NeuraCrypt: Hiding Private Health Data via Random Neural Networks for Public Training

Adam Yala†, Homa Esfahanizadeh†∗, Rafael G. L. D’Oliveira†, Ken R. Duffy‡,
Manya Ghobadi†, Tommi S. Jaakkola†, Vinod Vaikuntanathan†
Regina Barzilay†, Muriel Médard†

†Massachusetts Institute of Technology (MIT), 02139 USA
‡Maynooth University, Ireland

{adamyala, ghobadi, tommi, regina}@csail.mit.edu
{homaesf, rafaeld, vinodv, medard}@mit.edu
Ken.Duffy@mu.ie

Abstract

Balancing the needs of data privacy and predictive utility is a central challenge for machine learning in healthcare. In particular, privacy concerns have led to a dearth of public datasets, complicated the construction of multi-hospital cohorts and limited the utilization of external machine learning resources. To remedy this, new methods are required to enable data owners, such as hospitals, to share their datasets publicly, while preserving both patient privacy and modeling utility. We propose NeuraCrypt, a private encoding scheme based on random deep neural networks. NeuraCrypt encodes raw patient data using a randomly constructed neural network known only to the data-owner, and publishes both the encoded data and associated labels publicly. From a theoretical perspective, we demonstrate that sampling from a sufficiently rich family of encoding functions offers a well-defined and meaningful notion of privacy against a computationally unbounded adversary with full knowledge of the underlying data-distribution. We propose to approximate this family of encoding functions through random deep neural networks. Empirically, we demonstrate the robustness of our encoding to a suite of adversarial attacks and show that NeuraCrypt achieves competitive accuracy to non-private baselines on a variety of x-ray tasks. Moreover, we demonstrate that multiple hospitals, using independent private encoders, can collaborate to train improved x-ray models. Finally, we release a challenge dataset to encourage the development of new attacks on NeuraCrypt.

†Equal contribution
2github.com/yala/NeuraCrypt
3github.com/yala/NeuraCrypt-Challenge

Preprint. Under review.
1 Introduction

One of the central challenges of developing machine learning tools in healthcare is access to patient data. To protect patients’ privacy, regulations such as HIPPA [HIPAA] and GDPR [GDPR] greatly complicate creation of large, multi-institutional datasets, a necessary resource for training robust and equitable models. Consequently, the lack of public datasets has excluded the broad machine learning research community from contributing to clinical AI. Addressing this challenge is the focus of our paper. Consider the scenario where a hospital wishes to publicly release a dataset of mammograms with cancer labels. We are interested in developing a computationally effective mechanism for image encoding that both protects patient privacy (i.e., hiding the correspondence between patients and encoded samples), and facilitates learning.

There are a number of solutions that have been developed for different notions of privacy. While federated learning [McMahan et al., 2017] can offer privacy by training models in a distributed fashion, the framework relies on the coordination of data owners and model developers to run shared software. This framework is not designed to enable hospitals to deposit their datasets publicly. Cryptographic methods, such as secure multi-party computation, fully homomorphic encryption and functional encryption [Gentry 2009, Brakerski and Vaikuntanathan 2011, Boneh et al. 2012, Cho et al. 2018] could enable such public sharing and offer extremely strong security guarantees, hiding everything about the data. However, these security guarantees come at the cost of high computational overheads for use with modern methods. The notion of security adopted by these cryptographic methods is not suited to our setting, where image labels are public. Instead, we seek an efficient encoding scheme to protect the information not already implied by the image label.

We propose NeuraCrypt, a private encoding scheme designed to enable data owners to publish their data publicly while preserving both data privacy and modeling utility. NeuraCrypt encodes raw patient data using a randomly constructed neural network, known only to the data owner, and deposits both the encoded data and associated labels publicly for unknown third parties to develop models. When applied in the multi-institutional setting, each site utilizes independent private encoders to encrypt their data. With the help of label information, models trained in this setting can map these independently constructed encodings into a shared feature space. While the data remains private across the sites, each institution can still benefit from the larger combined dataset.

Our design is guided by theoretical considerations. First, we demonstrate a means, albeit an inefficient one, of offering for each sample anonymity amongst all other possible samples sharing the same label. Our approach relies on random selection of an encoder from a sufficiently rich family of encoding functions. Secondly, while we cannot construct the optimal family of encoding functions directly, i.e. all possible bijections, we demonstrate that we can iteratively enrich encoder families through function composition. These results, as well as recent work [Das et al. 2020] on the properties of random deep neural networks, motivate us to approximate the optimal family of encoding functions as deep neural networks. While our theoretical results do not guarantee the security of our specific NeuraCrypt network architecture, we provide empirical experiments to test the robustness of our approach against modern attacks, following standard practice in cryptanalysis [Standard 2001, Dworkin 2015].

We empirically test our method on two benchmark chest x-ray datasets MIMIC-CXR [Johnson et al. 2019] and CheXpert [Irvin et al. 2019], and compare it against the model operating on raw data. We demonstrate that, across a variety of diagnostic tasks, NeuraCrypt-based models achieve competitive performance. Moreover, we show that combining multiple datasets, using separate private encoders, enables the model to benefit from additional training data and thereby improve its accuracy. NeuraCrypt appears robust to adversarial attacks designed to either uncover the private encoder or to uncover additional sensitive attributes from the encoded data.

We believe our NeuraCrypt architecture, as well as the core idea of using random deep neural network encodings to achieve privacy, provide a novel direction for privacy-preserving machine learning. We encourage the development of new attacks on NeuraCrypt as well as further refinements of our method, and to this end, we release a challenge dataset.
2 Related Work

Cryptographic techniques. Cryptographic techniques, such as secure multiparty computation and fully homomorphic encryption \cite{Yao1986,Goldreich1987,Ben-Or1988,Chaum1988,Gentry2009,Brakerski2014,Cho2018}, allow one or more data owners, such as hospitals, to encode (encrypt) their data before providing them to a third party, say a cloud data center, for computation. Building models with homomorphic encryption requires leveraging specialized cryptographic primitives, an approach that requires an impractical overhead for modern deep learning models. In contrast, NeuraCrypt encodings can be directly leveraged by standard deep learning techniques.

The high complexity of constructing and running homomorphic encryption generally provide extremely strong guarantees, such as a semantic security \cite{Goldwasser1982}, wherein no information regarding the original data may be leaked by the encoding or computation. However, this strong security guarantee is an overkill for our context. For instance, we do not seek to hide the fact that a hospital hosts chest x-rays or the disease labels of those x-rays, as we release the labels publicly. Our goal is to avoid expending design and run-time resources \cite{Mohassel2017,Liu2017,Juvekar2018,Bourse2018} on hiding these facts through homomorphic encryption. Rather, we seek to provide privacy by protecting the characteristics of a chest x-ray not already captured by the disease label.

Federated learning and differential privacy techniques. Federated learning (FL) \cite{McMahan2017} enables collaborative learning among different data-owners (hospitals) through distributed training. The core idea of FL is to avoid transferring raw data by allocating an instance of the ML model at each data owner, and instead sharing model updates \cite{Rieke2020}. Despite a considerable amount of research in this area, including progress in secure aggregation \cite{Bonawitz2017} and differential privacy \cite{Dwork2014}, preserving privacy while maintaining modeling accuracy in FL remains an open challenge \cite{Schoon2021,Mcmahan2021}. In contrast to NeuraCrypt, FL is not designed to enable data-owners to publicly deposit their data.

Lightweight encoding techniques. Our approach is most closely related to prior focused on achieving privacy through lightweight encoding schemes. \cite{Ko2020,Tanaka2018,Srichotedumrong2019} have proposed de-identification techniques to carefully distort images to reduce their recognition rate by humans while preserving the accuracy of image classification models. Unfortunately, such methods do not offer privacy against realistic attacks. Recently, InstaHide \cite{Huang2020} proposed to encode images by linearly mixing them with other samples and applying a pixel-wise mask. While this approach provides collaborative learning for multiple data owners, its linear transforms preserve the relative distance between two samples in the original domain and the encoded domain. This drawback was exploited by \cite{Carlini2021} to decrypt the Instahide dataset challenge. DAUnTLeSS \cite{Xiao2021} proposed to encode images using fully-connected neural networks and linear sample mixing, and analyzed the computational difficulty of reversing this encoding to an attacker with access to parallel raw and encoded data pairs. They demonstrate that this encoding is easy to reverse if the source data distribution has low entropy (e.g. MNIST), and more difficult for complex datasets. Moreover, they demonstrate that mixing samples can increase the difficulty of the recovery task. In contrast to DAUnTLeSS, we consider a threat model where the adversary is computationally-unbounded and does not have access to parallel data.

3 Method

NeuraCrypt. The problem setting is depicted in Figure 1. We wish to enable a hospital to publish their image dataset \(X_d\) with diagnostic labels \(Y_d\) while protecting patient privacy. Given the dataset \(\{(x_i, y_i)\}_{x_i \in X_d}\), where \(y_i = L(x_i)\) is the label, a data-owner randomly samples a private NeuraCrypt encoder \(T_d\), a random neural network, and uses \(T_d\) to produce encoded samples \(Z_d = T_d(X_d)\), i.e., \(z_i = T_d(x_i)\) for every \(x_i \in X_d\). The data-owner can then deposit \(\{(T_d(x_i), L(x_i))\}_{x_i \in X_d}\) publicly for untrusted third parties to develop models to estimate \(\Pr[Y_d = Y_d|Z_d = Z_d]\). We note that multiple data owners can seamlessly collaborate to develop joint models by publishing datasets on the same task while using independent NeuraCrypt encoders. Given that model developers can only estimate \(\Pr[Y_d = Y_d|Z_d = Z_d]\) and not \(\Pr[Y_d = Y_d|X_d = X_d]\), only data-owners can directly
utilize the learned models, creating an incentive for data-owners and model developers to collaborate for model dissemination.

While many NeuraCrypt architectures are possible, we focus on medical imaging tasks, and thus implement our NeuraCrypt encoders as convolutional neural networks. Our encoder architecture is illustrated in Figure 2 and consists of convolutional layers with non-overlapping strides, batch normalization [Ioffe and Szegedy, 2015], and ReLU non-linearities. To encode positional information into the feature space while hiding spatial structure, we add a random positional embedding for each patch before the final convolutional and ReLU layers and randomly permute the patches at the output independently for each private sample. This results in an unordered set of patch feature vectors for each image. We note that this architecture is closely inspired by the design of patch-embedding modules in Vision Transformer networks [Dosovitskiy et al., 2020, Zhou et al., 2021].

Threat model. We assume a computationally unbounded adversary which knows all possible images $X$ as well as their labels, i.e., $\{(x, L(x))\}_{x \in X}$. The adversary also knows the distributions of both the data-owners' samples $Pr[X_d = X]$ and the choice of encoder function $Pr[T_d = T]$. This adversary knows $\{(T_d(x), L(x))\}_{x \in X_d}$, as the data owner deposits it publicly. Moreover, since we do not know the adversary’s classifying capabilities, we assume the worst case, in which she is able to perfectly classify the encoded samples. Thus, we assume the adversary also knows $\{(T_d(x), L(x))\}_{x \in X}$, where now the images are taken over the whole of $X$. The goal of the adversary is to learn more about the random variable $X_d$ using $Z_d, Y_d$ than from $Y_d$ alone. We define this formally in section 4.

Privacy intuition. We provide a formal privacy analysis for the described threat model given an encoder space $F$ in Section 4. We consider a theoretical $F$ which consists of all bijections from $X$ to $X$ with the same label assignment. These encodings map each image of the hospital’s dataset to another image in $X$ with the same label. This function family is exponentially large in size of $X$ and a computationally unbounded adversary, as described in our threat model, cannot distinguish between its members. In this setting, observing $(Z_d, Y_d)$ does not offer the adversary more information about the underlying $X_d$ than observing $Y_d$. While sampling from this optimal family directly is not feasible, as it would require full knowledge of $X$ and its labels, our theoretical analysis demonstrates that we can enrich the privacy of our encoding scheme via functional composition. This result motivates us to approximate the $F$ using deep neural networks.
4 Privacy Analysis

In this section, we present a detailed analysis of the threat model in Section 3. All proofs appear in the appendix. We denote the set of all samples by $\mathcal{X}$ and assume it is a finite set. Each sample $x \in \mathcal{X}$ is labeled by a function $L : \mathcal{X} \rightarrow \mathcal{Y}$, where the set of labels $\mathcal{Y}$ is also finite. We denote the set of all bijections from $\mathcal{X}$ to itself by $\text{Sym}(\mathcal{X}) = \{ T : \mathcal{X} \rightarrow \mathcal{X} : T \text{ is a bijection} \}$. As each data-owner acts independently, we perform our analysis individually. We consider a model with three participants, Alice, Bob and Eve, to be consistent with the common terminology of privacy (see also Figure 1):

Alice (the data owner). Alice has a private subset of samples $X_A \subseteq \mathcal{X}$ which is drawn from a distribution $\Pr[X_A = X_A]$. The subscript $A$ for Alice replaces the generic subscript $d$ used for $X$ in our Method discussion. Alice samples an encoder $T_A$ from a family $\mathcal{F} \subseteq \text{Sym}(\mathcal{X})$ according to a distribution $\Pr[T_A = T]$, and then transmits $M_A(X_A) = \{(T_A(x), L(x))\}_{x \in X_A}$ to Bob. Alice does not know or have control over $\Pr[X_A = X_A]$ but gets to choose $\mathcal{F}$ and the distribution $\Pr[T_A = T]$.

Bob (the model developer). Bob, who is not aware of the true labeling $L \in \mathcal{Y}^X$, knows a prior distribution $\Pr[L = L']$, and also the distribution $\Pr[T_A = T]$. After receiving $M_A(X_A) = \{(T_A(x), L(x))\}_{x \in X_A}$, he is interested in learning a classifier on Alice’s encoded data, $L_A = L \circ T_A^{-1}$, i.e., the distribution $\Pr[L_A = L_A | M_A(X_A) = M_A(X_A)]$, for every $x \in \mathcal{X}$.

Eve (the adversary). Eve knows $\{(x, L(x))\}_{x \in \mathcal{X}}$ and the distributions $\Pr[X_A = X_A]$ and $\Pr[T_A = T]$. Since Alice (e.g. a hospital) releases the data publicly, Eve also knows $M_A(X_A) = \{(T_A(x), L(x))\}_{x \in X_A}$. While impractical, we consider a worst case scenario where Eve is able to classify perfectly any sample and thus, we also assume that she knows $M_A(X_A) = \{(T_A(x), L(x))\}_{x \in \mathcal{X}}$. Eve is interested in learning the random variable $X_A$.

In our model we are interested in comparing what Eve learns of Alice’s samples $X_A$ from the observation of $M_A(X_A)$ and $M_A(X_A)$ compared to only having observed $Y_A = \{L(x)\}_{x \in X_A}$. In this context, we say Alice’s scheme is perfectly private if $\Pr[X_A = X_A | M_A(X_A) = M_A(X_A)], M_A(X_A) = M_A(X_A) = M_A(X_A) = \Pr[X_A = X_A | Y_A = Y_A]$. Since Alice can only choose $\mathcal{F}$ and its distribution $\Pr[T_A = T]$, we often refer to $\mathcal{F}$ as Alice’s scheme.

For notation purposes, we occasionally impose a total order $\preceq$ on $\mathcal{X}$ to represent it by a vector $(x_1, \ldots, x_{|\mathcal{X}|})$ such that $x_i \preceq x_j$ if $i \leq j$. We then represent a transformation $T$ by a vector with size $|\mathcal{X}|$ whose $i$-th element is $T(x_i)$.

Next we define the label configuration (LC) and the LC-anonymity list, as these notions capture Eve’s uncertainty about Alice’s private samples $X_A$.

Definition 1 (Label configuration). The original label configuration, denoted by $\text{LC}(\mathcal{X})$, is a vector of size $|\mathcal{X}|$ whose $i$-th elements is $L(x_i)$. The label configuration of an encoder $T$, denoted by $\text{LC}(T)$, is a permutation of $\text{LC}(\mathcal{X})$ according to $T$, i.e. the $i$-th element of $\text{LC}(T)$ is $L(T^{-1}(x))$.

Definition 2 (LC-anonymity list). Given a family of encoders $\mathcal{F}$, the LC-anonymity list of an encoder $T \in \mathcal{F}$ is defined as $\mathcal{F}_T = \{ T' \in \mathcal{F} : \text{LC}(T') = \text{LC}(T) \}$.

The LC-anonymity lists partition $\mathcal{F}$ into different equivalence classes (see Figure 3). We also note that under these new concepts, Eve’s knowledge of $M_A(X_A) = \{(T_A(x), L(x))\}_{x \in X_A}$ and $M_A(X_A) = \{(T_A(x), L(x))\}_{x \in \mathcal{X}}$ is equivalent to her knowing $T_A(X_A)$ and $\text{LC}(T_A)$.

Example 1. Let $\mathcal{X} = \{1, \ldots, 16\}$ and, for representation purposes, consider the ordering $1 \preceq 2 \preceq \ldots \preceq 16$. Let $\mathcal{Y} = \{-, +\}$, and $\mathcal{F} = \text{Sym}(\mathcal{X})$. Let the labeling $L$ be such that $\text{LC}(\mathcal{X}) = (++---+++---+++---+++)--$. Assume that Alice chooses $T_A = (12, 2, 11, 4, 6, 8, 16, 15, 13, 7, 9, 5, 3, 14, 1, 10)$, and that her data is $X_A = \{2, 5, 7, 10, 15\}$. Thus, she sends $\{(2, +), (6, +), (16, -), (7, -), (1, -)\}$ to Bob. According to the threat model, Eve knows $\text{LC}(T_A) = (-----++-----++-----++----++---++-----+++--)$. Although $|\mathcal{F}| = 16! = 2.13e13$, since Eve knows $\text{LC}(T_A)$, she can infer that Alice’s encoder is in the smaller set $\mathcal{F}_{T_A}$, with cardinality $4|\mathcal{X}|!2! = 1.1e10$. Thus, if Alice had chosen $T_A$ uniformly at random from $\mathcal{F}$, then the probability of Eve guessing $T_A$ correctly would be approximately $1/e-10$. Eve, however, is interested in learning $X_A$. In this case, since $T_A$ is uniform over all permutations, it can be shown that $\mathcal{F}$ is perfectly private, i.e. Eve learns no more than she would by solely observing $Y_A = \{L(x)\}_{x \in X_A}$.

As illustrated in the example above, the scheme where Alice chooses $T_A$ uniformly from $\text{Sym}(\mathcal{X})$ is perfectly private. This scheme, however, completely scrambles of the dataset, making Bob’s learning
We now describe a scheme which achieves perfect privacy without affecting learnability. We begin by characterizing the set of possible values for $X$ which are perfectly private, while still preserving the structure of Theorem 1. The following example showcases the theorem. The version of this example is given in the appendix.

Let $X_A$ be the random variable for Alice’s encoder. We note that, any privacy metric, be it a mutual information [Shannon, 1948], a mean squared error, or a guessing measure [Pliam, 1999], can be computed from it. Since Theorem 1 fully characterizes how much Eve learns about Alice’s private samples and the encoder $T_A$, any privacy task impossible. By observing a subset of the encoded data, Bob still cannot predict the label of a held-out sample better than the prior probability. In Theorem 2 we show that there exist schemes which are perfectly private, while still preserving the structure of $X_A$. Towards that, we begin by investigating the privacy of general families $\mathcal{F} \subseteq \text{Sym}(\mathcal{X})$.

We begin by characterizing the set of possible values for $X_A$, given Eve’s observations.

**Proposition 1.** The set of possible values for Alice’s dataset, given Eve’s observations, is

$$\text{Pos}(X_A) \triangleq \{ \bar{x} \in \mathcal{X} : \exists T \in \mathcal{F}_{T_A} \text{ with } T(\bar{x}) = T_A(X_A) \}.$$  

The next theorem characterizes what Eve learns about $X_A$ via Alice’s scheme and what she would learn if, instead, she only observed $Y_A = \{L(x)\}_{x \in X_A}$.

**Theorem 1.** Let $X_A$ be the random variable for Alice’s private samples and $T_A$ be the random variable for Alice’s encoder. We note $I(\text{condition})$ is an indicator function which takes value one when condition happens and zero otherwise. Then,

$$\Pr[X_A = X_A | T_A(X_A) = Z, \mathcal{L}C(T_A) = C] \propto \sum_{T: \mathcal{L}C(T) = C, T(X_A) = Z} \Pr[X_A = X_A] \Pr[T_A = T]$$

and

$$\Pr[X_A = X_A | Y_A = Y_A] \propto I(Y_A = \{L(x)\}_{x \in X_A}) \cdot \Pr[X_A = X_A].$$

Since Theorem 1 fully characterizes how much Eve learns about Alice’s private samples and the encoder $T_A$, any privacy metric, be it a mutual information [Shannon, 1948], a mean squared error, or a guessing measure [Pliam, 1999], can be computed from it.

The following example showcases the theorem.

**Example 2.** Let $\mathcal{X} = \{1, 2, 3, 4, 5\}$, $X_A = \{2, 3, 4\}$, $\mathcal{L}C(\mathcal{X}) = (+ + + -)$, and $\mathcal{F} = \{T_1, \ldots, T_6\}$ given below. Suppose $T_A = T_1$. Then, Alice transmits $\{(1, +), (4, -), (5, -)\}$. Eve knows $\mathcal{L}C(T_A) = (+ + + -)$, and thus knows $\mathcal{F}_{T_A}$ as well:

$$\mathcal{F} = \begin{cases} T_1 : (2, 1, 5, 4, 3) \\ T_2 : (2, 1, 3, 5, 4) \\ T_3 : (1, 2, 3, 5, 4) \\ T_4 : (4, 3, 1, 2, 5) \\ T_5 : (3, 4, 2, 1, 5) \\ T_6 : (5, 2, 4, 3, 1) \end{cases}, \quad \mathcal{F}_{T_A} = \begin{cases} T_1 : (2, 1, 5, 4, 3) \\ T_2 : (2, 1, 3, 5, 4) \\ T_3 : (1, 2, 3, 5, 4) \end{cases}.$$  

As a result, $\text{Pos}(X_A) = \{(2, 4, 3), (2, 5, 4), (1, 5, 4)\}$. Thus, Eve knows that $4 \in X_A$ independent of the distribution of $X_A$. If we assume uniform distributions for $T_A$ and $X_A$, then, by Theorem 1, we obtain a uniform distribution on $\text{Pos}(X_A)$ and can compute $\Pr[1 \in X_A] = 1/3$ and $\Pr[2 \in X_A] = 2/3$. If instead we assume uniform $T_A$ and $\Pr[X_A = \{2, 3, 4\}] = 0.1$, $\Pr[X_A = \{2, 5, 4\}] = 0.1$, and $\Pr[X_A = \{1, 5, 4\}] = 0.4$, then $\Pr[1 \in X_A] = 2/3$ and $\Pr[2 \in X_A] = 1/3$. A more detailed version of this example is given in the appendix.

We now describe a scheme which achieves perfect privacy without affecting learnability.
While we cannot directly sample from the optimal family of encoders, we can leverage our theoretical results on function composition to guide the design of NeuraCrypt. Starting with a weak encoder, i.e. a linear layer, we iteratively enrich the privacy of our function family through function composition (e.g. by adding with additional non-linear and linear layers), to build a random neural network. Moreover, given that we know that the labeling function $L_A$ on medical images is likely to be efficiently estimated with a convolutional neural network, we implement our NeuraCrypt encoders as convolutional neural networks. We emphasize that our privacy guarantees for the optimal family do not extend to our implementation of NeuraCrypt, and thus its privacy must be tested empirically, following standard practice in cryptanalysis \cite{standard2001,dworkin2015}.

5 Experiments

Datasets. For all experiments, we utilized two benchmark datasets of chest x-rays, MIMIC-CXR \cite{johnson2019} and CheXpert \cite{irvin2019} from Beth Israel Deaconess Medical Center and Stanford respectively. For each dataset, we evaluated the ability of each model to predict Edema, Pneumothorax, Consolidation, Cardiomegaly and Atelectasis. For each task, we excluded exams with an uncertain disease label, i.e., the clinical diagnosis did not explicitly rule out or confirm the disease, and randomly split the remaining data 60–20–20 for training, development and testing respectively.
We performed a grid search over different learning rates and weight decay values for each attack. where $T$ NeuraCrypt we performed the attack on we found this difficult to train. Instead, we conducted experiments using Maximum Mean Discrepancy we performed experiments on the combined MIMIC-CXR and CheXpert datasets with Cardiomegaly Adversarial attack. To validate robustness of our encoding approach to attacks aimed at estimating $F$ the exact architecture of NeuraCrypt, but not the weights of $T$ which are private. Given this information, the attacker tries to learn a $T^*$ such that $T(X) \approx T^*(X)$. We assume that if $T$ was leaked, the attacker could easily invert the encoding and recover the raw original images. To estimate $T$, we sampled an initial $T^*$ with the same architecture as $T$, and trained it to minimize the accuracy of a domain discriminator, which aims to distinguish between the generated $Z^*$ and true $Z$.

Our NeuraCrypt encoding leveraged a patch-size of $16 \times 16$, a depth of 7, and a hidden dimension of 2048. This model had $\sim 22.9M$ parameters and mapped $256 \times 256$ pixel images to $256 \times 2048$ vectors. Due to the patch-shuffling step in NeuraCrypt, this representation is unordered. As a result, we trained Vision Transformers (ViT) [Zhou et al., 2021], a self-attention based architecture that is invariant to patch ordering. Across all experiments, we used a one-layer ViT with a hidden dimension of 2048. We compared NeuraCrypt model performance with a non-private baseline, namely an identical ViT model where the patch encoder is learned jointly. We trained all models for 25 epochs using the Adam optimizer [Kingma and Ba, 2014], an initial learning rate of $1e-04$, weight decay of $1e-03$ and a batch size of 128.

5.1 Evaluating modeling utility

To evaluate the impact of NeuraCrypt encodings on downstream modeling performance, we compared NeuraCrypt-based models to standard architectures trained on raw images across both the single and multi-hospital setting. For each diagnosis task and training setting, we report the average AUC across the MIMIC-CXR and CheXpert test sets. We report results on the MIMIC-CXR and CheXpert datasets individually in the appendix. For NeuraCrypt multi-hospital training, we wished to evaluate the impact of leveraging independent NeuraCrypt encoders on modeling accuracy. As a result, we evaluate both model performance when leveraging a single encoder across both hospitals (Combined-Clear), and model performance when leveraging two independent encoders (Combined-Private). We note that performance in the Combined-Clear setting acts as an upper-bound for Combined-Private.

5.2 Evaluating robustness to attacks

Adversarial attack. To validate robustness of our encoding approach to attacks aimed at estimating $T$, we preformed experiments on the combined MIMIC-CXR and CheXpert datasets with Cardiomegaly labels. We assumed that the attacker has access to the entire labeled dataset $\{(x_i, y_i)\}_{x_i \in X}$, and labeled NeuraCrypt-encoded samples $\{(z_i = T(x_i), y_i)\}_{x_i \in X}$. We also assumed that the attacker knows $F$ (the exact architecture of NeuraCrypt), but not the weights of $T$ which are private. Given this information, the attacker tries to learn a $T^*$ such that $T(X) \approx T^*(X)$. We assume that if $T$ was leaked, the attacker could easily invert the encoding and recover the raw original images. To estimate $T$, we sampled an initial $T^*$ with the same architecture as $T$, and trained it to minimize the accuracy of a domain discriminator, which aims to distinguish between the generated $Z^*$ and true $Z$.

The discriminator is typically designed as a parameterized classifier [Tzeng et al., 2017, Goodfellow et al., 2014, Shen et al., 2017], and trained through a mini-max game with the encoder $T^*$. However, we found this difficult to train. Instead, we conducted experiments using Maximum Mean Discrepancy (MMD) [Gretton et al., 2012] as our discriminator. Specifically, we defined the MMD loss as for a batch of real ciphertext and generated ciphertext $Z$ and $Z^*$ as $L_{\text{MMD}} = \text{MMD}^2(Z, Z^*)$, where $\text{MMD}^2(Z, Z^*) = \|\sum_z \phi(z)/|Z| - \sum_z \phi(z^*)/|Z^*\|_2$ measures the discrepancy between $Z$ and $Z^*$ on a Reproducing Kernel Hilbert Space, and $\phi$ is a feature map induced by a linear combination of multiple RBF kernels $\kappa(z_i, z_j) = \sum_{\theta} \exp(-1/(2\sigma_\theta \|z_i - z_j\|^2))$. Our MMD formulation follows prior work in domain adaptation [Guo et al., 2018, Bousmalis et al., 2016].

In order to understand how the success of this attack would vary with the architecture of NeuraCrypt, we performed the attack on NeuraCrypt architectures with a depth of 2 and 7. As a baseline, we also performed the attack when using a simple linear encoder, implemented as single convolutional layer. Across all experiments, we used a hidden dimension of 2048 and trained $T^*$ for 25 epochs. We performed a grid search over different learning rates and weight decay values for each attack. For each experiment, we evaluated the attack by measuring the mean squared error (MSE) between generated ($Z^*$) and real ciphertext ($Z$) for the same plaintext images across the dataset. To understand if $T^*$ outperforms a trivial baseline, we compare the performance of $T^*$ to the mean baseline $T_{\mu}$, where $T_{\mu}(x) = \frac{1}{|Z|} \sum_z z$. $T_{\mu}$ ignores the input $x$ and predicts the mean of $Z$ for all inputs. We consider an attack to be successful if $T^*$ outperforms $T_{\mu}$, i.e., the ratio of $\frac{\text{MSE}_{T_{\mu}}}{\text{MSE}_{T^*}} < 1$. We consider additional experiments varying the architecture of $T$ and $T^*$ and provide additional analyses in the appendix.
Transfer learning attack. We also consider a scenario where an attacker may try to learn a sensitive attribute classifier, such as a gender predictor, on chest x-rays and tries to transfer this classifier onto the encoded data $Z$. If this transferred classifier performs better than random on $Z$, then an attacker could leverage this approach to learn sensitive information not released by the data-owner or to propagate labels to be used in a refined adversarial attack. To validate the robustness of NeuraCrypt to this type of attack, we began with the best estimated $T^*$ from our adversarial attack experiments, as measured by $T^*$ MSE, and built a new classifier to predict Edema given $Z^*$. We report the ROC AUC of this classifier on both $Z$ and $Z^*$. We performed these experiments on the combined MIMIC-CXR and CheXpert dataset. We performed this attack for both NeuraCrypt-encoding with a depth of 7 and a hidden dimension of 2048, as leveraged in the modeling utility experiments, and when using a linear encoding. For each experiment, we trained a ViT for 25 epochs using the Adam optimizer, an initial learning rate of 1e−04 and a batch size of 128.

6 Results

Table 1: Impact of NeuraCrypt on chest x-ray prediction tasks across different training settings. All metrics are average ROC AUCs across the MIMIC-CXR and CheXpert test sets. Combined-Clear and Combined-Private refer to using a single NeuraCrypt encoder across the combined MIMIC-CXR and CheXpert datasets and two independent NeuraCrypt encoders respectively. Guides of abbreviations for medical diagnosis: (E)dema, (P)neumothorax, (Co)nsolidation, (Ca)rdiomegaly and (A)telectasis.

| Model          | E  | P  | Co | Ca | A  | Average |
|----------------|----|----|----|----|----|---------|
| Train on MIMIC-CXR |    |    |    |    |    |         |
| ViT            | 85 ± 1 | 69 ± 3 | 74 ± 2 | 87 ± 0 | 85 ± 1 | 80      |
| NeuraCrypt-ViT  | 85 ± 2 | 72 ± 1 | 72 ± 1 | 87 ± 0 | 83 ± 1 | 80      |
| Train on CheXpert |    |    |    |    |    |         |
| ViT            | 82 ± 1 | 71 ± 1 | 72 ± 3 | 83 ± 1 | 80 ± 0 | 77      |
| NeuraCrypt-ViT  | 84 ± 1 | 71 ± 1 | 75 ± 2 | 82 ± 1 | 81 ± 0 | 79      |
| Train on Combined-Clear |    |    |    |    |    |         |
| ViT            | 86 ± 0 | 77 ± 1 | 76 ± 2 | 87 ± 1 | 85 ± 0 | 82      |
| NeuraCrypt-ViT  | 87 ± 0 | 76 ± 3 | 78 ± 1 | 88 ± 0 | 85 ± 1 | 83      |
| Train on Combined-Private |    |    |    |    |    |         |
| NeuraCrypt-ViT  | 87 ± 1 | 77 ± 3 | 77 ± 3 | 86 ± 1 | 84 ± 1 | 82      |

Evaluating modeling utility. We report our results in predicting various medical diagnoses from chest x-ray datasets in Table 1. NeuraCrypt-ViT obtained competitive AUCs to our non-private ViT baseline across all training settings. In the multi-hospital setting, we found that NeuraCrypt-ViT was effectively able to leverage the larger training set to learn an improved classifier, despite using separate encoders for each dataset. NeuraCrypt-ViT obtained an average AUC increase of 2 and 3 points compared to training only on the MIMIC-CXR and CheXpert datasets respectively. Moreover, NeuraCrypt demonstrated achieved equivalent performance in the Combined-Clear and Combined-Private settings, demonstrating that multiple institutions do not pay a significant performance cost to collaborate privately.

Table 2: Left: MSE of MMD-based adversarial attacks on different NeuraCrypt encodings. $T_\mu$ refers to the mean baseline, and $T^*$ refers to the encoder learned via the MMD attack. Right: Performance of transfer learning attack on Linear and NeuraCrypt encodings.

| Encoding          | $T^*/T_\mu$ MSE | $T^*$ MSE/Linear |
|-------------------|-----------------|-----------------|
| Linear            | 3.02 ± 0.02     | 0.43 ± 0.01     |
| NeuraCrypt-depth-2| 7.97 ± 0.28     | 2.44 ± 0.12     |
| NeuraCrypt-depth-7| 4.44 ± 0.12     |
| NeuraCrypt        | 84 ± 1          | 52 ± 4          |

Evaluating robustness to attacks. We report the performance of our adversarial and transfer learning attacks in Table 2, left and right panels respectively. As expected, using a linear encoding is not robust to either adversarial or transfer learning based attacks. We found that the adversarial attack outperformed the mean baseline (i.e. $\frac{T^*_{\text{MSE}}}{T_{\mu_{\text{MSE}}}} < 1$), and an Edema classifier trained on $T^*$ transferred well to $T$ encodings. In contrast, our adversarial attack on NeuraCrypt failed to outperform the mean...
baseline when both using 2 and 7 layers. Moreover, NeuraCrypt was robust to our transfer learning attack, with the Edema classifier failing to obtain a AUC significantly better than random on the true T encodings.

7 Conclusion

We proposed NeuraCrypt, a private encoding scheme based on random neural networks designed to enable data owners to publicly publish their datasets while retaining data privacy and modeling utility. On two benchmark chest x-ray datasets, MIMIC-CXR and CheXpert, we found that NeuraCrypt-models obtained competitive performance to our non-private baselines. In the multi-institutional setting, where each site leverages an independent name encoder, we demonstrated that NeuraCrypt-models could effectively leverage the larger training data to learn improved classifiers. While this paper focused on medical imaging and chest X-ray tasks, NeuraCrypt can easily be extended to new data modalities such as text or molecular graphs. While we are not able to sample from the optimal family of encoders identified in our theoretical analysis, our analysis also provided a useful guide (i.e. function composition) for the design of NeuraCrypt. Similar to prior work in cryptanalysis [Standard, 2001] [Dworkin, 2015], we note that our empirical results on adversarial robustness are not sufficient to prove the privacy of our architecture family for NeuraCrypt. Improved algorithms for domain adaptation [Tzeng et al., 2017] [Guo et al., 2018] or unsupervised translation [Lample et al., 2017] [Alvarez-Melis and Jaakkola, 2018] specialized to the design of NeuraCrypt may yield more successful attacks. We release a challenge dataset to both encourage the development of new attacks on NeuraCrypt as well as the development of improved NeuraCrypt architectures.

References

David Alvarez-Melis and Tommi S Jaakkola. Gromov-wasserstein alignment of word embedding spaces. arXiv preprint arXiv:1809.00013, 2018. pages 10

Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In Janos Simon, editor, Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA, pages 1–10. ACM, 1988. pages 3

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 1175–1191, 2017. pages 3

Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. Commun. ACM, 55(11):56–64, 2012. doi: 10.1145/2366316.2366333. URL https://doi.org/10.1145/2366316.2366333 pages 2

Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier. Fast homomorphic evaluation of deep discretized neural networks. In Advances in Cryptology, volume 10993, pages 483–512. Springer, 2018. pages 3

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. arXiv preprint arXiv:1608.06019, 2016. pages 8

Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011, pages 97–106. IEEE Computer Society, 2011. doi: 10.1109/FOCS.2011.12. URL https://doi.org/10.1109/FOCS.2011.12 pages 2

Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. SIAM J. Comput., 43(2):831–871, 2014. doi: 10.1137/120868669. URL https://doi.org/10.1137/120868669 pages 3

Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Shuang Song, Abhradeep Thakurta, and Florian Tramer. Is private learning possible with instance encoding?, 2021. pages 3
David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 11–19. ACM, 1988.

Hyunghoon Cho, David J Wu, and Bonnie Berger. Secure genome-wide association analysis using multiparty computation. *Nature biotechnology*, 36(6):547–551, 2018.

Abhimanyu Das, Sreenivas Gollapudi, Ravi Kumar, and Rina Panigrahy. On the learnability of random deep networks. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 398–410. SIAM, 2020. doi: 10.1137/1.9781611975994.24. URL https://doi.org/10.1137/1.9781611975994.24.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Morris J Dworkin. SHA-3 standard: Permutation-based hash and extendable-output functions. 2015.

Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178. ACM, 2009. doi: 10.1145/1536414.1536440. URL https://doi.org/10.1145/1536414.1536440.

Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *Proceedings of the 14th Annual ACM Symposium on Theory of Computing, May 5-7, 1982, San Francisco, California, USA*, pages 365–377. ACM, 1982.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Jiang Guo, Darsh J Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. *arXiv preprint arXiv:1809.02256*, 2018.

HIPAA. *Health Insurance Portability and Accountability Act of 1996*.

Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. InstaHide: Instance-hiding schemes for private distributed learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research*, pages 4507–4518. PMLR, 13–18 Jul 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilicus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 590–597, 2019. pages 2, 7

Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. arXiv preprint arXiv:1901.07042, 2019. pages 2, 7

Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. Gazelle: A low latency framework for secure neural network inference. In Proceedings of the 27th USENIX Conference on Security Symposium, SEC’18, page 1651–1668, USA, 2018. USENIX Association. ISBN 9781931971461. pages 3

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. pages 8

D. Ko, S. Choi, J. Shin, P. Liu, and Y. Choi. Structural image De-Identification for privacy-Preserving deep learning. IEEE Access, 8:119848–119862, 2020. pages 3

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. arXiv preprint arXiv:1711.00043, 2017. pages 10

Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious neural network predictions via mimonn transformations. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS ’17, page 619–631, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3134056. URL https://doi.org/10.1145/3133956.3134056. pages 3

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics, pages 1273–1282. PMLR, 2017. pages 2, 3

H Brendan McMahan et al. Advances and open problems in federated learning. Foundations and Trends® in Machine Learning, 14(1), 2021. pages 3

Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In 2017 IEEE Conference on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017, pages 19–38. IEEE Computer Society, 2017. doi: 10.1109/SP.2017.12. URL https://doi.org/10.1109/SP.2017.12. pages 3

John O Pliam. Guesswork and variation distance as measures of cipher security. In International Workshop on Selected Areas in Cryptography, pages 62–77. Springer, 1999. pages 6

Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R. Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N. Galtier, Bennett A. Landman, Klaus Maier-Hein, Sébastien Ourselin, Micah Shellcr, Ronald M. Summers, Andrew Trask, Daguang Xu, Maximilian Baust, and M. Jorge Cardoso. The future of digital health with federated learning. npj Digital Medicine, 3(1):119, 2020. URL https://doi.org/10.1038/s41746-020-00323-1. pages 3

Ben Schoon. Now EU regulators are worried about Google’s FLoC initiative, too (last accessed May 2021), 2021. URL https://9to5google.com/2021/04/30/google-floc-eu-regulators/. pages 3

Claude E. Shannon. A mathematical theory of communication. Bell Syst. Tech. J., 27(3):379–423, 1948. pages 6

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. arXiv preprint arXiv:1705.09635, 2017. pages 8
A Broader impact

Our study proposes a method to enable data-owners, e.g., hospitals, to share their data publicly while protecting both patient privacy and modeling utility. We hope that this technology will enable the construction of diverse multi-center patient cohorts, and allow the broader machine learning community to contribute to the development of healthcare algorithms. We believe that improved algorithms in this space will lead to more equitable and precise healthcare. However, we acknowledge that the same privacy-preserving technology could be used to accelerate unethical biomedical research.

B Proofs of theoretical analysis

In this section, we prove the results of Section 4 and give an extended version of Example 2. We begin by proving the following lemma.

Lemma 1. Let $X_A$ be the random variable for Alice’s private samples and $T_A$ be the random variable for Alice’s encoder. Then, $Pr[X_A = X_A, T_A = T | T_A(X_A) = Z, LC(T_A) = C]$ is proportional to $1(T(X_A) = Z) \cdot 1(LC(T) = C) \cdot Pr[X_A = X_A] \cdot Pr[T_A = T]$.

Proof of Lemma[1] Let $P_1 = Pr[X_A = X_A, T_A = T | T_A(X_A) = Z, LC(T_A) = C]$ be the conditional joint distribution of $X_A$ and $T_A$. $P_2 = Pr[X_A = X_A | T_A = T, T_A(X_A) = Z, LC(T_A) = C]$ and $P_3 = Pr[T_A = T | T_A(X_A) = Z, LC(T_A) = C]$. Then, $P_1 = P_2 P_3$, and a direct calculation shows that $P_2 = 1(T(X_A) = Z)$. As for $P_3$, it follows from Bayes’ theorem that

$$P_3 \propto Pr[T_A(X_A) = Z | T_A = T, LC(T_A) = C] \cdot Pr[T_A = T, LC(T_A) = C].$$

(1)

Next, we have $Pr[T_A(X_A) = Z | T_A = T, LC(T_A) = C] = Pr[X_A = T^{-1}(Z)]$. Also, by Bayes’ theorem, $Pr[T_A = T, LC(T_A) = C] \propto Pr[LC(T_A) = C] = Pr[T_A = T] \cdot Pr[T_A = T]$. Finally, from a direct calculation we obtain $Pr[LC(T_A) = C, T_A = T] = 1(LC(T) = C).$ The result follows from substituting everything into (1) and multiplying by $P_2$. We note that $1(T(X_A) = Z) \cdot Pr[X_A = T^{-1}(Z)] = 1(T(X_A) = Z) \cdot Pr[X_A = X_A]$.

\[\square\]
Theorem 1. Let $X_A$ be the random variable for Alice’s private samples and $T_A$ be the random variable for Alice’s encoder. Then,
\[
\Pr[X_A = X_A \mid T_A(X_A) = Z, \text{LC}(T_A) = C] \propto \sum_{\{T : \text{LC}(T) = C, T(X_A) = Z\}} \Pr[X_A = X_A] \Pr[T_A = T]
\]
and
\[
\Pr[X_A = X_A \mid Y_A = Y_A] \propto \mathbb{I}(Y_A = \{L(x)\}_{x \in X_A}) \cdot \Pr[X_A = X_A].
\]

Proof of Theorem 1. For the first statement,
\[
P_A = \sum_T \Pr[X_A = X_A, T_A = T \mid T_A(X_A) = Z, \text{LC}(T_A) = C] \tag{2}
\]
\[
\propto \sum_T \mathbb{I}(T(X_A) = Z) \cdot \mathbb{I}(\text{LC}(T) = C) \cdot \Pr[X_A = X_A] \cdot \Pr[T_A = T] \tag{3}
\]
\[
= \sum_{\{T : \text{LC}(T) = C, T(X_A) = Z\}} \Pr[X_A = X_A] \cdot \Pr[T_A = T], \tag{4}
\]
where (2) follows from obtaining the marginal distribution from the joint distribution, and (3) follows from Lemma 1.

As for the second statement, from an application of Bayes’ theorem, we obtain
\[
\Pr[X_A = X_A \mid Y_A = Y_A] \propto \mathbb{I}(Y_A = \{L(x)\}_{x \in X_A}) \cdot \Pr[X_A = X_A]
\]
\[
\propto \mathbb{I}(Y_A = \{L(x)\}_{x \in X_A}) \cdot \Pr[X_A = X_A].
\]

Proposition 1. The set of possible values for Alice’s dataset, given Eve’s observations, is
\[
\text{Pos}(X_A) \triangleq \{\tilde{X} : \exists T \in \mathcal{F}_{T_A} \text{ with } T(\tilde{X}) = T_A(X_A)\}.
\]

Proof of Proposition 1. Let $Z = T_A(X_A), C = \text{LC}(T_A)$, and $X \notin \text{Pos}(X_A)$. Then, there does not exist a $T \in \mathcal{F}_A$ such that $T(X) = Z$. But $T \in \mathcal{F}_A$ if and only if $\text{LC}(T) = C$. Thus, $\{T : \text{LC}(T) = C, T(X) = Z\} = \emptyset$, and therefore, it follows from Theorem 1 that
\[
\Pr[X_A = X \mid T_A(X_A) = Z, \text{LC}(T_A) = C] = 0.
\]

Extended version of Example 2. Let $X = \{1, 2, 3, 4, 5\}, X_A = \{2, 3, 4\}, \text{LC}(X) = (++++-)$, and
\[
\mathcal{F} = \left\{ \begin{array}{l}
T_1 : (2, 1, 5, 4, 3) \\
T_2 : (2, 1, 3, 5, 4) \\
T_3 : (1, 2, 3, 5, 4) \\
T_4 : (4, 3, 1, 2, 5) \\
T_5 : (3, 4, 2, 1, 5) \\
T_6 : (5, 2, 4, 3, 1)
\end{array} \right\}
\]

• Suppose $T_A = T_1$. Then, Alice transmits $\{(1,+), (5,-), (4,-)\}$. Under our threat model, we assume that Eve knows $\text{LC}(T_A) = (+++--)$, and therefore, that $T_A$ is in the set
\[
\mathcal{F}_{T_A} = \left\{ \begin{array}{l}
T_1 : (2, 1, 5, 4, 3) \\
T_2 : (2, 1, 3, 5, 4) \\
T_3 : (1, 2, 3, 5, 4)
\end{array} \right\}
\]
which implies that $X_A \in \text{Pos}(X_A) = \{(2, 3, 4), (2, 4, 5), (1, 4, 5)\}$. Thus, Eve knows that $4 \in X_A$ independent of any distribution for $X_A$. As for the the two other values, it follows from Theorem 1 that
\[
\Pr[1 \in X_A] = \sum_{(X \in X)} \Pr[X_A = X \mid T_A(X_A) = (1, 5, 4), \text{LC}(T_A) = (+++--)]
\]
\[
= \frac{\Pr[X_A = (1, 4, 5)] \Pr[T_A = T_3]}{\Pr[X_A = (2, 3, 4)] \Pr[T_A = T_1] + \Pr[X_A = (2, 4, 5)] \Pr[T_A = T_2] + \Pr[X_A = (1, 4, 5)] \Pr[T_A = T_3]}
\]

14
and
\[
\Pr[2 \in X_A] = \sum_{X \subseteq X_A} \Pr[X_A = X] = \Pr[X_A = \bar{X} = \{1, 5, 4\}, \text{LC}(T_A) = \{0, +, -, -\}]
\]
\[
= \Pr[X_A = (2, 3, 4)] \Pr[T_A = T_1] + \Pr[X_A = (2, 4, 5)] \Pr[T_A = T_2] + \Pr[X_A = (1, 4, 5)] \Pr[T_A = T_3]
\]

If we assume uniform distributions for \(T_A\) and uniform iid distribution for samples in \(X_A\), then \(\Pr[1 \in X_A] = 1/3\) and \(\Pr[2 \in X_A] = 2/3\). If we instead assume uniform \(T_A\) and that \(\Pr[X_A = \{2, 3, 4\}] = 0.1\), \(\Pr[X_A = \{2, 4, 5\}] = 0.1\), and \(\Pr[X_A = \{1, 4, 5\}] = 0.4\), then \(\Pr[1 \in X_A] = 2/3\) and \(\Pr[2 \in X_A] = 1/3\).

- Suppose \(T_A = T_4\). Then, Alice transmits \((3, +), (1, -), (2, -)\). Eve knows \(\text{LC}(T_A) = (---)\), and
  \[
  \mathcal{F}_{T_A} = \begin{cases} T_4 : & (4, 3, 1, 2, 5) \\ T_5 : & (3, 4, 2, 1, 5) \end{cases}
  \]
  and \(\text{Pos}(X_A) = \{(2, 3, 4), (1, 4, 3)\}\). Thus, Eve knows that \(\{3, 4\} \subset X_A\). The other probabilities can be derived in a similar way to the above item.

- Suppose \(T_A = T_6\). Then, Alice transmits \((2, +), (4, -), (3, -)\). Eve knows \(\text{LC}(T_A) = (-+-+)\), and
  \[
  \mathcal{F}_{T_A} = \begin{cases} T_6 : & (5, 2, 4, 3, 1) \end{cases}
  \]
  and, therefore, \(\text{Pos}(X_A) = \{(2, 3, 4)\}\). In this case, Eve can exactly determine \(X_A\).

**Theorem 2.** Let \(\mathcal{X}^y = \{x \in \mathcal{X} : L(x) = y\}\). Then, sampling \(T_A\) uniformly from the family \(\mathcal{F}_0 = \{T \in \text{Sym}(\mathcal{X}) : T(\mathcal{X}^y) = \mathcal{X}^y \quad \forall y \in \mathcal{Y}\}\) achieves perfect privacy without altering the structure of the labels.

**Proof of Theorem 2.** Let \(Z = T_A(X_A)\) and \(C = \text{LC}(T_A)\). We note that every \(T \in \mathcal{F}_0\) has the same label configuration, i.e., \(\text{LC}(T) = \text{LC}(X) = C\). It follows then from Theorem 1 that
\[
\Pr[X_A = X | T_A(X_A) = Z, \text{LC}(T_A) = C] = \sum_{T \in \mathcal{F}_0} \mathbb{1}(T(X) = Z) \cdot \mathbb{1}(\text{LC}(X) = C) \cdot \Pr[X_A = X] \cdot \Pr[T_A = T]
\]
\[
= \sum_{T \in \mathcal{F}_0} \mathbb{1}(T(X) = Z) \cdot \Pr[X_A = X] \cdot \Pr[T_A = T].
\]

Since \(T\) is chosen uniformly from \(\mathcal{F}_0\), it follows that
\[
\Pr[X_A = X | T_A(X_A) = Z, \text{LC}(T_A) = C] \propto \Pr[X_A = X] \sum_{T \in \mathcal{F}_0} \mathbb{1}(T(X) = Z) = \Pr[X_A = X] \cdot |\{T \in \mathcal{F}_0 : T(X) = Z\}|. \quad (5)
\]

Since Eve knows the set \(Z\) and the label configuration \(C\), then she also knows the set of labels \(Y_A = \{L(x)\}_{x \in X_A}\). The reason for this is that \(C\) is the vector representation of the labeling function \(L \circ T^{-1}\), and thus, since Eve knows \(C\) and \(Z\), she also knows \(Y_A = L \circ T^{-1}(Z)\). We also note that the equality \(\{L(x)\}_{x \in X} = Y_A\) holds if and only if the labels of \(X\) and \(Z\) are in agreement, i.e., \(|\{x \in X : L(x) = y\}| = |\{z \in Z : L \circ T^{-1}(z) = y\}|\) for every \(y \in \mathcal{Y}\). This is equivalent to the relation
\[
\mathbb{1}(Y_A = \{L(x)\}_{x \in X}) = \prod_{y \in \mathcal{Y}} \mathbb{1}(|X^y| = |Z^y|),
\]
where \(X^y = \{x \in X : L(x) = y\}\) and \(Z^y = \{z \in Z : L \circ T^{-1}(z) = y\}\).

On the other hand,
\[
|\{T \in \mathcal{F}_0 : T(X) = Z\}| = \prod_{y \in \mathcal{Y}} |X^y|! \cdot \mathbb{1}(|X^y| = |Z^y|).
\]

15
We note that $|X^y|$ does not depend on $X$ and is in fact given by the number of elements with value $y$ in $Y_A$. Thus,
\[
|\{T \in \mathcal{F}_0 : T(X) = Z\}| \propto \prod_{y \in Y} \mathbb{I}(|X^y| = |Z^y|) = \mathbb{I}(Y_A = \{L(x)\}_{x \in X}).
\]
Therefore, by substituting in (5), we obtain
\[
\Pr[X_A = X \mid T_A(X_A) = Z, \text{ LC } (T_A) = C] \propto \mathbb{I}(Y_A = \{L(x)\}_{x \in X}) \Pr[X_A = X].
\]
Then, by Theorem 1,
\[
\Pr[X_A = X \mid T_A(X_A) = Z, \text{ LC } (T_A) = C] = \Pr[X_A = X \mid Y_A = Y_A].
\]
In other words, sampling $T$ from $\mathcal{F}_0$ uniformly at random is perfectly private. The fact that it does not alter the structure of the labels follows directly from $\text{LC}(T) = \text{LC}(X)$ for every $T \in \mathcal{F}_0$. □

**Proposition 2.** Let $\mathcal{F}, \mathcal{F'} \subseteq \text{Sym}(\mathcal{X})$ be two families of encoders such that $\mathcal{F} \subseteq \mathcal{F'}$. Then, it is not generally true that $\mathcal{F}^\prime$ is more private than $\mathcal{F}$.

**Proof of Proposition 2.** Let $\mathcal{X} = \{1, 2, 3, 4\}$ with $\text{LC}(\mathcal{X}) = (+ + --)$, $\mathcal{F} = \{T_1 : (1, 2, 3, 4), T_2 : (2, 1, 3, 4), T_3 : (1, 2, 4, 3), T_4 : (2, 1, 4, 3), T_5 : (3, 4, 1, 2)\}$, and $\mathcal{F}' = \{T_1 : (1, 2, 3, 4), T_2 : (2, 1, 3, 4), T_3 : (1, 2, 4, 3), T_4 : (2, 1, 4, 3)\}$. Then, $\mathcal{F}$ is optimal; indeed it is equal to $\mathcal{F}_0$ as defined in Theorem 2. But $\mathcal{F}'$ is not, as it has an LC-anonymity list with a single member, $T_5$. In fact, whenever $T_5$ is selected from $\mathcal{F}'$, Eve can perfectly decode $X_A$. □

**Theorem 3.** Let $\mathcal{F}, \mathcal{F'} \subseteq \text{Sym}(\mathcal{X})$ and $\mathcal{F'} \circ \mathcal{F} = \{T' \circ T : T' \in \mathcal{F}', T \in \mathcal{F}\}$. Then, $\mathcal{F'} \circ \mathcal{F}$ is no less private than $\mathcal{F}$.

**Proof of Theorem 3.** Let $T_A' \circ T_A \in \mathcal{F'} \circ \mathcal{F}$ be Alice’s encoder. We show that if $T_A'$ is given to Eve, then the scheme is as private as if Alice had sampled her encoder from $\mathcal{F}$. From the definition of the label configuration function, $\text{LC}(T_A' \circ T_A) = C$ if and only if the $i$-th elements of $C$ is $C_i = L \circ T_A'^{-1} \circ (T_A')^{-1}(x_i)$. We define $\mathcal{F}_i = (T_A')^{-1}(x_i)$, for every $x_i \in \mathcal{X}$, which is essentially a new ordering for $\mathcal{X}$. Under this new ordering, we denote the label configuration function by $\text{LC}(T_A)$. We note that $L \circ T_A'^{-1}(x_i) = L \circ T_A^{-1} \circ (T_A')^{-1}(x_i) = C_i$, and therefore, $\text{LC}(T_A) = C$. We also define $\mathcal{Z} = (T_A')^{-1}(Z)$, (Alice’s encoded data under the new ordering). Thus,
\[
\Pr[X_A = X_A \mid T_A' \circ T_A(X_A) = Z, \text{ LC } (T_A' \circ T_A) = C, T_A' = T'] = \Pr[X_A = X_A \mid T_A(X_A) = \mathcal{Z}, \text{ LC } (T_A) = C],
\]
i.e., the same distribution as for $\mathcal{F}$ alone. Thus, $\mathcal{F'} \circ \mathcal{F}$ is at least as private as $\mathcal{F}$. Analogous arguments show that $\mathcal{F'} \circ \mathcal{F}$ is also at least as private as $\mathcal{F'}$.

**C Additional experiments**

**Dataset licenses** Both the MIMIC-CXR and CheXperf datasets are publicly available under their own licenses. The MIMIC-CXR and CheXperf datasets are available under the PhysioNet Credentialled Health Data License 1.5.0 license and Stanford University School of Medicine CheXpert Dataset Research Use Agreement respectively.

**Computational cost** All experiments were conducted using Nvidia Tesla V100 or Nvidia RTX A6000 GPUs. All experiments took between 4-6 hours and were primarily bottlenecked by network bandwidth, as our images were hosted on an NFS server. NeuraCrypt models had approximately the same runtime as ViT.
C.1 Evaluating modeling utility

For each diagnosis task and training setting, we also report the AUC on MIMIC-CXR and CheXpert test set individually in Table 3 and Table 4 respectively.

Table 3: Impact of NeuraCrypt on chest x-ray prediction tasks across different training settings. All metrics are ROC AUCs across the MIMIC-CXR test set. Combined-Clear and Combined-Private refer to using a single NeuraCrypt encoder across the combined MIMIC-CXR and CheXpert datasets and two independent NeuraCrypt encoders respectively. Guides of abbreviations for medical diagnosis: (E)dema, (P)neumothorax, (Co)nsolidation, (Ca)rdiomegaly and (A)telectasis.

| Model        | E  | P  | Co | Ca | A  | Average |
|--------------|----|----|----|----|----|---------|
| Train on MIMIC-CXR |    |    |    |    |    |         |
| ViT          | 88 ± 1 | 78 ± 3 | 77 ± 2 | 88 ± 1 | 85 ± 1 | 83      |
| NeuraCrypt-ViT | 88 ± 2 | 81 ± 1 | 73 ± 2 | 88 ± 1 | 84 ± 1 | 83      |
| Train on CheXpert |    |    |    |    |    |         |
| ViT          | 80 ± 1 | 70 ± 1 | 69 ± 4 | 81 ± 1 | 77 ± 0 | 75      |
| NeuraCrypt-ViT | 82 ± 1 | 69 ± 2 | 72 ± 3 | 79 ± 1 | 78 ± 0 | 76      |
| Train on Combined-Clear |    |    |    |    |    |         |
| ViT          | 89 ± 0 | 83 ± 1 | 77 ± 2 | 88 ± 0 | 86 ± 0 | 84      |
| NeuraCrypt-ViT | 89 ± 0 | 82 ± 4 | 79 ± 1 | 88 ± 0 | 85 ± 0 | 85      |
| Train on Combined-Private |    |    |    |    |    |         |
| NeuraCrypt-ViT | 90 ± 1 | 82 ± 2 | 76 ± 3 | 88 ± 1 | 85 ± 1 | 84      |

Table 4: Impact of NeuraCrypt on chest x-ray prediction tasks across different training settings. All metrics are average ROC AUCs on CheXpert test set. Combined-Clear and Combined-Private refer to using a single NeuraCrypt encoder across the combined MIMIC-CXR and CheXpert datasets and two independent NeuraCrypt encoders respectively. Guides of abbreviations for medical diagnosis: (E)dema, (P)neumothorax, (Co)nsolidation, (Ca)rdiomegaly and (A)telectasis.

| Model        | E  | P  | Co | Ca | A  | Average |
|--------------|----|----|----|----|----|---------|
| Train on MIMIC-CXR |    |    |    |    |    |         |
| ViT          | 82 ± 1 | 61 ± 3 | 72 ± 2 | 86 ± 0 | 82 ± 1 | 77      |
| NeuraCrypt-ViT | 83 ± 1 | 63 ± 2 | 71 ± 1 | 86 ± 0 | 83 ± 0 | 77      |
| Train on CheXpert |    |    |    |    |    |         |
| ViT          | 83 ± 1 | 72 ± 1 | 74 ± 1 | 85 ± 1 | 82 ± 0 | 79      |
| NeuraCrypt-ViT | 85 ± 1 | 74 ± 0 | 78 ± 1 | 85 ± 1 | 83 ± 0 | 81      |
| Train on Combined-Clear |    |    |    |    |    |         |
| ViT          | 84 ± 1 | 71 ± 1 | 75 ± 2 | 86 ± 1 | 84 ± 1 | 79      |
| NeuraCrypt-ViT | 85 ± 0 | 70 ± 2 | 78 ± 1 | 87 ± 0 | 84 ± 1 | 81      |
| Train on Combined-Private |    |    |    |    |    |         |
| NeuraCrypt-ViT | 84 ± 2 | 71 ± 3 | 77 ± 2 | 84 ± 2 | 82 ± 1 | 80      |

C.2 Additional attacks

Adversarial attacks. We conducted additional MMD-based adversarial attacks on NeuraCrypt while varying the architecture of $T^*$. We wished to understand if an over-parameterized $T^*$, e.g with 2x the width or 3x the width, would have better success in attacking $T$. For these experiments, we used a hidden dimension of 2048 and a depth of 7 for $T$, and leveraged a $T^*$ with 2x and 3x the hidden dimension of $T$. We trained $T^*$ for 25 epochs using the Adam optimizer and an initial learning rate of $1e-04$. As shown in Table 5, attacks leveraging an over-parameterized $T^*$ failed to uncover $T$, with a $T^*_{\text{MSE}} \geq 1$.

Plaintext attack. While we do not assume that our adversary has access to either parallel data or the NeuraCrypt encoder in our threat model, we also investigated the robustness of NeuraCrypt to plaintext attacks. We hypothesized that given a sufficient amount of parallel data, i.e $(x_i, T(x_i))$ pairs and $x_i \in X$, an attacker could easily learn $T$. For this experiment, we leveraged the combined Mimic-CXR and CheXpert dataset with Cardiomegaly labels, used NeuraCrypt encoding with a
Table 5: MSE of MMD-based adversarial attacks on NeuraCrypt encodings using an over-parametrized $T^*$.

| Encoding       | $\mu$MSE/$T^*$MSE |
|----------------|-------------------|
| NeuraCrypt-width-2x | 4.53 ± 0.05      |
| NeuraCrypt-width-3x | 4.57 ± 0.08      |

depth of 7 and a hidden dimension of 2048, as in prior experiments. We trained an estimated $T^*$ for 50 epochs using the Adam optimizer, an initial learning rate of $1e^{-04}$ and a batch size of 64, and report $T^*$MSE on the test set.

We found that NeuraCrypt cannot defend against plaintext attacks, illustrating that the security of our encoding scheme relies on the lack of parallel data. We found that the plaintext attack obtains $T^*$MSE of 0.25 ± 0.002.

C.3 Permutation encodings.

We hypothesized that for any NeuraCrypt encoder $T \in \mathcal{F}$ that maps $x_i$ to $T(x_i)$ for every $x_i \in X$ and a random permutation $\pi$, there exists a NeuraCrypt encoder $T_\pi \in \mathcal{F}$ that maps $x_i$ to $T(x_{\pi(i)})$ for every $x_i \in X$. For this experiment, we used 128 random samples from the combined Mimic-CXR and CheXpert dataset with cardiomegaly labels, and used a NeuraCrypt encoding with a depth of 7 and hidden dimension of 2048. We trained $T_\pi$ for 1000 epochs to minimize MSE($T(X), T_\pi(\pi(X))$) using the Adam optimizer, an initial learning rate of $1e^{-04}$ and a batch size of 64 and report $T^*$MSE on 128 random samples.

We found that $T_\pi$ could easily learn a mapping from $X$ to a randomly permuted $Z$. In this experiment, we found $T_\pi$ obtained an $T^*$MSE of 0.56 ± 0.020. We note that our permutation encoding experiment only included 128 random samples, and it is unclear at what dataset size a fixed-size $T$ is no longer able to represent arbitrary $X$ to $Z$ bijections.