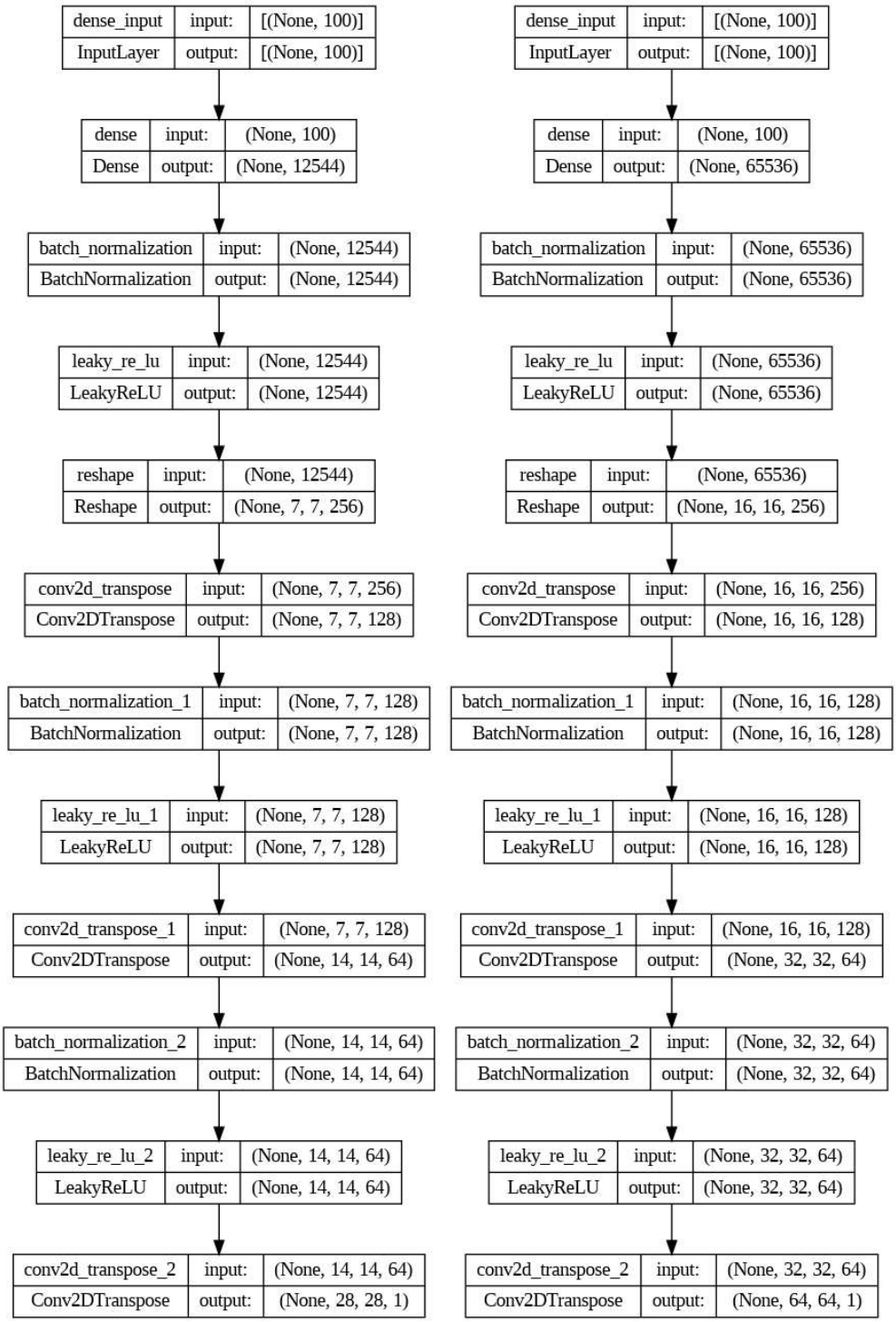

dense_input | input: | [(None, 100)]
InputLayer | output: | [(None, 100)]
dense | input: | (None, 100)
Dense | output: | (None, 12544)
batch_normalization | input: | (None, 12544)
BatchNormalization | output: | (None, 12544)
leaky_re_lu | input: | (None, 12544)
LeakyReLU | output: | (None, 12544)
reshape | input: | (None, 12544)
Reshape | output: | (None, 7, 7, 256)
conv2d_transpose | input: | (None, 7, 7, 256)
Conv2DTranspose | output: | (None, 7, 7, 128)
batch_normalization_1 | input: | (None, 7, 7, 128)
BatchNormalization | output: | (None, 7, 7, 128)
leaky_re_lu_1 | input: | (None, 7, 7, 128)
LeakyReLU | output: | (None, 7, 7, 128)
conv2d_transpose_1 | input: | (None, 7, 7, 128)
Conv2DTranspose | output: | (None, 14, 14, 64)
batch_normalization_2 | input: | (None, 14, 14, 64)
BatchNormalization | output: | (None, 14, 14, 64)
leaky_re_lu_2 | input: | (None, 14, 14, 64)
LeakyReLU | output: | (None, 14, 14, 64)
conv2d_transpose_2 | input: | (None, 14, 14, 64)
Conv2DTranspose | output: | (None, 28, 28, 1)
dense_input | input: | [(None, 100)]
InputLayer | output: | [(None, 100)]
dense | input: | (None, 100)
Dense | output: | (None, 65536)
batch_normalization | input: | (None, 65536)
BatchNormalization | output: | (None, 65536)
leaky_re_lu | input: | (None, 65536)
LeakyReLU | output: | (None, 65536)
reshape | input: | (None, 65536)
Reshape | output: | (None, 16, 16, 256)
conv2d_transpose | input: | (None, 16, 16, 256)
Conv2DTranspose | output: | (None, 16, 16, 128)
batch_normalization_1 | input: | (None, 16, 16, 128)
BatchNormalization | output: | (None, 16, 16, 128)
leaky_re_lu_1 | input: | (None, 16, 16, 128)
LeakyReLU | output: | (None, 16, 16, 128)
conv2d_transpose_1 | input: | (None, 16, 16, 128)
Conv2DTranspose | output: | (None, 32, 32, 64)
batch_normalization_2 | input: | (None, 32, 32, 64)
BatchNormalization | output: | (None, 32, 32, 64)
leaky_re_lu_2 | input: | (None, 32, 32, 64)
LeakyReLU | output: | (None, 32, 32, 64)
conv2d_transpose_2 | input: | (None, 32, 32, 64)
Conv2DTranspose | output: | (None, 64, 64, 1)

Fig 27: Architettura del generatore della GAN per MNIST (a sinistra) e per il dataset anime faces (a destra)

4.4.1 Qualità delle facce generate

Fig 28: Immagini generate dalla GAN dopo 900 epochs di training con 45789 immagini nel dataset

In Fig 28 possiamo vedere alcune immagini generate dalla GAN con 45789 immagini nel training set, alcune facce sono di buona qualità con dettagli del viso, occhi simmetrici naso e bocca. Ad alcune facce manca il naso o la bocca e per alcune gli occhi sono di grandezza asimmetrica.

In media la GAN riesce a generare una forma del viso e capelli realistici ma spesso ha difficoltà con gli occhi, sia per la simmetria che per dimensioni.

In Fig 29 si vede come con un dataset più piccolo la qualità delle facce generate diminuisca notevolmente.

Fig 29: Immagini generate dalla GAN dopo 1000 epochs di training con 1200 immagini nel dataset

Confrontando le immagini generate con delle immagini presenteinel training set originale (Fig 30) vediamo come lo stile di disegno sia stato catturato dalla GAN:

Fig 30: Immagini presenti nel training set

4.4.1.1 FID in funzione del numero di immagini nel dataset

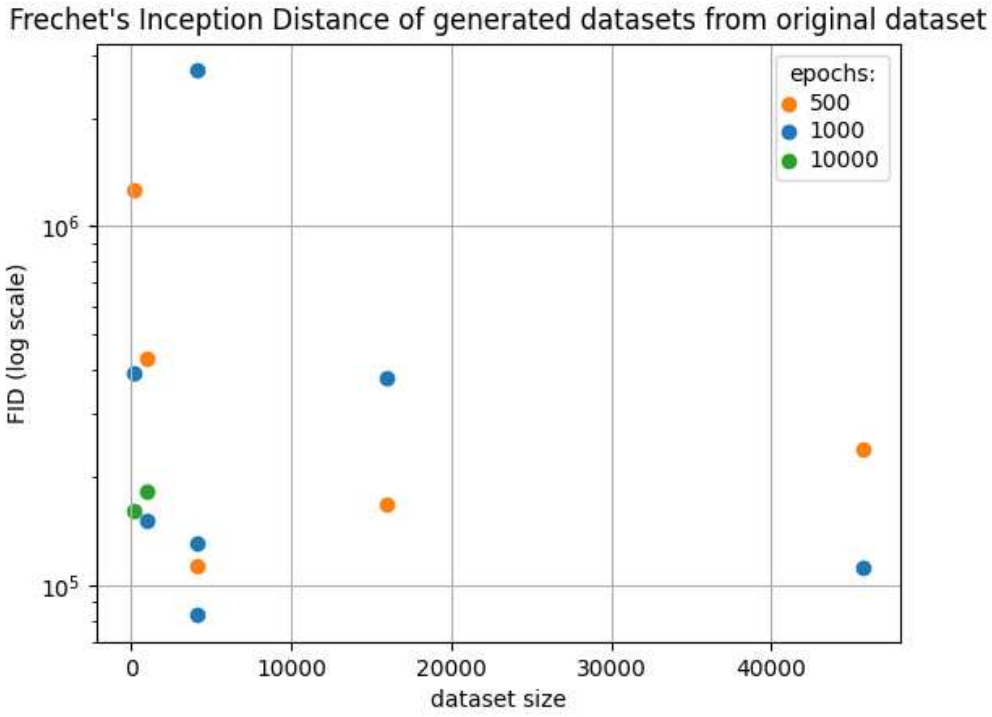


Fig 31-a: FID tra le immagini generate dalle GAN e le immagini del test set in funzione della grandezza del training set

Dalla Fig 31-a si può osservare come per 500 epoche di training la FID diminuisca con l'aumentare della grandezza del dataset, questo risultato è lo stesso trovato per il MNIST.

Per 1000 epoche invece si può notare come per una dataset size di 4096 la FID sia più bassa di quella con 16000 immagini, nonostante le immagini generate non siano di buona qualità (Fig 31-b). Questo mostra come la FID pur essendo uno strumento utile nel calcolo della qualità delle immagini generate non sia sempre indicativa della qualità percepita da un umano.

Fig 31-b: Immagini generate con 1000 epoche di training e 4096 immagini nel training set

4.4.2 Attacco whitebox

Per il dataset anime faces presentiamo alcuni attacchi white box per confermare i risultati trovati con il dataset MNIST.

4.4.2.1 GAN 128, 1200 Per le GAN target con 128, 1200 immagini nel training set il M.I.A. sulla GAN ottiene dei risultati molto simili a quelli per l'attacco sul MNIST. Ovvero con l'aumentare delle epoche di training migliora la qualità (Fig 32-b) delle immagini generate ma la GAN diventa più vulnerabile a M.I.A.

Questo si può notare dai grafici in Fig 32-a che mostrano come per la GAN 128 le likelihood diventino sempre più separate con l'aumentare del numero delle epoche di training

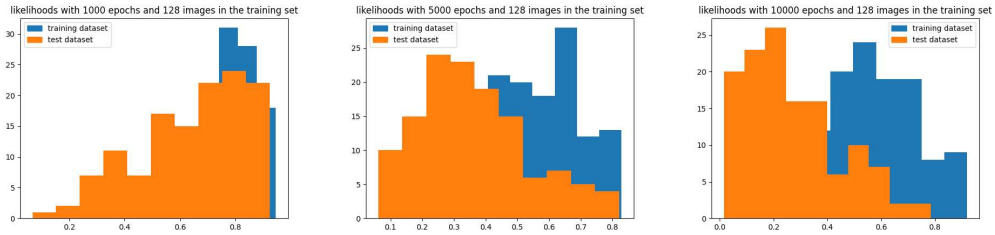
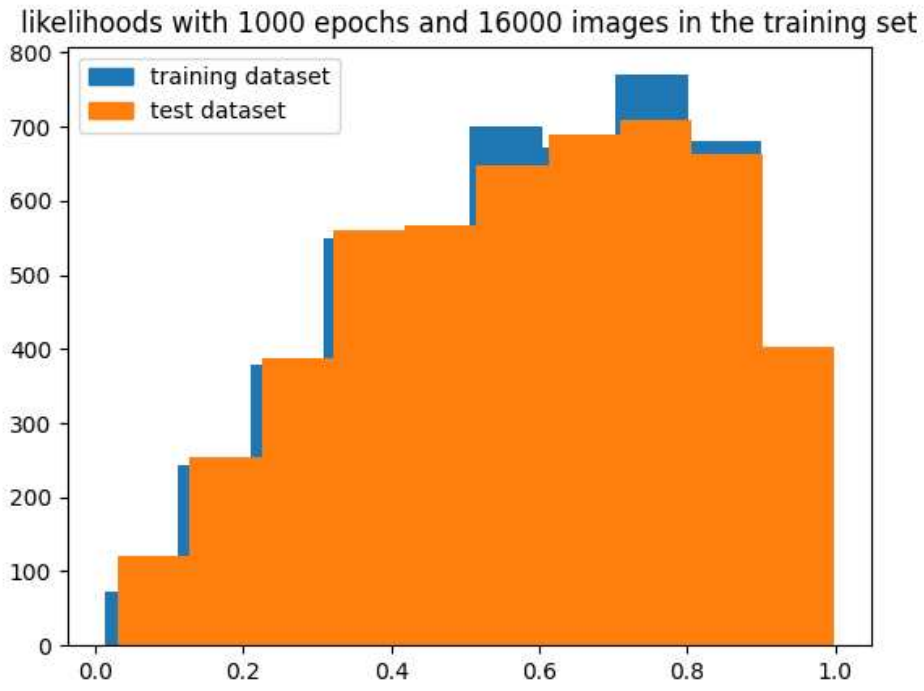


Fig 32-a: Le likelihood si separano con l'aumentare delle epoche di training

Fig 32-b: La qualità delle immagini generate migliora aumentando le epoche di training anche con un dataset di solo 128 immagini

4.4.2.2 GAN 4096, 16000 e 45789 Per le GAN target con 4096, 16000 e 45789 immagini nel training set il M.I.A. sulla GAN a differenza degli attacchi sul MNIST ottiene una precisione molto bassa uguale o peggiore al random guessing. Ovvero la GAN riesce a generare immagini di qualità sufficiente senza presentare overfitting sul training set. Le likelihood in Fig 33 sono completamente sovrapposte e mostrano come per 1000 epoche con 16000 immagini nel dataset l'attacco white box fallisca. Infatti guardando il grafico della precisione sulle top k in Fig 33 si vede come per 1000 epoche la precisione sia $\approx 50\%$ per ogni k.


likelihoods with 1000 epochs and 16000 images in the training set
training dataset
test dataset
800
700
600
500
400
300
200
100
0
0.0
0.2
0.4
0.6
0.8
1.0

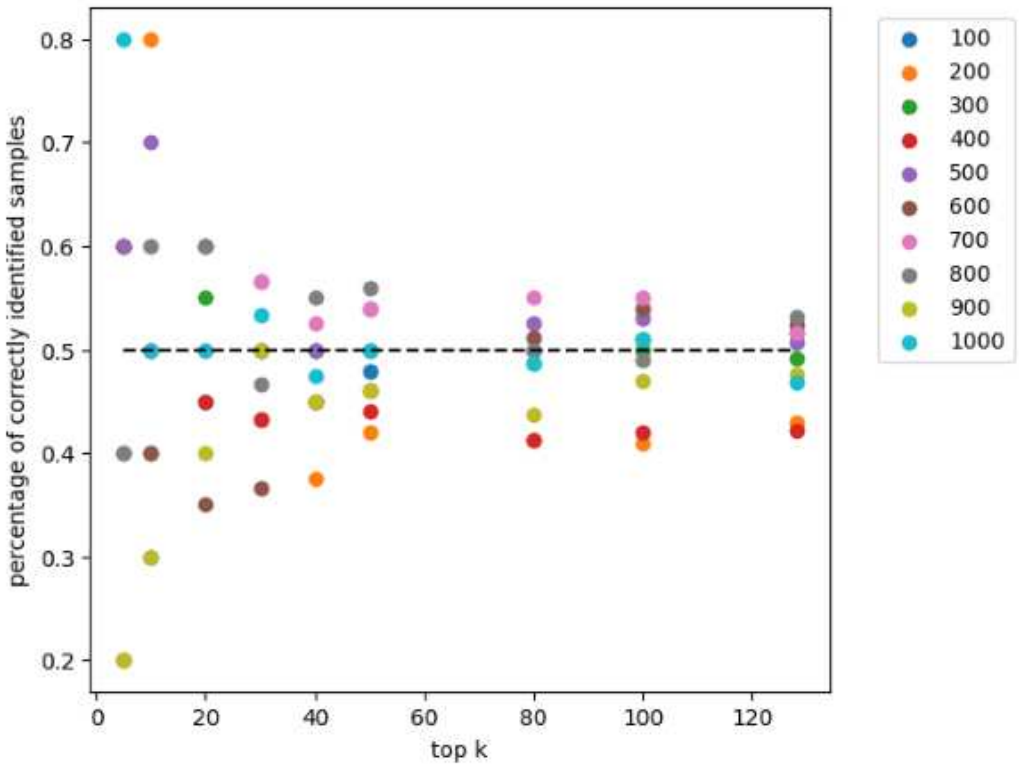


Fig 33: Separazione delle likelihood e precisione nelle top k per la GAN 16000 con 1000 epoche di training

Per la GAN con 45789 immagini si può fermare il training anche solo dopo 500 epoche, infatti la GAN riesce a produrre immagini di buona qualità resistendo agli attacchi white box, come è possibile vedere in Fig 34 e Fig 35.

Fig 34: Immagini generate dalla GAN 45789 con 500 epoche di training

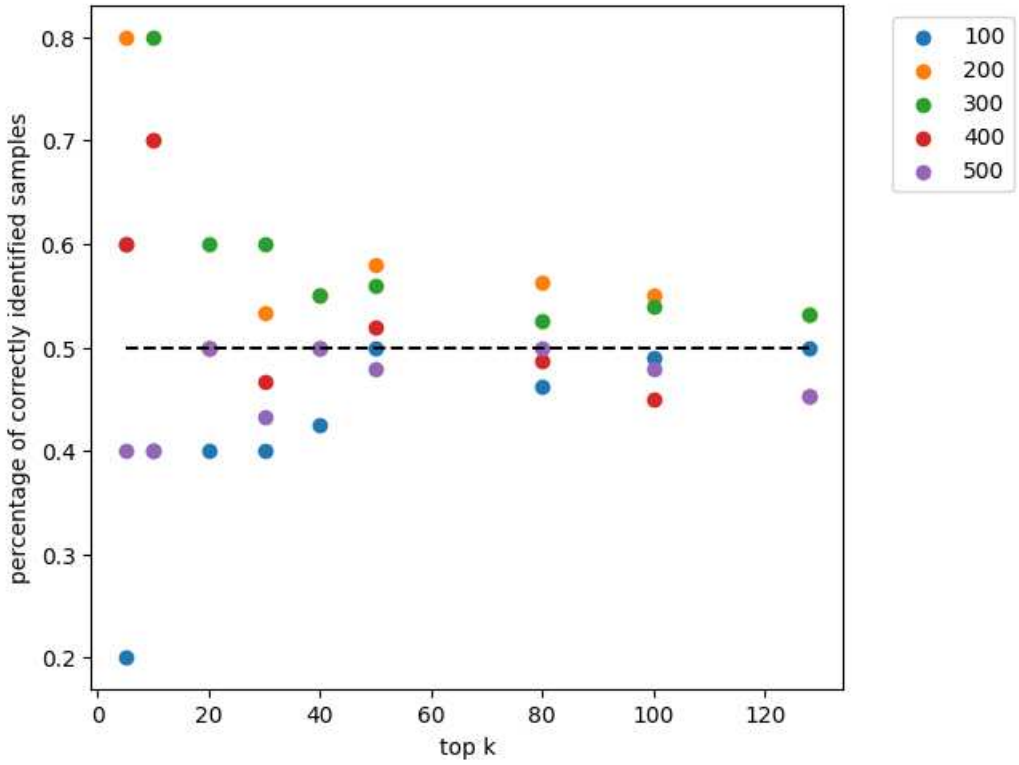


Fig 35: Precisione nelle top k per la GAN 45789 con 500 epoche di training

Quindi per il dataset anime faces [7] si riesce ad addestrare una GAN che riesce produrre immagini di buona qualità mantenendo una completa protezione da M.I.A.

5 Conclusioni e sviluppi futuri

Siamo riusciti a effettuare con successo attacchi black box e white box sulle GAN target addestrate sfruttando i metodi presentati in [6], ovvero il discriminatore della GAN target per gli attacchi white box e la shadow GAN per gli attacchi black box.

I principali risultati trovati sono relativi all'impatto che hanno i parametri di training sulla precisione degli attacchi, riassumiamo i risultati più importanti:

- Il numero di immagini presenti nel dataset di training della GAN target è la variabile che maggiormente influenza la precisione sia per gli attacchi white che black box. Infatti, aumentare la grandezza del training set permette di diminuire l'overfitting e abbassare la precisione degli M.I.A.
- Il numero di epoche di training della GAN target influenza la precisione degli attacchi. A parità di qualità delle immagini è quindi preferibile un numero di epoche minore per addestrare una GAN resistente ai M.I.A.
- La FID può essere usata come metrica di qualità delle immagini generate. Inoltre, a parità di grandezza del training set, avere una FID più bassa (qualità migliore delle immagini generate) rende più vulnerabile la GAN a M.I.A.
- Negli attacchi black box l'attaccante ha a disposizione due risorse per migliorare la precisione dell'attacco: il numero di immagini generate e il numero di epoche di training della shadow GAN. Aumentare il numero di immagini disponibili ha sempre migliorato la precisione dell'attacco. Invece aumentare il numero di epoche della shadow GAN satura ad una precisione massima che non può essere superata, se non aumentando il numero di immagini disponibili.

Da questi risultati si può sostenere che in futuro quando verrà addestrata una GAN su dataset sensibili (ad esempio nel campo medico o legale) bisognerà tenere conto della vulnerabilità della GAN ad attacchi di inferenza.

Per abbassare la precisione di questi attacchi potrebbe essere necessario diminuire il numero di epoche di training della GAN, peggiorando però la qualita delle immagini generate. In alternativa si potrebbe aumentare il numero di immagini presenti nel training set, tuttavia questo spesso non è

possibile in quanto non è disponibile una quantità di dati sufficiente ad evitare l'overfitting (soprattutto nel campo medico).

Un altro possibile meccanismo di difesa dai M.I.A. consiste nel limitare il numero di richieste che è possibile effettuare alla GAN. Questo va a ridurre la grandezza del dataset di training che un attaccante può utilizzare per addestrare una shadow GAN.

In ogni caso chi addestra una GAN dovrebbe considerare la possibilità di costruire una GAN shadow per determinare se la precisione di un attacco black box sia troppo alta per poter rendere pubblica la propria GAN.

Alcuni possibili sviluppi futuri da espandere sono:

- Studiare se la Fréchet's Inception Distance tra le immagini generate dalla GAN target e quelle generate dalla GAN shadow sia collegata alla precisione dell'attacco black box.
- Studiare come la grandezza delle immagini influenzi la precisione degli attacchi.
- Provare M.I.A. su dataset relativi a immagini mediche dove la scarsità di immagini per training delle GAN è un problema comune.
- Analizzare se con lo stesso dataset alcune architetture diverse delle GAN siano più resistenti a M.I.A.
- Analizzare se sia possibile addestrare una GAN addestrata sul MNIST completamente resistente ad attacchi white box, ovvero che generi immagini di qualità sufficiente ma gli attacchi white box ottengano una precisone $\approx 50\%$.

Bibliografia

[1] M. N. et al, "Comprehensive privacy analysis of deep learning," pp. 739–740, 2019.

[2] H. S. et al, "Adversarial attacks against deep generative models on data: A survey," 2023.

[3] S. M. Saverio Cavasin Daniele Mari, "Fingerprint membership and identity inference against generative adversarial networks," *Elsevier*, vol. submitted to Pattern Recognition Letter, pp. 4–6, 2023.

[4] R. Shokri, "Membership inference attacks against machine learning models," pp. 1–6, 2017.

[5] I. J. Goodfellow, "Generative adversarial networks," pp. 1–9, 2014.

[6] J. Hayes, "LOGAN: Membership inference attacks against generative models," pp. 1–6, 2017.

[7] S. Churchill, "Anime face dataset," 2019.