# Exploring Model Inversion Attacks in the Black-box Setting

Antreas Dionysiou
University of Cyprus
Nicosia, Cyprus
dionysiou.antreas@ucy.ac.cy

Vassilis Vassiliades
CYENS Centre of Excellence
Nicosia, Cyprus
v.vassiliades@cyens.org.cy

Elias Athanasopoulos
University of Cyprus
Nicosia, Cyprus
athanasopoulos.elias@ucy.ac.cy

## ABSTRACT

Model Inversion (MI) attacks, that aim to recover semantically meaningful reconstructions for each target class, have been extensively studied and demonstrated to be successful in the white-box setting. On the other hand, black-box MI attacks demonstrate low performance in terms of both effectiveness, i.e., reconstructing samples which are identifiable as their ground-truth, and efficiency, i.e., time or queries required for completing the attack process. Whether or not effective and efficient black-box MI attacks can be conducted on complex targets, such as Convolutional Neural Networks (CNNs), currently remains unclear.

In this paper, we present a feasibility study in regards to the effectiveness and efficiency of MI attacks in the black-box setting. In this context, we introduce Deep-BMI (Deep Black-box Model Inversion), a framework that supports various black-box optimizers for conducting MI attacks on deep CNNs used for image recognition. Deep-BMI's most efficient optimizer is based on an adaptive hill climbing algorithm, whereas its most effective optimizer is based on an evolutionary algorithm capable of performing an all-class attack and returning a diversity of images in a single run.

For assessing the severity of this threat, we utilize all three evaluation approaches found in the literature. In particular, we (a) conduct a user study with human participants, (b) demonstrate our actual reconstructions along with their ground-truth, and (c) use relevant quantitative metrics. Surprisingly, our results suggest that black-box MI attacks, and for complex models, are comparable, in some cases, to those reported so far in the white-box setting.

## KEYWORDS

Model inversion, inference attack, security, privacy

## 1 INTRODUCTION

Despite being so popular, Machine Learning (ML) models have been proven vulnerable to various security and privacy attacks, such as model extraction [55], membership inference [54], and adversarial sample generation [16]. In this paper, we focus on *Model Inversion (MI)* attacks. In this setting, an adversary aims to generate inputs resembling the original ones used for training the target model [19]. Such information leaks may enable the de-anonymization of users [20] and expose personal or sensitive information [5].

The effectiveness and efficiency of the proposed MI attacks vary significantly depending on a number of factors, such as the available to the adversary information about the target model as well as its complexity level. For example, a number of white-box MI attacks have been successfully conducted on a wide-range of targets [6, 19, 21, 55, 58, 62]. Furthermore, some MI attacks target low-complexity models (see Sec. 2 for a categorization of the target models' complexity) only trying to infer a small number of sensitive features which are drawn from a tractably small domain [20, 27, 58].

**Table 1: Qualitative positioning of the state-of-the-art MI attacks found in the literature. Papers shown more than once propose MI attacks that meet each cell's specifications. In this work, we explore the possibility of conducing MI attacks, which are fully agnostic regarding the target model's internals, on mid-complexity image recognition models.**

| | | Practicality | | |
|---|---|---|---|---|
| | | White-box | Black-box | |
| | | | Partly agnostic | Fully agnostic |
| Target's complexity | Mid | [19], [26], [29], [60], [62] | [26], [48] | Deep-BMI, [4], [19], [60] |
| | Low | [27], [48], [58] | [20], [58] | [39] |

Table 1 shows a *qualitative* positioning of the state-of-the-art MI attacks found in the literature. Each paper is placed in the respective cluster according to its practicality and the target's complexity. As shown, only a handful of black-box MI attacks, which are fully agnostic about the target model's internals, have been conducted on mid-complexity target models trained on image datasets [4, 19, 60]. Nonetheless, these papers impose significant limitations, in *training class inference* attack setting, in terms of both effectiveness, i.e., reconstructing samples which are identifiable as their ground-truth, and efficiency, i.e., computational overhead and queries made to the target model.

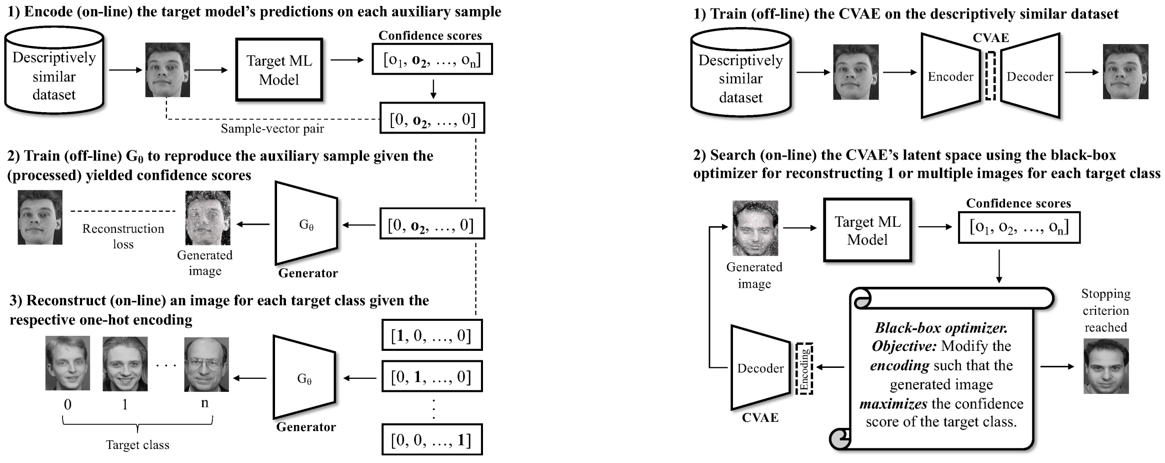
To be more specific, Fredrikson et al. [19] and Aïvodji et al. [4] *fail* on conducting MI in the black-box setting since *the inferred images are semantically meaningless and do not even form recognizable faces* [60], while also being dramatically inefficient, e.g., Fredrikson et al. [19] require 50-80 days to complete the attack process. Contrary, Yang et al. [60] manage to reconstruct identifiable faces requiring, however, 244,306 queries to be made to the target model and ≈ 12 hours for their attack to be completed [1].

In this paper, we present a *feasibility study* in regards to the effectiveness and efficiency of MI attacks in the black-box setting. We consider this work a feasibility study for two reasons. First, MI has a probabilistic nature in contrast to other attacks in the field which have clear baselines, e.g., the baseline for membership inference attacks is 50% [54]. In other words, there is *not* a scientific consensus in regards to when an MI attack is considered (in)effective. Thus, for concluding about our MI attacks' performance, we experiment with the same target models as the ones utilized in relevant

[1] Note that the authors utilize a workstation equipped with 2 Intel Xeon CPUs 16 cores/32 threads in total, 256GB RAM, and 2 NVIDIA Tesla P100 GPU cards.

*Proceedings on Privacy Enhancing Technologies 2023(1), 190–206*

https://doi.org/10.56553/popets-2023-0012


1) Encode (on-line) the target model's predictions on each auxiliary sample
Descriptively similar dataset
Target ML Model
Confidence scores
[o1, o2, ..., on]
[0, o2, ..., 0]
Sample-vector pair
2) Train (off-line) G0 to reproduce the auxiliary sample given the (processed) yielded confidence scores
Reconstruction loss
Generated image
G0
Generator
[0, o2, ..., 0]
3) Reconstruct (on-line) an image for each target class given the respective one-hot encoding
[1, 0, ..., 0]
[0, 1, ..., 0]
[0, 0, ..., 1]
G0
Generator
0
1
n
Target class
1) Train (off-line) the CVAE on the descriptively similar dataset
Descriptively similar dataset
CVAE
Encoder
Decoder
2) Search (on-line) the CVAE's latent space using the black-box optimizer for reconstructing 1 or multiple images for each target class
Target ML Model
Confidence scores
[o1, o2, ..., on]
Generated image
Black-box optimizer. Objective: Modify the encoding such that the generated image maximizes the confidence score of the target class.
Stopping criterion reached
Decoder
Encoding
CVAE

(a) Yang et al. [60]

(b) Deep-BMI

**Figure 1: Conceptual comparison between Yang et al. and Deep-BMI black-box MI frameworks. As shown, for the first two steps, Yang et al. require $k$ queries to be made to the target model ($k$: size of the descriptively similar dataset), one for each auxiliary sample, for training the inversion model. Contrary, Deep-BMI trains the CVAE (inversion model) on the descriptively similar dataset off-line, without querying the target model. Furthermore, in Yang et al.'s MI approach, the inversion model learns to reconstruct *different* auxiliary samples that yield the *same* target class with a certain confidence score, i.e., the model's weights are adapted so as to generate an *average sample* that represents each target class. Instead, Deep-BMI trains off-line the inversion model by encoding and then decoding (reconstructing) each auxiliary sample such that it minimizes the respective reconstruction loss. For performing the actual inversion, Yang et al. feed the respective one-hot encoding for each target class to the inversion model and receive, as a response, the reconstructed image –one query per target class is required. Contrary, Deep-BMI employs black-box optimization techniques for performing structured perturbations to the 20 dimensional encoding given as input to the decoder part of the inversion model, such that the generated image maximizes the confidence score of the target class. Finally, Yang et al. can only return one solution/reconstruction per target class, whereas Deep-BMI is capable of returning multiple diverse solutions/reconstructions depending on the deployed black-box optimizer.**

works [4, 19, 60], in order to have a direct comparison. Second, the black-box MI attacks introduced so far, that operate in training class inference setting, are considered hard and in certain cases impossible. Thus, our work aims at exploring the extent to which black-box MI attacks are possible (feasible) on mid-complexity targets, with minimal overhead. In this context, we propose Deep-BMI (Deep Black-box Model Inversion), a modular MI framework that supports various black-box continuous optimization algorithms for *maximizing* the target model's confidence scores.

So, how do we tackle the MI problem in the black-box setting? One approach would be to start from random pixels and continually perturb them until the confidence score of the target class is maximized. This has the advantage of being simple and generative. However, this approach is known to generate *fooling images* [42]. That is, images that yield the target class with high confidence score but are visually irrelevant to the ground-truth. This is because discriminative ML models allocate large areas of a high-dimensional space to a class, and these areas contain regions with points (images) that are far away from the natural ones. However, even if fooling images were not an issue, it would need a huge amount of iterations and queries to the target model for generating a single image, let alone a complete dataset, i.e., the target model's training set.

Another approach, firstly proposed by Yang et al. [60], is to draw a more generic dataset based on the adversary's background knowledge for training a generative inversion model. In particular, one could use images from a public dataset that is *descriptively similar* to the target model's training set. The descriptively similar dataset may share basic descriptive characteristics and (by assumption) similar distribution with the target model's training set, but does *not* contain the same training instances. For example, one can utilize faces datasets, such as AT&T [50], CelebA [35], and FERET [44], for targeting some popular facial recognition ML models, such as Google Vision [3], Amazon Rekognition [1] and Clarifai [2]. One can probe the target model to obtain basic descriptive information, such as the task's context, images' resolution and type (color or grayscale), for finding a descriptively similar dataset to download and use. The advantage of this approach lies in the assumption that the images contained in both sets share features, which would make some of the images of the public set likely to maximize the confidence score of a certain class of the target model.

Nonetheless, Yang et al.'s approach for performing MI cannot exploit the full potential of the descriptively similar dataset in *training class inference* attack setting. To be more specific, it suffers from the following limitations: (a) it causes the inversion model to reconstruct images that are essentially the average of all the auxiliary samples that yield each target class with a certain confidence score, (b) the number of queries made to the target model is directly related to the size of the descriptively similar dataset, and (c) it lacks of an optimization mechanism for searching the inversion model's latent space and generating more representative, for each target class, reconstructions. In Sec. 2, we provide more details in regards to these limitations and how Deep-BMI copes with them.

In this paper, our key insight is that it is possible to have the best of both approaches by training off-line, without involving/querying the target model, an inversion model using the descriptively similar dataset and searching its latent space using various black-box optimizers. More specifically, our approach, shown in Fig. 1(b), is split into two phases. At a first phase, we train a generative model, in our case a deep Convolutional Variational Auto-Encoder (CVAE), on a descriptively similar public dataset, which allows us to learn the structure of the input and, by assumption, features that are shared between the two sets. By doing so, the CVAE enables at a second phase to sample its latent space and perform not pixel perturbations, but *structured perturbations* which respect the learned features of the public dataset (and by assumption of the target model's training dataset), thus, mitigating the issue of generating fooling images. Sampling is computationally cheap and is done until the confidence score of a target class is maximized. We motivate the selection of CVAEs over alternative generative models in Sec. 3.

For assessing the severity of this threat, we deploy Deep-BMI on a target Convolutional Neural Network (CNN) trained to classify digits (scenario 1) and recognize faces (scenario 2). In doing so, we are able to observe the fluctuations of our MI framework's performance as the complexity of the problem increases. Scenario 1 is based on digit classification and it is used primarily to showcase the mechanics of Deep-BMI applied on a target of modest difficulty, but with low actual privacy implications. Scenario 2 is based on face recognition, a target of higher complexity with profound privacy implications. We focus on targeting CNNs since the literature showed that they demonstrate state-of-the-art robustness against MI attacks compared to other ML models [4, 19, 60]. Note that we consider the exploration of Deep-BMI's effectiveness on other target models as future work.

For evaluating Deep-BMI's performance we follow a three-fold approach (see Sec. 4) composed by (a) showing actual reconstructions along with their ground-truth, (b) using relevant quantitative metrics, and (c) conducting a user study with human participants. We show that Deep-BMI performs well in terms of all three evaluation approaches. For example, for the user study, our results suggest that Deep-BMI is effective managing to infer identifiable digits with 95.45% success rate on average across all target classes, with 83.58% of the respondents feeling "very confident" or "confident" when giving their answers. Moreover, Deep-BMI manages to infer identifiable faces with 60.05% success rate on average across all target classes, with 52.91% of the respondents feeling "very confident" or "confident" when giving their answers. Furthermore, Deep-BMI is efficient requiring 234,306 fewer queries to the target model, for reconstructing identifiable images compared to Yang et al.'s MI framework [60]. Note that apart from Yang et al., no other black-box MI framework has been successfully conducted on discriminative ML models, in the image recognition field, at least to our knowledge.

Last, but not least, Deep-BMI reconstructs images that yield the target class with similar, and in many cases identical, confidence scores to those observed when feeding as input the ground-truth images. This is important since it can lead to further security implications, such as evading authentication systems based on face recognition that leverage the confidence scores [1–3].

**Our contributions** can be summarized as follows.

(1) We conduct a feasibility study for the effectiveness and efficiency of black-box MI attacks, when facing mid-complexity targets. For doing so, we introduce a black-box MI framework (Deep-BMI), and deploy it on two *well-generalizable*[2] target models used for image recognition. Deep-BMI is *modular*, meaning that potential adversaries can plug their own black-box optimizers for attacking specific target models.
(2) Although not being the first to utilize a descriptively similar dataset to conduct MI attacks [60], we are the *first* to reconstruct identifiable images (77.75% success rate on average) with increased efficiency (234,306 fewer queries compared to Yang et al.) in *training class inference* attack setting. For example, Deep-BMI reconstructs 100% identifiable digits and 90.47% identifiable faces in just 1.59 and 21.24 seconds, respectively, for specific target classes, i.e., for specific digits and faces. In addition, we demonstrate that black-box MI attacks achieve comparable performance to Fredrikson et al.'s [19] white-box MI attacks, on a target of similar complexity.
(3) Deep-BMI supports several black-box optimizers, or adversary types, that trade attack time for performance. The most effective optimizer is based on Centroidal Voronoi Tesselation (CVT) Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [56], achieving 100% & 77.77% success rates for scenarios 1 & 2, respectively. In terms of efficiency, there is not a single metric that can rule out the best optimizer. For instance, if one wants to optimize for *fewer queries* they should use CVT-MAP-Elites since it is the most sample-efficient [3] optimizer requiring 20 & 3 queries per reconstruction/solution for scenarios 1 & 2, respectively. Contrary, if one wants to optimize for a *faster* attack, that takes less computational time, they should go with adaptive hill climbing algorithm as it requires 1.59 & 21.24 seconds for attacking a specific target class for scenarios 1 & 2, respectively.
(4) We experimentally demonstrate that optimizers returning multiple *diverse* solutions for each target class, such as CVT-MAP-Elites, are the most effective ones on conducting MI attacks on deep image recognition models. This is because the diversity of the returned solutions leaks *qualitative* information from *different perspectives* for each target class.
(5) To foster further research on this topic and ease reproducibility, we release the code for our experiments [4].

## 2 MODEL INVERSION

**The MI Problem.** MI attacks aim at resembling a part or whole of an input sample included in the target model's training set. Take for example a target ML model, $M_t$, which is a function $f$ that takes as input a feature vector $x_1, ..., x_n$ with $n$ dimensions and outputs a prediction $y = f(x_1, ..., x_n)$ [19]. The first MI attack proposed by Fredrikson et al. [20] uses black-box access to $f$ for inferring a sensitive feature, $x_1$, given some knowledge about the other non-sensitive features, $x_2, ..., x_n$, the output of the target

[2] Such models classify previously unseen input samples with high success rates.

[3] Sample-efficiency is inversely proportional to the number of queries required for generating a single sample. Thus, high sample-efficiency means less queries required and low sample-efficiency means more queries required.

[4] https://bitbucket.org/srecgrp/deep-bmi-public/

model, $y$, and any other auxiliary knowledge regarding $M_t$ or the marginal priors for the individual features in the feature vectors. Then, their algorithm picks the value for $x_1$ which maximizes the confidence score for a specific target class. However, as Fredrikson et al. [19] explain in their follow-up paper, their previously proposed MI attack suffers from significant limitations, one of them being that it cannot be applied when the unknown features cover an intractably large set or they are drawn form a large domain.

In our case, similar to Yang et al. [60], we explore the possibility of conducting black-box MI attacks on image recognition models for inferring all the features (pixels' values) of an image, by only utilizing a dataset which is descriptively similar to the one used for training the target model.

MI comes in two flavours/variations:

- *Data reconstruction:* The adversary aims to reconstruct the input sample given the confidence score vector on it. For example, in a facial recognition ML model, the attacker's goal is to reconstruct the facial image of a person given its yielded confidence score vector [60]. The same attack setting applies to other ML-based biometric authentication systems, where the adversary's goal is to recover the biological data of an individual given the system's yielded confidence score vector on them [30].
- *Training class inference:* The adversary aims at recovering a semantically meaningful reconstruction for each target class given a trained ML model. Using the same facial recognition example, the attacker's goal is to recover a recognizable facial image of an arbitrary person (class) in the training dataset [60]. In a similar setting, considering other ML-based biometric authentication systems, an adversary aims to infer the biological data of an arbitrary individual (class) in the training dataset [30].

We focus solely on *training class inference* attack setting for which Yang et al. [60] reconstruct samples with low recognizability (see Fig. 8). This is because having the yielded confidence score vector of the target, to be inferred, sample is a *rather strong* assumption.

**Threat Model.** In this paper, we focus on the *black-box* attack setting where an adversary *cannot* access, by any means, the target model's internals, such as its architecture, parameters, or training data, being only capable of querying the target model and receiving the confidence score for each class as a response. The difficulty and practicality of this attack setting is dramatically *higher* compared to *white-box* MI attacks, such as those proposed by Fredrikson et al. [19]. This is because white-box MI attacks assume adversaries that can access the target model's internals and use gradient descent to minimize a cost function (prediction error), for reconstructing images which are identifiable as their ground-truth. In contrast, Deep-BMI reconstructs identifiable digits and faces by only exploiting the confidence score of the winning class. We consider the exploration of fully black-box MI attacks which can only access the discrete label of the winning class, without its respective confidence score, as future work (see Sec. 5).

An important goal for black-box adversaries is to minimize the number of queries made to the target model when delivering their MI attacks. This is because black-box MI attacks with excessive query demands: (a) are inefficient, and therefore impractical, and (b) may raise the suspicion of the target system, in which case it will classify them as malicious. Besides, black-box adversaries with infinite queries can approximate the performance of white-box adversaries since they can estimate the target model's gradients or "steal" the whole target model in a black-box fashion. Thus, the number of queries made to the target model is an important factor limiting black-box adversaries' ability to conduct MI attacks compared to white-box ones.

**Target Model's Complexity.** There is no single widely accepted categorization of the target model's complexity. Below, we introduce such a categorization only for facilitating the presentation of the experiments and accurately positioning Deep-BMI against the state-of-the-art. In particular, we categorize target models into three groups based on their architecture and no. of parameters: (a) low-complexity: shallow models, e.g., support vector machines [15] or Artificial Neural Networks (ANNs) with up to 500K parameters, (b) mid-complexity: CNNs with 500K to 500M parameters, and (c) high-complexity: CNNs with more than 500M parameters.

**Yang et al.'s** black-box MI framework, see Fig. 1(a), works as follows. First, it encodes, on-line, the target model's largest predicted class with confidence on each auxiliary sample, to an $n$-dimensional vector ($n$: no. of output classes), by filling rest classes with zeros. Second, it uses the encoded predictions as features to train, off-line, the inversion model $G_\theta$. In other words, $G_\theta$ is trained to reconstruct each auxiliary sample given the yielded encoded vector on it. Third, after $G_\theta$ is trained, it creates an one-hot encoding for each target class and feeds it to $G_\theta$; its output is the inferred image of the respective class. Put simply, $G_\theta$, after considering all image-vector pairs, is asked to generate an image that will most probably yield the target class with the highest confidence score, i.e., 1.

This MI strategy imposes the following limitations. First, it causes the inversion model to reconstruct images that are essentially the *average* of all the auxiliary samples that yield each target class with a certain confidence score. This is because each target class will be most probably yielded by more than one auxiliary samples. As a result, $G_\theta$ will learn to generate an image that minimizes the reconstruction loss for all different auxiliary samples that yield each target class with a certain confidence score. Second, it requires one query to the target model for each auxiliary sample. Thus, the total number of queries equals to the size of the descriptively similar dataset. Third, it lacks of an optimization mechanism for searching the inversion model's latent space and generating *more representative*, for each target class, reconstructions. That is, reconstructions that approximate the most the ground-truth. In its current form, Yang et al.'s attack effectiveness is bounded to the similarity between the auxiliary samples and the samples included in the target model's dataset. In other words, the higher the similarity between the distributions of the descriptively similar dataset and the target model's dataset, the higher the MI performance.

**Deep-BMI** tackles Yang et al.'s limitations in the following order. First, in Deep-BMI's context, the inversion model does not learn to reconstruct *different* auxiliary samples that yield the *same* target class with a certain confidence score. Contrary, our approach adopts the classic training paradigm for Auto-Encoders and trains, off-line, a CVAE by encoding and then decoding/reconstructing each auxiliary sample such that it minimizes the respective reconstruction loss. Second, Deep-BMI decouples the number of queries made to the target model with the size of the descriptively similar dataset;

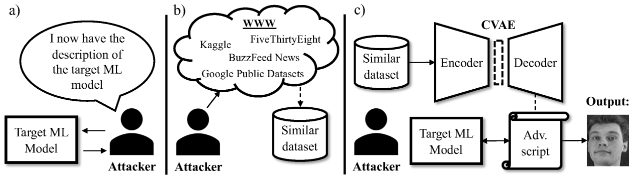
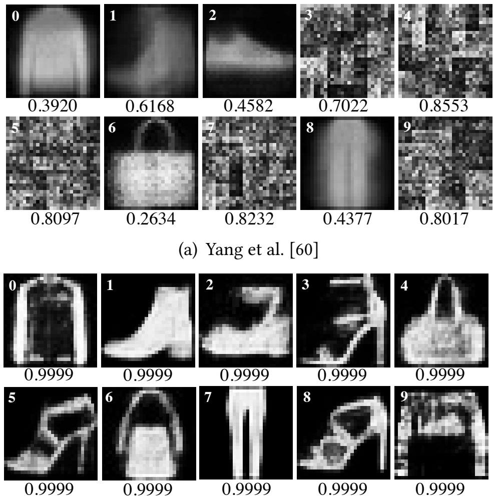


**Figure 2: Deep-BMI attack pipeline. The attacker: (a) probes the target model to get a basic description of its functionality, (b) searches the web and downloads a publicly available dataset, which is descriptively similar to the target model's training set, and (c) trains a CVAE on the downloaded dataset and selects an adversarial script, black-box optimizer, according to their priority, time vs. performance, for performing the MI. During the whole attack process the adversarial script has a duplex communication with the target model.**



(a) Yang et al. [60]

(b) Deep-BMI

**Figure 3: Reconstructed images for digits 0-9, along with the target model's confidence scores, for Yang et al. and Deep-BMI, when having access to a background dataset (Fashion-MNIST [59]) which has different distribution to the target dataset (MNIST [34]). As shown, although the two MI frameworks manage to maximize, to some extent, the confidence score of each target class, fooling images, which are visually irrelevant to the ground-truth, are resulted. Note that in this work, we do *not* try to address the problem of fooling images when the background dataset has a different distribution.**

in Deep-BMI's context, the number of queries made to the target model is *explicitly* controlled by the utilized black-box optimizer. Third, Deep-BMI is by-design a modular MI framework that supports several black-box optimization methods, such as hill-climbing and Covariance Matrix Adaptation Evolution Strategies (CMA-ES), for exploring the CVAE's latent space and inferring more representative, for each target class, reconstructions. See Appx. A for more details on black-box continuous optimization. Fig. 1 shows a *conceptual* comparison between Deep-BMI and Yang et al.

## 3 DEEP-BMI

**Overview.** At a high level our MI framework is composed of a deep CVAE, shown in Fig. 15 (Appx. B), which acts as the attacker model and is trained on a descriptively similar, yet different, to the target model's training set. As shown in Deep-BMI's attack pipeline (Fig. 2), an attacker probes the target model to obtain basic descriptive information (phase a), for finding a descriptively similar dataset to download (phase b) and use (phase c). In our case, we select the similar dataset by performing a *visual inspection*, same as Yang et al. Note that an in-depth analysis on *how* to find such a descriptively similar dataset is *orthogonal* with the work of this paper. For more details on this aspect see Sec. 4.1 in Yang et al.'s work [60].

Yang et al. were the first to utilize a descriptively similar dataset for performing MI attacks in the black-box setting [5]. The authors showed that the distribution similarity between the target and the descriptively similar dataset largely affects the overall MI performance. This is because black-box MI frameworks can only utilize the target model's confidence scores, which they aim to maximize for each target class, for performing the inversion, thus increasing the risk of reconstructing *fooling images* –see Fig. 3 for an experimental example. Consequently, for black-box MI frameworks, the choice of the similar dataset to be used for training the attacker model is of utmost importance as it will constrain the search towards the recognizable regions of respective areas in image space.

Similar to Yang et al., we assume that the input/output shapes of the target model are known to *any* user, and thus they are known to the attacker as well. While a detailed analysis in regards to the exact descriptive properties/attributes that a descriptively similar dataset must have is an interesting research angle, we consider it as *out of scope* for this paper (see Sec. 5).

**Using Generative Models.** Our MI attacks are based on the fact that CVAEs are able to generate similar input samples to the ones included in the target model's training set, when fed with random noise vectors, matching the latent space encoding dimension found in the middle of the CVAE architecture, generated through a Gaussian distribution with a specific mean and variance. Thus, we cast MI in the image domain as a search problem and optimize it by searching the CVAE's latent space using various sampling algorithms (black-box optimizers) from naive to more intelligent ones, until the confidence score of the target class is maximized. Employing naive sampling algorithms will help us understand the extent to which MI attacks are possible on image recognition models.

**Motivating the Selection of CVAE Over Alternatives.** Deep-BMI needs to be able to generate images. The simplest option is to utilize an Auto-Encoder (AE) [47]. However, the latent space of an AE is not guaranteed to be continuous, so we cannot easily interpolate between solutions. Then, we have Variational AEs (VAEs) and Generative Adversarial Networks (GANs). We choose VAEs, and specifically CVAEs, over GANs since vanilla GANs are known to be harder to train than vanilla VAEs and we want to keep a simple model to check the feasibility of our MI framework. Finally, there are even more advanced generative models, however, we stick to CVAEs for the same aforementioned reasons. That is, we want a simple approach that is well-established with available source code.

[5] Apart from Deep-BMI and Yang et al., no other black-box MI framework is based on a descriptively similar dataset.

**Problem Formulation.** Given a pre-trained image classification model $F : X \longrightarrow Y$, which maps grayscale images to a set of $Y$ classes, an attacker aims to craft an image input $X_{adv}$, by any means, and whose ground-truth label is $Y_{tar} \in Y$ so that $F(X_{adv}) = Y_{tar}$ with the highest possible confidence score. Intuitively, the $X_{adv}$ that yields $Y_{tar}$ with the highest confidence score will be visually similar to the mean of the samples included in the target model's training dataset for that specific target class. In fact, the attacker also requires that $Similar(X_{adv}, X_{mean}) \geq \epsilon$ ($X_{mean}$ being the mean of training samples with label $Y_{tar}$) for a domain-specific function $Similar$, where the bound $\epsilon \in \mathbb{R}^+$ captures the notion of visual semantics preserving alteration. We simulate the $Similar$ function by: (a) conducting a user study on the reconstructed images [4, 19], (b) showing actual reconstructions along with their ground-truth [60], and (c) using relevant quantitative metrics [62].

**Scenarios.** We consider two scenarios when deploying our MI framework which differ in terms of complexity. In particular, for the first scenario we conduct MI attacks on a target model trained on MNIST [34] using an attacker model trained on EMNIST-letters [11]. For the second scenario we conduct MI attacks on a target model trained on AT&T database of faces [50] using an attacker model trained on CelebA [35]. Note that, for both scenarios, the classes of the target and the attacker dataset do *not* overlap. For a detailed description of these datasets see Sec. 4.1 & 4.2.

We deliberately choose to conduct MI attacks on these two settings in order to conclude about whether or not, and to what extent, the target model's complexity affects the performance of black-box MI frameworks, such as Deep-BMI. Moreover, our choice of datasets facilitates the direct comparison with other approaches found in the literature, since they use the exact same datasets [4, 19, 60]. We choose *not* to attack commercially available image recognition models [1–3], due to ethical reasons. Instead, we train and test our own target image recognition models to evaluate Deep-BMI's performance free of any ethical considerations.

We focus on targeting well-generalizable models since the literature showed that they are significantly more robust against similar attacks compared to overfitted ones [49, 54]. In particular, the target model for scenario 1 achieves 99.34% training accuracy and *99.13% testing* accuracy. Likewise, the target model for scenario 2 achieves 99.48% training accuracy and *99.44% testing* accuracy. Thus, for both tested scenarios the target models are correctly trained on the respective datasets achieving high performance and generalization. The same holds for the attacker models. The train-test loss per epoch graphs for scenarios 1 & 2 for both the target and attacker models are shown in Fig. 14 (Appx. A).

**Adversaries – Black-box Optimizers.** A summary of our adversaries (black-box optimizers) compared to the adversaries found in the literature is depicted in Table 2. Note that we only compare our approach against MI attacks on ML models and datasets with similar to our target models' complexity. That is, deep CNNs with 500K to 500M parameters. We do this because MI attacks on mid-complexity models trained on image datasets are inherently harder to be performed compared to MI attacks on common/low-complexity ones, such as regression models trained on the IWPC dataset [20]. Later, in this section, we provide a taxonomy of the utilized black-box optimizers in terms of sample-efficiency and number of returned solutions.

In total, we utilize six black-box optimizers which can be roughly categorized into two groups, namely those that are able to attack a *single* target class per invocation (single-class) and those that are able to attack *all* target classes simultaneously (all-class). The first optimizer, namely random sampling, is heavily based on random vectors retrieved from a Gaussian distribution. The rest of the optimizers, namely simple and adaptive hill climbing, CMA-ES, MAP-Elites and CVT-MAP-Elites, are based on stochastic, derivative-free optimization and Quality Diversity (QD) algorithms.

The common aspect of all the adversaries is the searching for the *best*, in terms of maximizing the target model's confidence score, reconstruction for each target class. Thus, optimizers that *only* exploit the target model's confidence scores come in handy. Here, we stress that Deep-BMI is by-design a *modular* MI framework, meaning that potential adversaries can plug their own black-box optimizers for testing the feasibility of MI attacks on any target model.

All the adversaries shown below receive as input the trained target model, $M_t$, and the trained attacker model, $M_a$. In the MLaaS setting, the $M_t$ and $M_a$ arguments can be the keys to access the target and attacker models, respectively, via an API.

**Single-class Black-box Optimizers.** *Random Sampling:* First, this optimizer creates $rand\_vecs\_no$ (parameter given by the user) random vectors drawn from a Normal distribution. Next, it reconstructs the images from those random vectors using $M_a$'s decoder part. After that, the optimizer feeds the target model with the reconstructed images and gets the predictions for each image. Finally, it finds and shows the image that yields the *highest* confidence score for the given target class. The computational complexity of this optimizer is $O(n)$, where $n$ is the total number of random vectors to be decoded. The reported computational complexity of all optimizers is related to the number of queries made to the target model.

*Hill Climbing (HC):* This optimizer utilizes a hill climbing algorithm in order to maximize an evaluation function, namely the confidence score of the target class, by adjusting a single, randomly chosen, element, and then determining whether that adjustment improved the evaluation score or not. Every change that improves the evaluation score is accepted. The search for a better solution stops when the algorithm reaches a predefined maximum number of iterations. The computational complexity of this optimizer is $O(n)$, where $n$ is the total number of iterations. More details are shown in Appx. C.

*Adaptive Hill Climbing (Adaptive HC):* This optimizer performs hill climbing using adaptive changes on each element of the current solution, while also adapting the learning acceleration along the way. The algorithm stops either if the difference between two subsequent vectors' scores is smaller than a predefined value *epsilon* or it reaches the maximum number of iterations $max\_iterations$. The computational complexity of this optimizer is $O(nd)$, where $n$ is the number of iterations and $d$ is the latent space dimensionality. More details are shown in Appx. C.

*CMA-ES:* This optimizer utilizes the CMA-ES algorithm [25] for searching the CVAE's latent space. The computational complexity is $O(nm)$, where $n$ is the total number of iterations and $m$ is the function evaluations per iteration. For more details see Appx. C.

**All-class Black-box Optimizers.** *MAP-Elites:* This optimizer uses MAP-Elites to search the latent space, while maintaining diversity in the space of classes in order to perform an all-class attack.

**Table 2: An overview of the adversaries that conduct MI attacks in the image recognition field. The • and ○ symbols mean that the corresponding paper proposes MI attacks that fully or partially meet the column's point, respectively.**

| Adversary type | Use of surrogate models | Agnostic to target model's structure | Agnostic to target model's training data | Need for post-processing techniques | Need for a training dataset | All-class attack | Multiple solutions per target class | Attack spectrum: All / Part of target model's training data |
|---|---|---|---|---|---|---|---|---|
| Fredrikson et al. [19] | | ○ | ○ | • | | | | • / – |
| Aïvodji et al. [4] | • | • | • | • | | | | • / – |
| He et al. [26] | • | ○ | ○ | | • | | | • / – |
| Hitaj et al. [29] | | | • | | • | | | • / – |
| Zhang et al. [62] | | | ○ | | • | | | • / – |
| Salem et al. [48] | • | | | | • | | | – / • |
| Yang et al. [60] | | ○ | ○ | | • | | | • / – |
| **Random Sampling** | | • | • | | • | | | • / – |
| **HC** | | • | • | | • | | | • / – |
| **Adaptive HC** | | • | • | | • | | | • / – |
| **CMA-ES** | | • | • | | • | | | • / – |
| **MAP-Elites** | | • | • | | • | • | | • / – |
| **CVT-MAP-Elites** | | • | • | | • | • | • | • / – |

The computational complexity is $O(n)$, where $n$ is the total number of iterations. The algorithm finishes once it reached the predefined maximum number of iterations $max\_iterations$. The exact mutation and crossover techniques are discussed in Appx. C.

*CVT-MAP-Elites:* This optimizer uses CVT-MAP-Elites in order to maintain diversity in both the class and latent space in order to return multiple solutions that are well-spread. Note that this is not possible with MAP-Elites as it cannot effectively use 20 dimensions. The algorithm for this optimizer is the same as the one for MAP-Elites the only differences being the following: (a) we use a map of size $100 \times c \times 20$, 100 clusters with $c$ solutions per cluster −1 for each class− so we have 1,000 solutions for scenario 1 and 4,000 solutions for scenario 2, and (b) we generate the centroids for each cluster using a Normal distribution. The computational complexity of this optimizer is $O(n)$, where $n$ is the total number of iterations.

**Optimizers' Taxonomy.** Table 3 shows the sample-efficiency and the number of returned solutions for each black-box optimizer. Random sampling is just lucky guess so there is no smart optimization mechanism −low sample efficiency. HC is using isotropic Gaussian perturbations, but does not adapt the variance of the perturbations to accelerate optimization −moderate sample efficiency. Contrary, adaptive HC and CMA-ES are capable of adapting the perturbations per dimension −high sample efficiency. MAP-Elites sample-efficiency can be considered as moderate, since it requires a similar number of queries to return about as many solutions as HC would need for attacking all classes (see Table 5). CVT-MAP-Elites sample-efficiency could be considered as high, since it can attack many classes at once and return multiple, *diverse* solutions −it needs only $20K$ queries to return more solutions than adaptive HC when the latter attacks all classes multiple times to return multiple solutions (see Table 5). Note that adaptive HC does not have a mechanism to ensure the diversity of the returned solutions.

## 4 DEEP-BMI EVALUATION

**Evaluating MI Frameworks.** No single evaluation methodology for measuring the performance of an MI framework exists. This is mainly due to the probabilistic nature of MI attacks which lack of clear baselines. For instance, some papers conduct user studies with human participants [4, 19], whereas others utilize different metrics, like Peak Signal-to-Noise Ratio (PSNR) [62]. Moreover, some papers showcase some of their actual reconstructions, along with their ground-truth, for concluding about their attack's effectiveness [60].

**Table 3: Sample-efficiency & number of returned solutions per black-box optimizer. $c$ equals to the number of target classes and $k$ ($\geq 1$) is the number of solutions per class.**

| Optimizer | Sample-Efficiency (Low, Moderate, High) | No. of returned solutions |
|---|---|---|
| **Random Sampling** | Low | 1 |
| **HC** | Moderate | 1 |
| **Adaptive HC** | High | 1 |
| **CMA-ES** | High | 1 |
| **MAP-Elites** | Moderate | $c$ |
| **CVT-MAP-Elites** | High | $k \times c$ |

In this paper, we measure Deep-BMI's performance using all three evaluation approaches. First, in Fig. 8, we compare Deep-BMI's reconstructions with those reported by Yang et al., in training class inference attack setting. In addition, for each reconstruction, we provide the target model's confidence score of the ground-truth class.

Second, we conduct two user studies, one for each scenario, where, given a reconstructed by our MI framework image, we ask the participants to either identify the digit (0-9) depicted in it, or select the most similar looking face from the given alternatives. The participants can also respond that they "don't know" which digit is depicted in the given reconstructions or select that the target face is "not present" among the given alternatives. For both studies, we collect the responses from the same 22 individuals. From those individuals, 59.1% are males and the rest 40.9% females. In addition, 77.3% are in 18-25 age group and the rest 22.7% in 26-35. The participating individuals do *not* have any specific educational requirements and/or criteria.

Note that we have received an ethics approval for performing the user studies. For acquiring the approval we have submitted a documentation describing: (a) the full experimental setup, (b) the data that will be collected from the participants, and (c) any benefits/risks that participating individuals may experience, to the respective ethics department in our country. In addition, all data have been collected in an anonymous way.

Third, in Tables 4 and 6, we measure Deep-BMI's performance on each scenario using the following metrics [6]: (a) *PSNR*: ratio of an image's maximum squared pixel fluctuation over the mean squared error between the target and the reconstructed image, (b) *attack accuracy*: success rate of an evaluation classifier that

[6] Metrics (a-c) were proposed by Zhang et al. [62]. We omit considering KNN distance since Zhang et al. showed that it reports very similar to feature distance results.

predicts the class of the reconstructed inputs [7], (c) *feature distance*: $l_2$ feature distance between the reconstructed image and the centroid of the target class, and (d) Fréchet inception distance (FID): distance between feature vectors calculated for real and generated images. The higher the PSNR and attack accuracy the better. Similarly, the lower the feature distance and FID the better.

When comparing Deep-BMI to other baseline approaches we make sure that all the assumptions are even. For example, when comparing Deep-BMI to Yang et al.'s MI framework, which both operate in black-box setting and are based on a background dataset, we make sure that for both MI frameworks: (a) the attacker's capabilities are the same, (b) the target dataset is the same, and (c) the descriptively similar dataset is the same. Furthermore, for all MI frameworks, equivalent effort is spent for training the attacker models and tuning their hyperparameters. In fact, where possible, we utilize the authors' suggested optimal values.

For estimating Deep-BMI's *efficiency* we count the number of queries made to the target model and measure the time needed for conducting our attacks, same as [4, 19, 60]. The query budget, i.e., the total number of queries made to the target model, was empirically selected; increasing the query budget will lead to better reconstructions, and thus better performance for each optimizer. Note that we omit presenting the time required for training the attacker model since Deep-BMI, in contrast to Yang et al., can train this model off-line, without requiring any communication or interaction with the target. Thus, the training time is solely dependent on the attacker's computational power, e.g., multiple GPUs running in parallel, and does not influence the attack mechanics.

As explained in Sec. 2, large demands in terms of queries made to the target model is a drawback for black-box MI frameworks. As a result, a potential adversary may utilize a bot-net in order to perform those queries in a distributed manner, and thus mitigate this limitation. Nonetheless, doing so raises the overall complexity of the attack, while also limiting its practicality. Deep-BMI, on the other hand, aims to minimize the number of queries made to the target model by controlling them through the selected black-box optimizer. In addition, when using Deep-BMI, potential adversaries can either set an upper limit on the query budget based on their needs or halt the querying/attack process once the reconstructed image(s) yield(s) the target class(es) above a (predefined) confidence threshold. In our case, we vary the number of maximum iterations for each optimizer, which in turn affects the total number of queries made to the target model, until we get reconstructed digits and faces which are classified as their ground-truth class with high confidence score for *all* target classes. Thus, for each optimizer, the query budget is the *same* across all target classes.

**Experimental Setup.** We construct and train the attacker's model (CVAE –see Fig. 15, Appx. B) using the following parameter settings: (a) optimizer: Adam, with default settings [31], (b) batch size: 128, (c) loss function: binary cross entropy + KL-divergence [32]. The attacker's CVAE is composed of three convolutional layers for the encoder part. The first layer extracts low-level features, such as lines and edges. The second layer extracts multiple lines and shapes. Finally, the last layer extracts digits, for scenario 1, and faces, for scenario 2. Generally speaking, the abstraction of extracted features increases to higher orders analogous to the network's depth. For the target CNN (see Fig. 13, Appx. B), we use the following parameter settings: (a) optimizer: Adadelta, with default settings [61], (b) batch size: 64, (c) loss function: negative log likelihood. For performing our experiments we utilize PyTorch (version 1.4.0) [43] and a 4-core Xeon machine with 64GB of memory and no GPUs. The underlying OS is Ubuntu 18.04.5 LTS 64 bit.

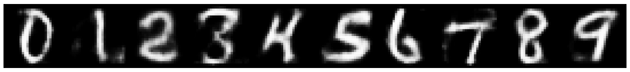


Figure 4: Digits' reconstructions from Deep-BMI.

## 4.1 Inverting Digit Recognition Models

Initially, we deploy Deep-BMI on a target of modest difficulty, but with low actual privacy implications, for showcasing its mechanics. In particular, we deploy Deep-BMI for inverting a handwritten digit recognition model. For this scenario, the target model is trained on MNIST, whereas the attacker model is trained on EMNIST-letters.

- MNIST consists of 70,000 handwritten digits having 10 classes in total, 1 for each digit 0-9, with a training set of 60,000 samples and a test set of 10,000 samples. The grayscale digits' images have been size-normalized and centred in a fixed-size image of 28 × 28 pixels.
- EMNIST-letters contains 145,600 characters in a total of 26 balanced classes with a training set of 124,800 samples and a test set of 20,800 samples. In particular, it contains handwritten *letters* derived from the NIST Special Database 19 and converted to a 28 × 28 grayscale image format that directly matches that of the MNIST.

**Effectiveness.** *Showing Actual Reconstructions.* As shown in Fig. 4, Deep-BMI reconstructs images which are identifiable as their ground-truth digits.

*User Study.* We compose a questionnaire with 18 questions in total (we *randomly* select 3 reconstructions per optimizer), where the participants are given reconstructed images and are asked to identify the digit that is depicted on each image, or select that they "don't know", and provide their confidence level when answering each question, i.e., Likert scale questions with 5 points [9]. Thus, we are able to perform a *qualitative* analysis that captures the performance of our framework, not only for reconstructing identifiable images, but also for the *fidelity* of those images.

We present the following graphs: (a) percentage of correct responses per black-box optimizer (Fig. 5), and (b) distribution of confidence identifying the reconstructed images (Fig. 6).

Fig. 5 shows the average performance for each black-box optimizer. Overall, all the optimizers perform comparably well. Adaptive HC is the most effective single-class optimizer achieving 98.48% success rate, which is 3.03% and 12.12% higher than the success rates achieved from the 2nd best and worst performing single-class optimizers, i.e., CMA-ES and random sampling, respectively. In addition, both all-class optimizers, i.e., MAP-Elites and CVT-MAP-Elites, perform equally well achieving 100% success rate. On average, our *single-* and *all-class* black-box optimizers reconstruct identifiable digits with *93.18%* and *100%* success rates, respectively. Finally, the percentage of "don't know" responses is 1.26%, meaning that the majority of reconstructed digits were identifiable.

[7] We follow Zhang et al.'s [62] instructions for training the evaluation classifier.

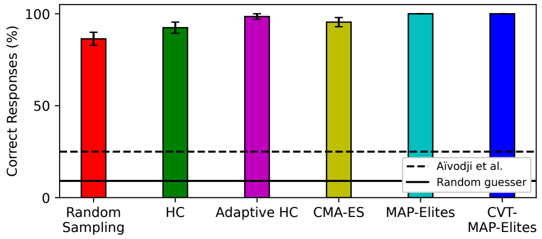


**Figure 5: Percentage of correct responses per black-box optimizer along with the standard error of the mean - Scenario 1.**

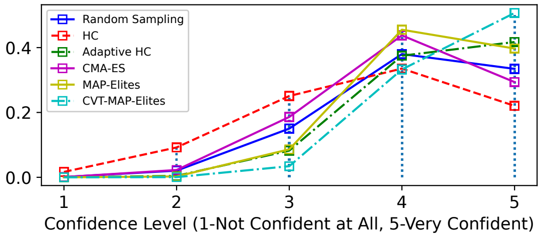


**Figure 6: Distribution of confidence per black-box optimizer - Scenario 1.**

**Table 4: The average results for each metric along with the 95% confidence interval for Deep-BMI - Scenario 1.**

| Metric | Deep-BMI |
|---|---|
| **PSNR** | 11.61 (11.58, 11.64) |
| **Feature Distance** | 298.68 (297.76, 299.60) |
| **Attack Accuracy** | 98% |
| **FID** | 124.74 (123.91, 125.57) |

Overall, for this scenario, MAP-Elites and CVT-MAP-Elites are the most effective black-box optimizers achieving 1.51% higher attack success rate compared to the best performing single-class optimizer, i.e., adaptive HC. Note, however, that *all* optimizers achieve dramatically higher attack success rates compared to those reported by Aïvodji et al. [4].

Fig. 6 shows the distribution of confidence for scenario 1. The graph peaks at the "very confident" and "confident" answers for *all* optimizers. In particular, 78.03% and 94.69% of the respondents felt "very confident" or "confident" identifying the reconstructed digits derived from our single- and all-class optimizers, respectively. Only 6.06% and 3.03% of the respondents felt "not confident" or "not confident at all" identifying the reconstructed images from our single- and all-class optimizers, respectively.

*Quantitative Metrics.* For completeness, in Table 4, we measure Deep-BMI's effectiveness using the relevant quantitative metrics, despite that: (a) scenario 1 is used primarily as a proof-of-concept for showcasing Deep-BMI's mechanics, and (b) we do not have any direct comparison to other approaches found in the literature.

**Efficiency.** The efficiency results for single- and all-class black-box optimizers are depicted in Table 5. As shown, *all* of them are *efficient* managing to reconstruct identifiable digits in only *1.29 seconds* (CMA-ES) for single-class and *270.89 seconds* (MAP-Elites) for all-class attack. For completeness, we also provide the total number of queries made to the target model for each optimizer. As shown, the deployed optimizers are capable of conducting MI attacks with only *388* queries (adaptive HC) for single-class and *20,000* queries (MAP-Elites & CVT-MAP-Elites) for all-class attack. On average, our single- and all-class black-box optimizers require *16.41 & 295.66 seconds*, and *25,448 & 20,000 queries*, respectively.

**Table 5: Single- and all-class black-box optimizers' time and queries requirements - Scenario 1.**

| | Optimizer | Time (sec) | No. of Queries |
|---|---|---|---|
| **Single-class** | **Random Sampling** | 58 | 100,000 |
| | **HC** | 4.76 | 1,000 |
| | **Adaptive HC** | 1.59 | 388 |
| | **CMA-ES** | 1.29 | 404 |
| **All-class** | **MAP-Elites** | 270.89 (4.5 min.) | 20,000 |
| | **CVT-MAP-Elites** | 320.43 (5.3 min.) | 20,000 |

**Sum up.** For this scenario, we show that Deep-BMI supports both effective and efficient black-box optimizers. The most effective optimizers are MAP-Elites and CVT-MAP-Elites achieving *100%* success rate on average. The most efficient optimizer, in terms of computational time, is CMA-ES requiring *1.29 seconds* (and *404 queries*) for attacking a specific target class. Observing all the results reported in this section, one can easily see that handwritten digit recognition ML models can be *practically* inverted.

## 4.2 Inverting Facial Recognition Models

Facial recognition ML models are functions that map a given image containing a face to a specific identifier corresponding to the individual depicted in the image [19]. These models are increasingly used for user authentication [14] and subject surveillance [51]. A growing number of web APIs support facial recognition [19].

In this section, we deploy Deep-BMI on a facial recognition ML model to violate its security as well as the privacy of the individuals whose faces' images have been included in the target model's training dataset. For this scenario, the target model is trained on AT&T, whereas the attacker model is trained on CelebA.

- AT&T contains 10 grayscale images of 40 individuals (classes) in various lightning conditions, facial expressions, and details, for a total of 400 images. AT&T was used by Fredrikson et al. [19] as well as from various other papers [10, 37, 40, 45, 53]. We divide the images of each person into a training and a validation set consisting of 7 and 3 images, respectively, same as [19]. Next, we train the target model on the training set and evaluate its accuracy on the validation set.
- CelebA includes 202,599 face images belonging to 10,177 different individuals (classes) covering large pose variations and background clutter. CelebA is a popular choice for different computer vision tasks, such as face attribute recognition, face detection, and landmark localization [35]. We convert the images of CelebA in grayscale to match the format of AT&T.

Fig. 7 shows the reconstructed images for an individual using Fredrikson et al.'s [19] white-box MI attacks (a,i) and Deep-BMI (b,ii). As shown, Fredrikson et al. reconstruct recognizable faces having, however, *white-box* access to the target model, in contrast to Deep-BMI which acts in the black-box setting. In our case, we could utilize the descriptively similar dataset to average the samples yielding the target class with high confidence (b,i), achieving, however, poor results, which are similar to those reported by Yang et al. [60] (see Fig. 8), and only with Deep-BMI (b,ii) we can approximate the performance of Fredrikson et al.

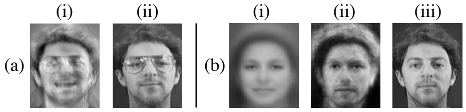


**Figure 7: Reconstructed images for a specific individual, using Fredrikson et al.'s [19] white-box attacks (a,i) and Deep-BMI black-box attacks (b,ii). (b,i) is the reconstructed image when averaging the samples of the descriptively similar dataset that yield the target class with confidence ≥ 0.6. (a,ii) & (b,iii) are the ground-truth images of the target individual.**

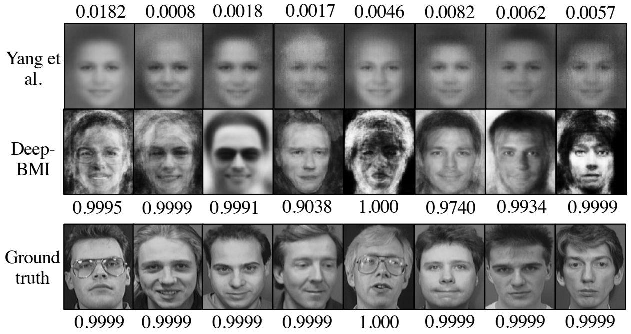


**Figure 8: Recovered faces of victims from Yang et al.'s MI framework [60] and Deep-BMI, along with their ground-truth images and the target model's confidence scores.**

**Effectiveness.** *Showing Actual Reconstructions.* Fig. 8 shows a comparison of our reconstructions and those reported by Yang et al. [60], in *training class inference* attack setting. As shown, Deep-BMI reconstructs images with recognizable characteristics, whereas Yang et al.'s approach reconstructs images which only preserve general faces' attributes, such as eyes, mouth, nose, without being easily identifiable as their ground-truth individuals. For example, observe that some of Yang et al.'s reconstructions look very similar between them although referring to different target individuals. As expected, Yang et al.'s reconstructions are somehow similar to those observed when averaging the samples of the background dataset that yield each target class above a confidence threshold (Fig. 7(b,i)).

Notice that the target model's confidence scores on the ground-truth images are close, and in many cases identical, to those observed when feeding as input Deep-BMI's reconstructions. This fact makes Deep-BMI's reconstructions possible to replace the originals and this can have further security implications. For example, one can evade authentication systems based on face recognition that leverage the target model's confidence scores [1–3] [8]. Contrary, the target model's confidence scores on the ground-truth images differ significantly to those observed when feeding as input Yang et al.'s reconstructions.

*User Study.* We compose a questionnaire with 22 questions in total (we *randomly* select 3 reconstructions per optimizer), having, however, different structure compared to the previous scenario (Sec. 4.1). In particular, we utilize the *same structure* as the one reported by Fredrikson et al. [19]. That is, we ask participants to match each reconstructed image to 1 of the 5 given options from the AT&T set, or to respond that the displayed image does not correspond to any of the provided alternatives. In 80% of the experiments, 1 of the 5 images contained the individual corresponding to the reconstructed image, the rest were randomly selected. As a control, 10% of the experiments used a plain image from the AT&T dataset rather than one produced by Deep-BMI. This allowed us to gauge the baseline ability of the participants at matching faces from the training set [9]. Finally, the remaining 10% of the experiments contained a reconstructed image that did not correspond to any of the 5 given options. Notice that this is the *same layout of questions* as the one used by Fredrikson et al. [19]. In addition, the respondents are asked to provide their confidence level when answering each question, i.e., Likert scale questions with 5 points [9]. Thus, we are able to capture Deep-BMI's performance, not only for reconstructing identifiable faces, but also for their *fidelity*.

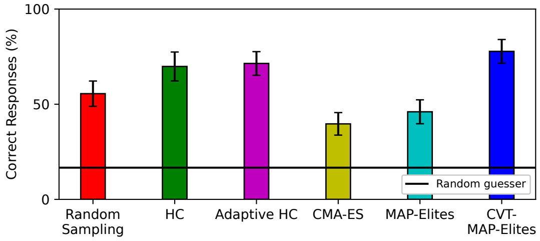


**Figure 9: Percentage of correct responses per black-box optimizer along with the standard error of the mean - Scenario 2.**

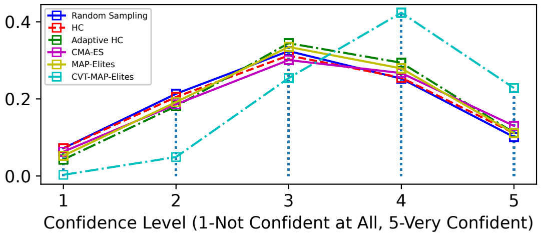


**Figure 10: Distribution of confidence per black-box optimizer - Scenario 2.**

We present the following graphs: (a) percentage of correct responses per black-box optimizer (Fig. 9), (b) distribution of confidence identifying the reconstructed images (Fig. 10), and (c) percentage of *overall*: all correct responses, i.e., the respondent selected the image corresponding to the individual targeted in the attack when present, and otherwise selected "not present", *identified*: instances where the targeted individual was displayed among the test images, and the respondent identified the correct image, and *excluded*: instances where the targeted individual was not displayed, and the respondent correctly answered "not present", metrics (Fig. 11).

Fig. 9 shows the average performance for each black-box optimizer. Adaptive HC is the most effective single-class optimizer achieving 71.42% success rate, 1.58% higher than HC. In addition, CVT-MAP-Elites is the most effective all-class optimizer achieving 77.77% success rate, 31.74% higher than MAP-Elites. On average, our *single-* and *all-class* black-box optimizers reconstruct identifiable faces with *59.12%* and *61.90%* success rates, respectively. Overall, for this scenario, CVT-MAP-Elites is the most effective black-box optimizer achieving 6.34% higher attack success rate than adaptive HC.

[8] Exploring whether or not specific authentication systems based on face recognition can authenticate the generated images as specific individuals is *out of scope*.

[9] All the participants correctly identified all given actual (non-inverted) control images.

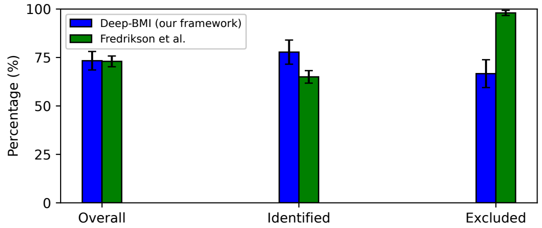


**Figure 11: Percentage of overall, identified & excluded metrics for Deep-BMI and Fredrikson et al. [19] - Scenario 2.**

Fig. 11 shows the success rate for overall, identified and excluded metrics, which are the same metrics used by Fredrikson et al. [19]. As shown, Deep-BMI achieves 73.33% *overall* accuracy, up to 77.77% *identification* rate and 66.66% *exclusion* rate. These results are comparable to those reported by Fredrikson et al.'s white-box MI attacks on a target model of similar complexity. As a result, Deep-BMI shows that *black-box* MI attacks, that minimize the adversarial assumptions, can be practical even on a target of higher complexity, such as face recognition models.

Fig. 10 shows the distribution of confidence for scenario 2. The graph peaks at the "neither confident nor not confident" answer for all optimizers except from CVT-MAP-Elites for which it peaks at the "confident" answer. In particular, 50.39% and 57.93% of the respondents felt "very confident" or "confident" identifying the reconstructed faces derived from our single- and all-class optimizers, respectively. Only 24.60% and 10.31% of the respondents felt "not confident" or "not confident at all" identifying the reconstructed faces from our single- and all-class optimizers, respectively. The drop of the respondents' confidence, compared to the previous scenario, was expected. This is because this scenario's complexity is higher than the first scenario. However, the large majority of the respondents still matches the reconstructed by our framework faces with the target ones correctly (see Fig. 11).

*Quantitative Metrics.* Table 6 compares Deep-BMI to the black-box MI framework proposed by Yang et al., using the relevant quantitative metrics. As shown, Deep-BMI reconstructs images that: (a) are less distant from their ground-truth in term of both feature distance and FID, and (b) expose 8% more private information compared to Yang et al.'s approach (see attack accuracy). The only case where Deep-BMI falls behind is for the PSNR metric for which Yang et al. achieve 4.51 higher PSNR.

**Efficiency.** The efficiency results for single- and all-class black-box optimizers are depicted in Table 7. As shown, Deep-BMI is still efficient, even when facing a more complex target, managing to reconstruct identifiable faces in *21.24 seconds* (adaptive HC) for single-class and *158.23 minutes* (MAP-Elites) for all-class attack. For completeness, we also provide the total number of queries made to the target model for each optimizer. As shown, the deployed optimizers are capable of conducting MI attacks with only *230* queries (adaptive HC) for single-class and *10,000* queries (MAP-Elites & CVT-MAP-Elites) for all-class attack. Deep-BMI requires *244,076 & 234,306 fewer queries* compared to Yang et al.'s black-box MI framework, when using adaptive HC & CVT-MAP-Elites, respectively, for a target of similar complexity. On average, our *single-* and *all-class* black-box optimizers require *3.8 minutes & 5.5 hours*, and *5,464 & 10,000 queries*, respectively.

**Table 6: The average results for each metric along with the 95% confidence interval for Deep-BMI and Yang et al.'s black-box MI framework - Scenario 2. Bold values indicate that the corresponding framework has better performance.**

| Metric | Yang et al. [60] | Deep-BMI |
|---|---|---|
| **PSNR** | **18.01 (17.92, 18.09)** | 13.50 (13.38, 13.62) |
| **Feature Distance** | 419.36 (412.15, 426.57) | **403.66 (398.01, 409.31)** |
| **Attack Accuracy** | 53% | **61%** |
| **FID** | 263.75 (259.08, 268.43) | **223.16 (218.54, 227.77)** |

**Table 7: Single- and all-class black-box optimizers' time and queries requirements - Scenario 2.**

| | Optimizer | Time (sec) | No. of Queries |
|---|---|---|---|
| **Single-class** | **Random Sampling** | 339.82 (5.6 min.) | 20,000 |
| | **HC** | 374.38 (6.2 min.) | 1,000 |
| | **Adaptive HC** | 21.24 | 230 |
| | **CMA-ES** | 181.42 (3 min.) | 625 |
| **All-class** | **MAP-Elites** | 9,494.01 (2.6 hrs.) | 10,000 |
| | **CVT-MAP-Elites** | 30,175.05 (8.3 hrs.) | 10,000 |

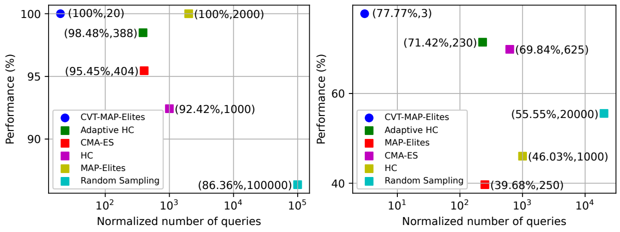
**Sum up.** For this scenario, we show that Deep-BMI reconstructs identifiable faces with recognizable facial characteristics (see Fig. 8). According to the user study, Deep-BMI achieves *60.05%* success rate, on average; again the most effective black-box optimizer being CVT-MAP-Elites achieving *77.77%* success rate, on average. Note that CVT-MAP-Elites reconstructs faces that cause the respondents to answer with higher confidence compared to other optimizers. In particular, 65.07% of the respondents felt "very confident" or "confident" when identifying CVT-MAP-Elites's reconstructed images, which is 4.76% higher than the 2nd highest "very confident" or "confident" yielding optimizer, namely CMA-ES.

Adaptive HC is the most efficient black-box optimizer, in terms of computational time, requiring *21.24 seconds* (and *230 queries*) for attacking a specific target class, while also achieving high reconstruction success rate (71.42%). Observing all the results reported in this section, we can see that Deep-BMI manages to conduct practical black-box MI attacks on facial recognition ML models.

## 5 DISCUSSION

**Comparing the Black-box Optimizers.** Fig. 12 shows the average performance vs. normalized number of queries (total no. of queries divided by the no. of returned reconstructions/solutions – see Table 3), per black-box optimizer for scenarios 1 & 2 [10]. As shown in Fig. 12(a), CVT-MAP-Elites is the most effective and sample-efficient black-box optimizer achieving 100% attack success rate and requiring 20 queries per solution, for scenario 1. CVT-MAP-Elites achieves 1.51% higher success rate and requires 368 fewer queries than the 2nd most effective and sample-efficient optimizer (adaptive HC). Note that adaptive HC and CMA-ES achieve comparable performance in terms of both effectiveness and sample-efficiency requiring, however, 19× and 20× more queries to be made to the target model than CVT-MAP-Elites per returned solution.

[10] Fig. 12 is used primarily to compare the performance of utilized black-box optimizers. Indeed, returning an identifiable reconstruction for a specific individual may require more queries compared to another individual. However, this fact holds for all optimizers used. Thus, when *comparing* the optimizers, taking the average performance is enough.



(a) Scenario 1

(b) Scenario 2

**Figure 12: Average performance vs. normalized number of queries per black-box optimizer. Circles represent the non-dominated ones.**

As shown in Fig. 12(b), CVT-MAP-Elites is the most effective and sample-efficient black-box optimizer achieving 77.77% attack success rate and requiring 3 queries per solution, for scenario 2. In particular, CVT-MAP-Elites achieves 6.34% higher success rate and requires 227 fewer queries than the 2nd most effective and sample-efficient optimizer (adaptive HC). Note that adaptive HC and CMA-ES achieve similar performance as well (> 69%), while also requiring a low number of queries to the target model (≤ 625).

Observing Fig. 12, we can see that for both scenarios CVT-MAP-Elites is the most effective and sample-efficient black-box optimizer. These results were expected as CVT-MAP-Elites can be successfully applied to *high-dimensional* feature spaces returning multiple (diverse) solutions per cluster. Thus, the respondents had multiple reconstructed images (we showed them 5 for each class) to decide about the digit or the face that was depicted on them.

However, as shown in Fig. 12, *other* black-box optimizers achieve similar effectiveness and sample-efficiency as well. For example, adaptive HC achieves 98.48% & 71.42% attack success rate for scenarios 1 & 2, respectively. Furthermore, in terms of sample-efficiency, adaptive HC requires *388 & 230 queries* to the target model for scenarios 1 & 2, respectively. A similar story holds for CMA-ES.

In terms of computational time, adaptive HC is the *most efficient* optimizer requiring *1.59 & 21.24 seconds* for attacking a specific target class, for scenarios 1 & 2, respectively. We omit presenting the average performance vs. *time* comparison per optimizer since their implementation could be optimized, e.g., by using GPUs.

Overall, if we were to prescribe what optimizer to use, we would say adaptive HC when attacking a single-class and we want fast results, and CVT-MAP-Elites when attacking all-classes and we want many solutions per class. In other words, CVT-MAP-Elites is sample-efficient but only when returning multiple solutions.

**Applying Deep-BMI on Face Embeddings Target Models.** Although the target models used in this work have been also utilized in similar studies [4, 26, 60, 62], it does not necessarily mean that our research will be immediately relevant for practical approaches to face recognition. For example, state-of-the-art face recognition models, e.g., facenet [52], either allow to fine-tune a pre-trained model and select the input's identity based on the model's output logits or utilize face embeddings for comparing the distance of the input's embedding to the embeddings of the prototypes. In our paper, we mainly present experiments for the former case. However, Deep-BMI can be also applied to the latter scenario, the only difference being that the optimizers will be adapted in order to search for the solution (image) that *minimizes* the distance between the reconstructed image's embedding and the embedding of the target image. While presenting experiments with such face embeddings models is indeed interesting, we consider it as *out of scope*.

**Limitations and Future Work.** Although, in this paper, we showcase the possibility of conducting successful black-box MI attacks in training class inference setting, specific aspects/assumptions on which both Deep-BMI and Yang et al. build upon still remain unexplored. Below, we elaborate on these assumptions/limitations, which directly affect the practicality of Deep-BMI and Yang et al.'s MI framework, and provide directions for future research. The assumptions/limitations discussed below cover three main angles: (a) the access to a background dataset which is descriptively similar to the target dataset, (b) the utilized generative (inversion) model and (c) the access to the target model's top-1 confidence score.

*Access to a Descriptively Similar Dataset.* First, obtaining a background dataset which is descriptively similar to the target dataset might be hard in certain cases, e.g., in biomedical settings. Thus, the adversary may manage to acquire only a very small set of training samples, which is by no means representative of the overall distribution. Consequently, an experimental analysis showcasing the effect of the background dataset's size to the performance of black-box MI frameworks, such as Deep-BMI, is an interesting direction.

Second, the properties/attributes that a descriptively similar dataset must have, still remain unclear. Providing a detailed analysis in regards to this dataset's exact characteristics, and how they can interplay with the mechanics of Deep-BMI, or the attack presented by Yang et al., will aid in demystifying the best way of determining the level of similarity between the target and the background dataset, and thus establish guidelines on how to find such a dataset.

Third, an exhaustive exploration of how the level of similarity between the target and the background dataset affects the MI performance is absent from the literature. For example, what is an acceptable level of similarity between the two datasets for acquiring satisfactory results? Is it possible to reconstruct identifiable images even when having access to dataset from the same domain, but with low similarity to the target dataset? These questions, if cleared up, will significantly contribute to determining the real-world practicality of such background dataset-dependent MI frameworks.

*The Utilized Generative (Inversion) Model.* Deep-BMI's inversion performance depends, not only on the descriptively similar dataset, but also on the capability of the utilized generative (inversion) model. Tuning the generative model indeed affects, to some extent, the overall MI performance. However, what is important in our approach is the general methodology of the MI attack and how the different components interplay, not the selected generative model per se. That is, in Deep-BMI one can potentially replace CVAEs with other generative models as well. Exploring how the various parameters of the CVAE, and potentially of other generative models, affect the overall MI performance is considered as future work.

The choice of the generative model depends on the target domain. For example, in the computer vision domain, one can utilize VAEs, GANs, flow-based models, autoregressive models or energy-based models [8]. For determining whether or not a particular generative model is suitable, by means of achieving a minimum required performance, one should do the following. First, measure its effectiveness on the samples included in the background dataset using relevant quantitative and qualitative metrics. For example, one can use Fréchet inception distance or show a batch of synthetic images and inspect (visually) their quality and deviation from the ground-truth. Showcasing synthetic images will help us determine whether the generative model only produces blurry images which could be the result of averaging over a specific subset. Generally, the selected model should generate images with the variation and quality of the ones contained in the background dataset. Second, examine whether it can generalize with respect to producing confidently classified images, for each target class, after searching its latent space using various black-box optimizers. If this is not the case, one must either utilize a different generative model or enrich the background dataset since generated images that yield a target class with low confidence score are not representative of the respective class.

The size and shape of the latent space may render our MI attacks ineffective. In particular, the fraction of the latent space that can be navigated by our black-box optimizers, along with the sparsity of the solutions which are close to the natural ones, might impact the MI performance. An in-depth analysis of how the size and shape of the latent space specifically impacts the MI performance is not considered trivial and we plan to explore it in our future work.

Training generative models is a hard and time-consuming procedure, especially in the computer vision domain. The computational overhead required for training the generative model is directly related to the complexity of both the selected model and the background dataset. For the experiments presented in this paper, the required computational overhead is within the range of a determined individual. However, larger architectures trained on bigger and more complex datasets may require more powerful resources.

*Access to the Target Model's Top-1 Confidence Score.* Both Deep-BMI and Yang et al. use the top-1 confidence score, i.e., a continuous value, instead of a discrete label about the winning class. Thus, a potential countermeasure that will decrease the risk against such black-box MI frameworks is restricting the target model to only return the predicted label of the winning class without reporting any of the probabilities [11]. As a result, developing fully black-box MI frameworks that can operate using *only* the predicted label of the winning class, without requiring any of the target model's confidence scores, is another interesting direction for future research.

## 6 RELATED WORK

Fredrikson et al. [20] proposed the first black-box MI attack on sensitive genomic data. Their attack, however, is only applicable to low-complexity models and datasets with a limited number of sensitive target features which are drawn from small domains [19]. In a subsequent study, Fredrikson et al. [19] further demonstrated the severe consequences of this threat by conducting MI attacks on face recognition models and retrieving recognizable facial features for individuals contained in the target model's dataset. However, the majority of those attacks are white-box and the one black-box attack is both ineffective, it produces semantically meaningless reconstructions [60], and inefficient, it needs 50-80 days to complete.

Yang et al. [60] proposed black-box MI attacks utilizing a descriptively similar dataset and reconstructing recognizable images in data reconstruction attack setting. However, Yang et al.'s attacks reconstruct images with low recognizability in training class inference attack setting (Fig. 8). In this paper, we focus solely on the latter setting and show that it is possible to reconstruct digits and faces with recognizable characteristics at a fraction of the queries required by Yang et al. For example, Deep-BMI's CVT-MAP-Elites requires 10,000 queries for reconstructing recognizable faces, whereas Yang et al. require 244,306. Thus, Deep-BMI requires *234,306 fewer queries* compared to Yang et al.'s approach. Yang et al. avoid conducting a user study for evaluating their MI framework's effectiveness, and thus we omit presenting a detailed comparison in terms of overall, identified and excluded metrics with Deep-BMI and Fredrikson et al. [19]. Instead, the authors visualize the reconstructed images of *specific* individuals along with their ground-truth (Fig. 8).

Aïvodji et al. [4] proposed a black-box MI framework, namely GAMIN, targeting mid-complexity models and datasets. GAMIN infers identifiable digits with 25% success rate on average. In contrast, Deep-BMI achieves 95.45% success rate on average, that is, 70.45% higher than GAMIN. Moreover, GAMIN fails on conducting MI on face recognition models since its reconstructions do not form recognizable faces (see Figs. 9-11 in [4]); the authors manage to reconstruct only blurry contours of the human face without any recognizable characteristics of the target individuals. This is the main reason why we choose *not* to present a detailed comparison of Deep-BMI with GAMIN. Finally, GAMIN imposes significant computational overhead which is mainly associated with: (a) training, on-line, a surrogate model for mimicking the target model's behaviour, (b) training multiple different attacker models, one for each target class, and (c) issuing computationally-intensive image post-processing techniques for improving the quality of its reconstructions; Deep-BMI is devoid of all these operations.

Zhang et al. [62] proposed a generative MI attack for inverting deep ANNs with high success rates. Nonetheless, their MI attacks are white-box based and in some cases depend on a set of blurred or partially blocked images from the target model's training set, which are the sensitive images to be inferred.

Hitaj et al. [29] and He et al. [26] showed that collaborative learning systems are susceptible to both white-box and black-box MI attacks. However, the quality of the recovered inputs, for black-box MI attacks and different data distributions, is poor [26].

Hidano et al. [28] conduct white-box MI attacks by injecting malicious data samples into the target model's training dataset in order to alter its decision approximation. However, the authors evaluate their MI attacks on two low-complexity ML models and datasets similar to Fredrikson et al. [20].

Salem et al. [48] proposed four MI attacks that manage to infer diverse information about an updating set, by exploiting the difference in the output of a black-box target model before and after being updated.

## 7 CONCLUSION

In this paper, we presented a feasibility study for the effectiveness and efficiency of black-box MI attacks on mid-complexity targets. In this context, we introduced Deep-BMI, a modular MI framework that supports various black-box optimizers which can practically invert image recognition models. This fact raises major concerns regarding the security/privacy of the widely deployed ML models.

[11] Note that applying this countermeasure significantly *degrades* the utility/usefulness of the information provided by the target model [54].

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive feedback. We also thank the shepherd for helping us to improve the final version of this paper. This work was supported by the European Union's Horizon 2020 research and innovation programme under grant agreements No. 830929 (CyberSec4Europe), No. 739578 (RISE), No. 101007673 (RESPECT) and the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

## REFERENCES

[1] 2021. Amazon Rekognition. https://aws.amazon.com/rekognition/

[2] 2021. Clarifai. https://docs.clarifai.com

[3] 2021. Google Cloud Vision. https://cloud.google.com/vision

[4] Ulrich Aïvodji, Sébastien Gambs, and Timon Ther. 2019. GAMIN: An Adversarial Approach to Black-Box Model Inversion. *arXiv preprint arXiv:1909.11835* (2019).

[5] Idan Amit, John Matherly, William Hewlett, Zhi Xu, Yinnon Meshi, and Yigal Weinberger. 2018. Machine learning in cyber-security-problems, challenges and data sets. *arXiv preprint arXiv:1812.07858* (2018).

[6] Giuseppe Ateniese, Luigi V. Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. 2015. Hacking Smart Machines with Smarter Ones: How to Extract Meaningful Data from Machine Learning Classifiers. *Int. J. Secur. Netw.* 10, 3 (2015), 137–150.

[7] Hans-Georg Beyer and Hans-Paul Schwefel. 2002. Evolution strategies–A comprehensive introduction. *Nat. Comput.* 1, 1 (2002), 3–52.

[8] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. 2021. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. *TPAMI* (2021), 1–1.

[9] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usab. Eval. in Ind.* 189, 194 (1996), 4–7.

[10] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, Vol. 1. 539–546.

[11] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. 2017. EMNIST: an extension of MNIST to handwritten letters. *arXiv preprint arXiv:1702.05373* (2017).

[12] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015), 503–507.

[13] Antoine Cully and Yiannis Demiris. 2017. Quality and diversity optimization: A unifying modular framework. *TEVC* 22, 2 (2017), 245–259.

[14] MA Dabbah, WL Woo, and SS Dlay. 2007. Secure authentication for face recognition. In *CIISP*. 121–126.

[15] Antreas Dionysiou, Michalis Agathocleous, Chris Christodoulou, and Vasilis Promponas. 2018. Convolutional Neural Networks in Combination with Support Vector Machines for Complex Sequential Data Classification. In *ICANN*. 444–455.

[16] Antreas Dionysiou and Elias Athanasopoulos. 2021. Unicode Evil: Evading NLP Systems Using Visual Similarities of Text Characters. In *AISEC*. 1–12.

[17] Carl Doersch. 2016. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016).

[18] Andries P Engelbrecht. 2007. *Computational intelligence: an introduction.*

[19] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*. 1322–1333.

[20] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *USENIX Security*. 17–32.

[21] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *CCS*. 619–633.

[22] Nikolaus Hansen, Youhei Akimoto, and Petr Baudis. 2019. CMA-ES/pycma on Github. Zenodo, DOI:10.5281/zenodo.2559634.

[23] Nikolaus Hansen, Anne Auger, Raymond Ros, Steffen Finck, and Petr Pošík. 2010. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *SIGEVO*. 1689–1696.

[24] Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. 2003. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* 11, 1 (2003), 1–18.

[25] Nikolaus Hansen and Andreas Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* 9, 2 (2001), 159–195.

[26] Zecheng He, Tianwei Zhang, and Ruby B. Lee. 2019. Model Inversion Attacks against Collaborative Inference. In *ACSAC*. 148–162.

[27] Seira Hidano, Takao Murakami, Shuichi Katsumata, Shinsaku Kiyomoto, and Goichiro Hanaoka. 2017. Model inversion attacks for prediction systems: Without knowledge of non-sensitive attributes. In *PST*. 115–11509.

[28] Seira Hidano, Takao Murakami, Shuichi Katsumata, Shinsaku Kiyomoto, and Goichiro Hanaoka. 2018. Model inversion attacks for online prediction systems: Without knowledge of non-sensitive attributes. *TOIS* 101, 11 (2018), 2665–2676.

[29] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *CCS*. 603–618.

[30] Mahdi Khosravy, Kazuaki Nakamura, Naoko Nitta, and Noboru Babaguchi. 2020. Deep face recognizer privacy attack: Model inversion initialization by a deep generative adversarial data space discriminator. In *APSIPA ASC*. 1400–1405.

[31] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[32] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[33] Diederik P Kingma, Max Welling, et al. 2019. An introduction to variational autoencoders. *Found. Trends Mach. Learn.* 12, 4 (2019), 307–392.

[34] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. of the IEEE* 86, 11 (1998), 2278–2324.

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *ICCV*. 3730–3738.

[36] James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. on Math. Statist. and Prob.*, Vol. 1. 281–297.

[37] Tanaya Mandal, Angshul Majumdar, and QM Jonathan Wu. 2007. Face recognition by curvelet based feature extraction. In *ICIAR*. 806–817.

[38] Brian Mc Ginley, John Maher, Colm O'Riordan, and Fearghal Morgan. 2011. Maintaining healthy population diversity using adaptive crossover, mutation, and selection. *TEVC* 15, 5 (2011), 692–714.

[39] Shagufta Mehnaz, Ninghui Li, and Elisa Bertino. 2020. Black-box model inversion attribute inference attacks on classification models. *arXiv preprint arXiv:2012.03404* (2020).

[40] Andrea Melle and Jean-Luc Dugelay. 2014. Scrambling faces for privacy protection using background self-similarities. In *ICIP*. 6046–6050.

[41] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).

[42] Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*. 427–436.

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*. 8024–8035.

[44] P Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J Rauss. 1998. The FERET database and evaluation procedure for face-recognition algorithms. *Image Vis. Comput.* 16, 5 (1998), 295–306.

[45] Roberto Pieraccini, Esther Levin, and Wieland Eckert. 1997. AMICA: The AT&T mixed initiative conversational architecture. In *EUROSPEECH*. 1875–1878.

[46] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. 2016. Quality diversity: A new frontier for evolutionary computation. *Front. in Robotics and AI* 3 (2016), 40.

[47] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. *Learning internal representations by error propagation.* Technical Report. California Univ. San Diego La Jolla Inst. for Cognitive Science.

[48] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. 2020. Updates-Leak: Data Set Inference and Reconstruction Attacks in Online Learning. In *USENIX Security*. 1291–1308.

[49] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *NDSS*.

[50] Ferdinando S Samaria and Andy C Harter. 1994. Parameterisation of a stochastic model for human face identification. In *WACV*. 138–142.

[51] Charles Savage. 2013. Facial scanning is making gains in surveillance. *The New York Times* (2013).

[52] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*. 815–823.

[53] J Shermina. 2011. Face recognition system using multilinear principal component analysis and locality preserving projection. In *GCC*. 283–286.

[54] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *SP*. 3–18.

[55] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction apis. In *USENIX Security*. 601–618.

[56] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. 2017. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *TEVC* 22, 4 (2017), 623–630.

[57] Vassiiis Vassiliades and Jean-Baptiste Mouret. 2018. Discovering the elite hypervolume by leveraging interspecies correlation. In *SIGEVO*. 149–156.

[58] Xi Wu, Matthew Fredrikson, Somesh Jha, and Jeffrey F Naughton. 2016. A methodology for formalizing model-inversion attacks. In *CSF*. 355–370.

[59] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[60] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. 2019. Neural network inversion in adversarial setting via background knowledge alignment. In *CCS*. 225–240.

[61] Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).

[62] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *CVPR*. 253–261.

## A BLACK-BOX OPTIMIZATION METHODS

**Hill-climbing.** Optimizing black-box models of real numbers, as in our case, requires the use of stochastic, derivative-free continuous optimizers. One such family of black-box optimizers is hill-climbing, which is also known as greedy local search. In its simplest form, the algorithm starts with an arbitrary solution and tries to find a better solution by making incremental changes, that is, adding noise derived from a Gaussian distribution either to all features or a randomly chosen feature of the solution. If that stochastic change produces a better solution, then the new solution is kept. This procedure is repeated until the algorithm reaches some stopping criteria, e.g., a maximum number of iterations. Variants of this algorithm can adapt the noise distribution or the learning rate per feature until no further significant improvements can be observed. Thus, such *adaptive* hill climbing algorithms can converge faster, even offering better solutions.

The simplicity of hill-climbing makes it a good approach for tackling a wide range of simple problems. However, its greedy nature can be a disadvantage in many problems that have non-convex functions with multiple optima, ridges and plateaus. As a result, evolutionary algorithms have emerged as global optimizers, with better mechanisms for escaping local optima.

**Evolutionary Algorithms (EAs).** EAs have become a popular choice when it comes to dealing with non-convex optimization problems that cannot be solved by gradient-based methods [18]. An EA is based on the principle of biological evolution, using mechanisms such as selection, variation and replacement. The procedure for a simple EA goes as follows. A population of solutions is randomly initialized and evaluated. The fitness (performance) of each solution is calculated by the fitness (objective) function. In each generation, new individuals (candidate solutions or offspring) are formed by variation (recombination and/or mutation) of selected parental individuals, and replace the least fit individuals. This process is repeated until reaching some stopping criteria.

EAs are inherently robust due to their population-based approach which offers them better exploration capabilities and more chances in finding a global optimum. Moreover, EAs can be easily parallelized, thus, having increased efficiency in modern hardware [24]. In this work, we use algorithms from two EA families, namely *evolution strategies* [7] and *quality-diversity* [12, 13, 41, 46, 56, 57].

Evolution strategies aim to accelerate evolutionary optimization by augmenting the candidate solution with certain "strategy parameters", e.g., mutation strength per feature, that are adapted over time. CMA-ES [25] is a state-of-the-art algorithm that has experimentally shown advantageous convergence properties across a wide range of problems [23]. It uses a multivariate Gaussian search distribution to sample new candidate solutions and adapts its mean towards the direction of fitter solutions and the covariance matrix in a way that increases the likelihood of previously successful search steps.

QD algorithms, on the other hand, do not aim for fast convergence to a single, globally optimal solution, but instead they return a large and diverse set of high-quality solutions in a single run. In essence, they attempt to illuminate the fitness potential of various regions of a feature space, which is the space the user is interested in maintaining diversity, e.g., it could be the weight and height of an evolved robot morphology, while the fitness function could be the distance the robot travelled forward. MAP-Elites algorithm [12, 41] is one of the simplest QD algorithms as it discretizes the feature space into a grid and attempts to place an offspring into the corresponding bin if the bin is empty or if the offspring has better fitness than the solution that occupies the bin.

CVT-MAP-Elites [56] addresses MAP-Elites main drawback of not being able to use high-dimensional feature spaces. It does so by splitting the feature space into a number of homogeneous regions each being represented by its centroid point. This is typically done by uniformly sampling the feature space and using the $k$-means clustering algorithm [36] to find $k$ clusters. The algorithm works as before by identifying the "bin" of an offspring to be the one of its closest centroid.

## B UTILIZED MODELS

**Mathematical Formulation of VAEs.** VAEs are generative models that aim to approximate the unknown probability distribution $P(\boldsymbol{x})$ of input samples, such as images, $\boldsymbol{x} \in R^{d_x}, \boldsymbol{x} \sim P(\boldsymbol{x})$. They do so using an encoder-decoder architecture, similarly to classic autoencoders [47], with the aim of compressing (encoding) the samples to a much lower dimension $\boldsymbol{z} \in R^{d_z}, d_z << d_x$, and reconstructing (decoding) the samples as accurately as possible, as well as organizing the latent space in a regular way so that samples from it follow a Normal distribution.

More specifically, VAEs use a probabilistic encoder to compute $q_\phi(z|\boldsymbol{x}) = N(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\sigma}(\boldsymbol{x}))$, as an approximation of the posterior distribution $p(z|\boldsymbol{x})$, and a probabilistic decoder to compute the conditional likelihood distribution $p_\theta(\boldsymbol{x}|z)$. The mean $\boldsymbol{\mu}(\boldsymbol{x}) \in R^{d_z}$ and standard deviation $\boldsymbol{\sigma}(\boldsymbol{x}) \in R^{d_z}$ are computed through functions $f_\mu$ and $f_\sigma$ respectively, which typically have a shared component (a neural network), i.e., $\boldsymbol{\mu}(\boldsymbol{x}) = f_\mu(f_e(\boldsymbol{x}))$, $\boldsymbol{\sigma}(\boldsymbol{x}) = f_\sigma(f_e(\boldsymbol{x}))$.

The latent code $\boldsymbol{z}$ is sampled from the approximate posterior $\boldsymbol{z} \sim N(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\sigma}(\boldsymbol{x}))$ and then fed to the decoder function $f_d$ (typically a neural network) to reconstruct the input, i.e., $\hat{\boldsymbol{x}} = f_d(z), \hat{\boldsymbol{x}} \in R^{d_x}$. In order to make backpropagation feasible, that is, compute gradients through the aforementioned sampling operation, the reparameterization trick is used, i.e., $\boldsymbol{z} = \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}(\boldsymbol{x})$, where $\odot$ is the element-wise product, and $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{I}), \boldsymbol{\epsilon} \in R^{d_z}$ is an external input that injects noise in the latent space.

Given a datapoint $\boldsymbol{x}$, the loss function of the VAE is the following:

$$\begin{aligned} L_{\theta,\phi}(\boldsymbol{x}) &= -\mathbb{E}_{\boldsymbol{z} \sim q_\phi(z|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|z)] + D_{KL}(q_\phi(z|\boldsymbol{x})||p(z)) \\ &= ||\boldsymbol{x} - \hat{\boldsymbol{x}}||^2 + D_{KL}(N(\boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\sigma}(\boldsymbol{x}))||N(\boldsymbol{0}, \boldsymbol{I})) \end{aligned} \tag{1}$$

The first term is the reconstruction loss, or expected negative log-likelihood of $\boldsymbol{x}$. The second term is the Kullback-Leibler divergence between the encoder's distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z}) = N(\boldsymbol{0}, \boldsymbol{I})$; this term can be seen as a regularizer that forces the encoder outputs to become Normally distributed. For more details, see [17, 32, 33].

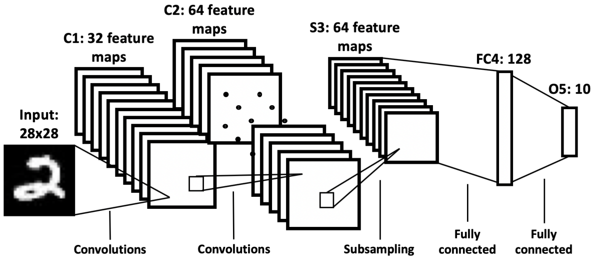
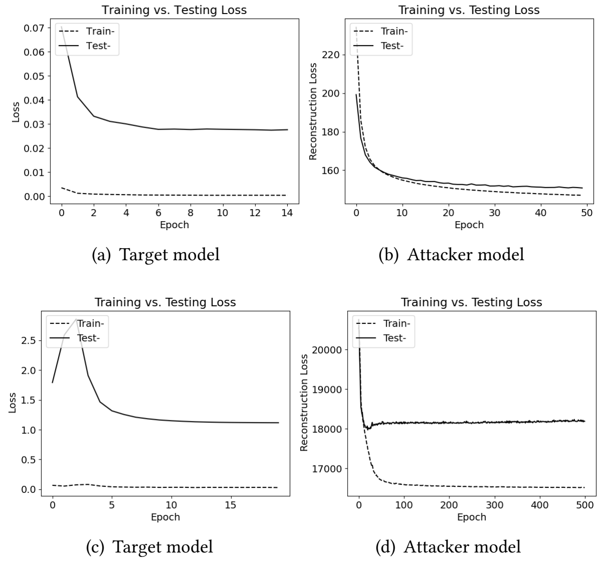


**Figure 13: The target CNN is composed of 5 layers: 2 convolutional layers (C1, C2), 1 sub-sampling layer (S3) and 2 fully-connected layers (FC4 and O5). C1 and C2 use 3 × 3 convolutions with stride 1 and the ReLU as an activation function. The S3 layer uses max-pooling with a 2 × 2 filter and stride 2. FC4 and O5 use ReLU and Log-Softmax as an activation function, respectively. Note that we use a similar, but deeper, CNN for the AT&T dataset (scenario 2).**

In this study, our focus is image data, therefore, we use a CVAE, meaning that the encoder ($f_e$) has a convolutional architecture and the decoder ($f_d$) has a deconvolutional one (see Fig. 15). For avoiding overfitting when training the CVAE we utilize dropout on the convolutional layers and early stopping. Note that the procedure for searching the CVAE's latent space operates at the inference stage so the concept of overfitting is irrelevant.

**Focusing on Well-generalizable Targets.** Demonstrating that the ML models used for our MI framework's evaluation are *well-generalizable* is of utmost importance. This is because: (a) *overfitted* ML models have been shown significantly more vulnerable to similar privacy attacks compared to well-generalizable ones [49, 54], and (b) facing *well-generalizable* targets is a common scenario; many sophisticated MLaaS platforms and ML experts exist that will maximize the *performance* and the *generalizability* of a privacy-sensitive ML model before deploying it in the wild. Thus, including the training–testing loss vs. epoch graphs for the target and attacker models, for scenarios 1 & 2, is important (see Fig. 14). As shown, for all the ML models, both the train and test losses keep decreasing, finally reaching a stable point. Those graphs, in combination with the train/test accuracies mentioned in Sec. 3, showcase that our models have been correctly trained on the respective datasets, achieving high performance and generalization.



(a) Target model (b) Attacker model

(c) Target model (d) Attacker model

**Figure 14: The target and attacker models' training–testing loss vs. epoch graphs for scenario 1 (a-b) and scenario 2 (c-d).**

## C DEEP-BMI'S BLACK-BOX OPTIMIZERS

**HC.** First, HC creates a random vector of size 1×20, which is the vector to be optimized – *current_point*. Then, the optimizer evaluates the randomly created vector by passing it through the decoder part of the CVAE. Afterwards, the algorithm creates a copy of the initial vector, *new_point*, selects a random feature, from the 20 features in the vector to be optimized, and adds a random noise factor derived from a Gaussian function, to the selected feature of *new_point* vector. Then, the algorithm evaluates the *new_point*. If the score of the *new_point* is larger than the score of the *current_point* then the algorithm saves the *current_point*, along with its score, and proceeds to the next iteration. Finally, the algorithm shows the image yielding the highest confidence score for the given *target_class*.

**Adaptive HC.** This is an adapted version of the previous optimizer. In particular, this optimizer has a sort of controlled acceleration by performing adaptive changes to the current solution. The overall operation is the same as the HC. However, for adaptive HC we also specify: (a) a *step_size*, which holds the amount of change to be made for each feature and has the same size as the vector to be optimized, (b) the *acceleration* factor, which controls the learning pace, i.e., the changes on each element of the vector to be optimized, and (c) the table *candidate*, which holds the available choices to be multiplied with the respective *step_size* and update a feature of the vector to be optimized. Overall, the algorithm tests all the available candidate changes and adapts the learning pace accordingly. The algorithm finishes as soon as it reaches the maximum number iterations *max_iterations* or the difference, in terms of performance, of two subsequent vector versions is smaller than the provided by user *epsilon* value.

**CMA-ES.** The core implementation for this optimizer has been taken from [22]. The objective function returns the confidence score of a given sample for the target class. Initially, the optimizer creates a random vector, *top_similar_encoding*, of size 1 × 20. Then, the script initializes the CMA-ES instance by giving the vector to be optimized, that is, *top_similar_encoding*, and the initial standard deviation, i.e., 0.5. Then, the optimizer starts the optimization process while also giving as argument the objective function, which determines the uncertainty score for each solution. Finally, CMA-ES reconstructs and shows the image of the solution that yields the target class with the highest confidence score by passing the optimized vector through the decoder part of the CVAE.

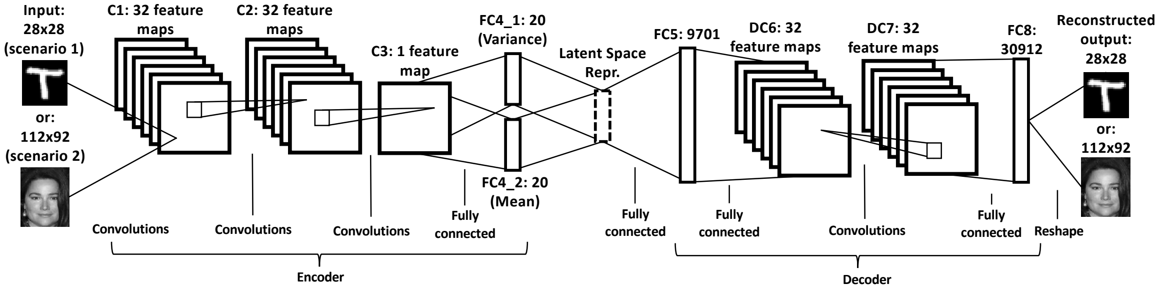


**Figure 15: The attacker's CVAE is composed of 8 layers:** 3 **convolutional layers (C1, C2, C3),** 2 **deconvolutional layers (DC6, DC7), and** 4 **fully-connected layers (FC4_1, FC4_2, FC5, FC8). C1, C2 and C3 use** $2 \times 2$ **convolutions with stride 1 and the ReLU as an activation function. DC6 and DC7 layers use** $2 \times 2$ **deconvolutions with stride 2 and the ReLU as an activation function. FC4_1, FC4_2, FC5 and FC8 use ReLU and Sigmoid as an activation function, respectively. Note that we also apply dropout on the convolutional layers for increasing the model's generalization.**

**MAP-Elites.** Optimization algorithms try to find the highest-performing solution in a search space, whereas *illumination* algorithms are meant to return the highest-performing solution at each point in the feature space [41]. Thus, they illuminate, hence the name, the fitness potential of each region of the feature space. MAP-Elites belongs to the *illumination* family of algorithms as it returns multiple solutions for each target class, while also maximizing their diversity. First, the optimizer creates a *map* of size $10 \times 20$ to host a solution for each target class and a $performances$ table of size $1 \times 10$ to store their respective performance. Then, for each iteration, the algorithm creates 2 new vectors, $child1$ and $child2$, using: (a) the crossover function described in [38], and (b) polynomial mutation, for scenario 1, or Gaussian mutation, for scenario 2. The mutation and crossover strategies are implemented by $random\_variation()$ function which gets the map of vectors as an argument. After that, the algorithm evaluates the two children and places them in the map if their performance is better than the performance of the current vectors. The algorithm finishes as soon as it reaches the maximum number iterations $max\_iterations$. Finally, the optimizer reconstructs and shows the images that yield each target class with the highest confidence score by passing each vector in the map through the decoder part of the CVAE.