Generative Adversarial User Privacy in Lossy Single-Server Information Retrieval

Chung-Wei Weng, Yauhen Yakimenka, Hsuan-Yin Lin, *Senior Member, IEEE*, Eirik Rosnes, *Senior Member, IEEE*, and Jörg Kliewer, *Senior Member, IEEE*

***Abstract*—We consider the problem of information retrieval from a dataset of files stored on a single server under both a user distortion and a user privacy constraint. Specifically, a user requesting a file from the dataset should be able to reconstruct the requested file with a prescribed distortion, and in addition, the identity of the requested file should be kept private from the server with a prescribed privacy level. We initiate the study of the tradeoff between download rate, distortion, and propose a new data-driven framework by leveraging recent advancements in generative adversarial models which allows a user to learn efficient schemes in terms of download rate from the data itself. Learning the scheme is formulated as a constrained minimax game between a user which desires to keep the identity of the requested file private and an adversary that tries to infer which file the user is interested in under a distortion constraint. In general, guaranteeing a certain privacy level leads to a higher rate-distortion tradeoff curve, and hence a sacrifice in either download rate or distortion. We evaluate the performance of the scheme on a synthetic Gaussian dataset as well as on the MNIST dataset. For the MNIST dataset, the data-driven approach significantly outperforms a general achievable scheme combining source coding with the download of multiple files.**

I. Introduction

Efficient information retrieval (IR) from a single or several servers storing such datasets under a strict user privacy constraint has been extensively studied within the framework of private information retrieval (PIR). In PIR, first introduced by Chor *et al.* [1], a user can retrieve an arbitrary file from a dataset without disclosing any information (in an information-theoretic sense) about which file she is interested in to the servers storing the dataset. Typically, the size of the queries is much smaller than the size of a file. Hence, the efficiency of a PIR protocol is usually measured in terms of the download cost, or equivalently, the download (or PIR) rate, neglecting the upload cost of the queries. PIR has been studied extensively over the last decade, see, e.g., [2]–[7] and references therein.

Recently, there has been several works proposing to relax the perfect privacy condition of PIR in order to improve on the download cost, see, e.g., [8]–[10]. Inspired by this line of research, we propose to simultaneously relax both the perfect privacy condition and the perfect recovery condition, by allowing for some level of distortion in the recovery process of the requested file, in order to achieve even lower download costs (or, equivalently, higher download rates). We concentrate on the practical scenario in which the dataset is stored on a single server. A problem formulation with arbitrary distortion and leakage functions is presented, which establishes a tri-fold tradeoff between download rate, privacy leakage to the server storing the dataset, and distortion in the recovery process for the user. We show that the optimal rate-distortion-leakage tradeoff is convex (see Lemma 1) and that it allows for a concise information-theoretical formulation in terms of mutual information in the limit of large file sizes (see Theorem 1). In the special case of full leakage to the server, the proposed formulation yields the well-known rate-distortion curve. The typical behavior of the rate-distortion-leakage tradeoff shows that an increased level of privacy leads to a higher rate-distortion tradeoff curve, and hence a sacrifice in either download rate or distortion. Moreover, to overcome the practical limitation of unknown statistical properties of real-world datasets, we consider a data-driven approach leveraging recent advancements in generative adversarial networks (GANs) [11], which allows a user to learn efficient schemes (in terms of download rate) from the data itself. In our proposed GAN-based framework, learning the scheme can be phrased as a constrained minimax game between a user which desires to keep the identity of the requested file private and a server that tries to infer which file the user is interested in, under both a user distortion and a download rate constraint. Similar to [12], where a cross-entropy loss function is used as a discriminative classifier for unlabeled or partially labeled data, the server is modeled as a discriminator in the generalized GAN framework, and also trained with cross-entropy for labeled data. We evaluate the performance of the proposed scheme on a synthetic Gaussian dataset as well as on the MNIST [13] dataset. For the MNIST dataset, the data-driven approach significantly outperforms an achievable scheme based on combining source coding with the download of multiple files, while for the Gaussian dataset, where the source statistics is known, it performs close to this achievable scheme using a variant of the generalized Lloyd's algorithm [14], [15] for the source code.

Similar data-driven approaches, under the names of generative adversarial privacy [16], [17], privacy-preserving adverisal networks [18], and compressive privacy GAN [19], have recently been proposed for learning a privatization mechanism directly from the dataset in order to release it to the public and for generating compressed representations that retain utility while being able to withstand reconstruction attacks. A similar approach was also taken in [20] where a tri-fold tradeoff between rate, distortion, and perception in lossy image

C.-W. Weng, Y. Yakimenka, H.-Y. Lin, and E. Rosnes are with Simula UiB, N–5006 Bergen, Norway (e-mail: chungwei@simula.no; yauhen@simula.no; lin@simula.no; eirikrosnes@simula.no).

J. Kliewer is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, New Jersey 07102, USA (email: jkliewer@njit.edu).

compression was established. To the best of our knowledge, addressing PIR or the extension to nonperfect privacy and recovery in the context of generative adversarial models has not been considered in the open literature so far.

II. Preliminaries and System Model

A. Single-Server Information Retrieval

Consider a dataset containing M files $\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(\mathsf{M})}$ stored on a single server, where each file $\boldsymbol{X}^{(m)} = \left(X_1^{(m)}, \ldots, X_\beta^{(m)}\right)^\mathsf{T}$, $m \in [\mathsf{M}]$, can be seen as a $\beta \times 1$ random vector (according to some probability distribution $P_{\boldsymbol{X}^{(m)}}$) over $\mathbb{F}^\beta$, where $\mathbb{F}$ is any field (finite or infinite). Assume that the user wishes to retrieve the M-th file, $M \in [\mathsf{M}] \triangleq \{1, 2, \ldots, \mathsf{M}\}$, where, for simplicity, M is assumed to be uniformly distributed over $[\mathsf{M}]$.[1] The formal definition of a general single-server IR scheme is as follows.

Definition 1. *An IR scheme $\mathscr{C}$ for a single-server storing M files consists of: (i) A random strategy $\boldsymbol{S}$. (ii) A deterministic query function f_Q that generates a query $\boldsymbol{Q} = f_\mathrm{Q}(M, \boldsymbol{S})$, where query $\boldsymbol{Q}$ is sent to the server. (iii) A deterministic answer function f_A that returns the answer $\boldsymbol{A} = f_\mathrm{A}(\boldsymbol{Q}, \boldsymbol{X}^{[\mathsf{M}]})$ back to the user. (iv) A deterministic reconstruction function $\hat{\boldsymbol{X}} \triangleq f_{\hat{\mathrm{X}}}(\boldsymbol{A}, M, \boldsymbol{Q})$ giving an estimate of the desired file using the answer from the server together with the requested file index M and the query $\boldsymbol{Q}$.*

We are interested in designing an IR scheme such that both the user's *utility* and *privacy* are preserved. On the one hand, this scheme should satisfy the condition of retrievability with a distortion measure $d(\cdot, \cdot)$, i.e.,

$$\mathsf{E}_{M,\boldsymbol{Q}}\left[d(\boldsymbol{X}^{(M)}, \hat{\boldsymbol{X}})\right] \leq \mathsf{D}, \tag{1}$$

where $\mathsf{E}_X[\cdot]$ denotes expectation with respect to the random variable X and D is a given distortion constraint. On the other hand, the user would like to preserve her privacy with the query function, in the sense that the server should not be able to fully determine the identity M of the requested file. The server receives the query $\boldsymbol{Q}$, and the leakage is measured in terms of a leakage metric $\rho(P_{\boldsymbol{Q}|M})$. The query function should be designed such that

$$\rho(P_{\boldsymbol{Q}|M}) \leq \mathsf{L}, \tag{2}$$

where L is the maximum allowed leakage.

Given a query function f_Q and an answer function f_A of an IR scheme, we measure its efficiency in terms of the download rate (in bits per symbol) defined as

$$\mathsf{R}(f_\mathrm{Q}, f_\mathrm{A}) \triangleq \frac{\mathsf{E}_{\boldsymbol{Q},\boldsymbol{A}}\left[\ell_{\boldsymbol{Q}}(\boldsymbol{A})\right]}{\beta} = \frac{1}{\beta}\sum_{\boldsymbol{q}} P_{\boldsymbol{Q}}(\boldsymbol{q})\,\mathsf{E}_{\boldsymbol{A}}\left[\ell_{\boldsymbol{q}}(\boldsymbol{A})\right], \tag{3}$$

where $\ell_{\boldsymbol{q}}(\boldsymbol{a})$ is the length of the answer $\boldsymbol{a}$ for query $\boldsymbol{q}$. Further in the paper, the length $\ell_{\boldsymbol{q}}(\boldsymbol{a})$ is a constant (i.e., independent of $\boldsymbol{a}$ and $\boldsymbol{q}$) that is chosen according to the desired rate.

[1]The assumption that M is uniformly distributed can be lifted.

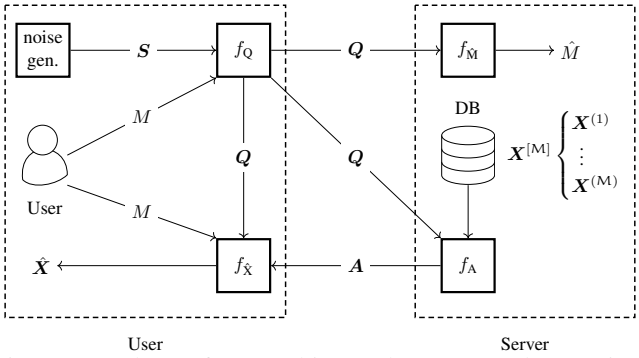


Fig. 1: IR scheme for an arbitrary dataset stored on a single server.

B. Problem Formulation

We consider a model that protects the privacy of the user against a honest-but-curious server. The IR model above guarantees that a user can retrieve an arbitrary file stored on a single server (with some distortion), while the server can only partially infer the identity of the requested file. In this work, we will study the tri-fold tradeoff between download rate, expected distortion, and privacy leakage to the server for an IR scheme. In particular, for a given constraint D on the expected distortion $\mathsf{E}_{M,\boldsymbol{Q}}\left[d(\boldsymbol{X}^{(M)}, f_{\hat{\mathrm{X}}}(\boldsymbol{A}, M, \boldsymbol{Q}))\right]$ for the retrieval of the requested file indexed by M and for a given constraint L on the leakage to the server $\rho(P_{\boldsymbol{Q}|M})$, the goal is to minimize the download rate $\mathsf{R}(f_\mathrm{Q}, f_\mathrm{A})$. This can be formulated as the constrained optimization problem

$$\min_{f_\mathrm{Q}} \min_{f_\mathrm{A}} \quad \mathsf{R}(f_\mathrm{Q}, f_\mathrm{A}) \tag{4a}$$

$$\text{subject to} \quad \mathsf{E}[d(\boldsymbol{X}^{(M)}, \hat{\boldsymbol{X}})] \leq \mathsf{D},\ \rho(P_{\boldsymbol{Q}|M}) \leq \mathsf{L}. \tag{4b}$$

To further capture the information leakage for training, we define $\hat{M} \triangleq f_{\hat{\mathrm{M}}}(\boldsymbol{Q})$ to be the server's best inference of the user's requested file index M from the query $\boldsymbol{Q}$ using a decision decoder $f_{\hat{\mathrm{M}}}$. The proposed model is illustrated in Fig. 1. The leakage on the identity of the requested file can also be quantified by a loss function denoted as $f_{\mathrm{Loss}}(M, \hat{M})$. Hence, the expected loss with respect to M and $\boldsymbol{Q}$ is

$$\mathsf{J}(f_\mathrm{Q}, f_{\hat{\mathrm{M}}}) = \mathsf{E}_{M,\boldsymbol{Q}}\left[f_{\mathrm{Loss}}(M, \hat{M})\right].$$

Here, the leakage metric $\rho(P_{\boldsymbol{Q}|M})$ in (2) is connected to the expected loss $\mathsf{J}(f_\mathrm{Q}, f_{\hat{\mathrm{M}}})$ as $\rho(P_{\boldsymbol{Q}|M}) = \max_{f_{\hat{\mathrm{M}}}} \mathsf{J}(f_\mathrm{Q}, f_{\hat{\mathrm{M}}})$.

Lemma 1. *The set of achievable rate-distortion-leakage triples $(\mathsf{R}, \mathsf{D}, \mathsf{L})$ from (4) is a convex set, for any leakage metric ρ that is convex in $P_{\boldsymbol{Q}|M}$.*

We remark that Lemma 1 also holds for any loss function f_{Loss}, since any $\rho(P_{\boldsymbol{Q}|M})$ that can be expressed as a $\mathsf{J}(f_\mathrm{Q}, f_{\hat{\mathrm{M}}})$ is always convex in $P_{\boldsymbol{Q}|M}$.

Note that if $f_{\mathrm{Loss}}(m, \hat{M})$ is the log-loss function defined as

$$f_{\mathrm{Loss}}(m, \hat{M}) = \mathsf{H}(M) + \log F_{\hat{M}}(m|\boldsymbol{Q}), \tag{5}$$

where $\mathsf{H}(M)$ denotes the entropy of M, then the leakage to the server is measured in terms of MI, i.e., $\rho(P_{\boldsymbol{Q}|M}) = \mathsf{I}(M; \boldsymbol{Q})$, where $\mathsf{I}(\cdot;\cdot)$ denotes MI.

Theorem 1. *The download rate given in* (4) *is equal to the following information rate-distortion-leakage function defined as*

$$\mathsf{R}(\mathsf{D},\mathsf{L}) \triangleq \min_{P_{\boldsymbol{Q}|M},P_{\hat{\boldsymbol{X}}^{[\mathsf{M}]}|\boldsymbol{X}^{[\mathsf{M}]},\boldsymbol{Q}}} \mathsf{I}\big(\boldsymbol{X}^{[\mathsf{M}]};\hat{\boldsymbol{X}}^{[\mathsf{M}]}\,\big|\,\boldsymbol{Q}\big), \tag{6}$$

as $\beta \to \infty$*, where the minimization is subject to* $\mathsf{E}_{M,\boldsymbol{Q}}\big[d(\boldsymbol{X}^{(M)},\hat{\boldsymbol{X}}^{(\mathsf{M})})\big] \le \mathsf{D}$ *and* $\rho(P_{\boldsymbol{Q}|M}) \le \mathsf{L}$.

C. Generative Adversarial Approach

The inference of the server can also be modeled in a generative adversarial fashion [16]–[19]. In particular, the *leakage-distortion* tradeoff for any fixed download rate constraint R is formally described as follows. Consider a family of IR schemes with query generators f_{Q}, answer functions f_{A}, and with download rate $\mathsf{R}(f_{\mathrm{Q}},f_{\mathrm{A}})$ at most R. The goal of the server is to maximize the expected loss $\mathsf{J}(f_{\mathrm{Q}},f_{\hat{\mathrm{M}}})$ of the user by designing the decision function $f_{\hat{\mathrm{M}}}$. In contrast, the user would like to design a scheme with f_{Q}, f_{A}, and $f_{\hat{\mathrm{X}}}$ such that the download rate $\mathsf{R}(f_{\mathrm{Q}},f_{\mathrm{A}}) \le \mathsf{R}$ and such that the expected loss $\mathsf{J}(f_{\mathrm{Q}},f_{\hat{\mathrm{M}}})$ is minimized, while preserving the utility of the scheme, i.e., in the sense that the user can still retrieve the requested file with an expected distortion smaller than a prescribed D. This leads to the constrained minimax optimization problem

$$\min_{f_{\mathrm{Q}}}\max_{f_{\hat{\mathrm{M}}}} \quad \mathsf{J}(f_{\mathrm{Q}},f_{\hat{\mathrm{M}}}) \tag{7a}$$

$$\text{subject to } \mathsf{E}_{M,\boldsymbol{Q}}\left[d(\boldsymbol{X}^{(M)},\hat{\boldsymbol{X}})\right] \le \mathsf{D},\ \mathsf{R}(f_{\mathrm{Q}},f_{\mathrm{A}}) \le \mathsf{R}. \tag{7b}$$

Note that the minimax formulation above can be written in a GAN form [11] in which $f_{\hat{\mathrm{M}}}$ plays the role of the *discriminator* and f_{Q} plays the role of the *generator*. Thus, the machinery of GANs can be used to determine the leakage-distortion tradeoff. The minimax game in (7) with the log-loss function will be the basis for training, as described below in Section III.

III. Data-Driven Approach

In this section, we describe in detail our proposed data-driven framework for constructing an efficient IR scheme for downloading an arbitrary file from an arbitrary dataset stored on a single server. The four functions in Fig. 1 are represented as deep neural networks and we assume, for now, that they have already been trained.

The user wishes to retrieve the M-th file $\boldsymbol{X}^{(M)}$ and encodes M as a one-hot $\boldsymbol{Y} = (Y_1,Y_2,\ldots,Y_{\mathsf{M}}) \in \{0,1\}^{\mathsf{M}}$, where $Y_j = 1$ if $j = M$, and $Y_j = 0$ otherwise. Next, the user generates a "noise" vector $\boldsymbol{S} = (S_1,S_2,\ldots,S_{\mathsf{M}})$, where $S_1,\ldots,S_{\mathsf{M}}$ are independent and identically distributed (i.i.d.) according to the standard Gaussian distribution $\mathcal{N}(0,1)$. The concatenation $(S_1,\ldots,S_{\mathsf{M}},Y_1,\ldots,Y_{\mathsf{M}})$ is the input to a deep neural network, representing the function f_{Q}, for query generation. This network produces the query $\boldsymbol{Q}$ that is sent to the server. The intuition behind this neural network is to hide the value of M. The server's answer is produced by concatenating the stored data $\boldsymbol{X}^{[\mathsf{M}]}$ and the received query $\boldsymbol{Q}$ and then feed the result into a deep neural network, representing the function f_{A}, for answer construction. The deep neural network produces the answer vector $\boldsymbol{A}$ that is sent back to the user. The user then concatenates $\boldsymbol{A}$ and $(S_1,\ldots,S_{\mathsf{M}},Y_1,\ldots,Y_{\mathsf{M}})$ and feeds the result into a deep neural network for decoding, representing the function $f_{\hat{\mathrm{X}}}$, to produce the estimate $\hat{\boldsymbol{X}}$ of the requested file $\boldsymbol{X}^{(M)}$.

On the server side, a deep neural network, representing the function $f_{\hat{\mathrm{M}}}$, is used to guess the identity of the requested file. The input to the network is the query vector $\boldsymbol{Q}$ and the output (using softmax) is a distribution-like vector $\boldsymbol{F} = (F_1,F_2,\ldots,F_{\mathsf{M}})$, where $\sum_{j=1}^{\mathsf{M}} F_j = 1$, $0 \le F_j \le 1$, and where F_j can be interpreted as "with probability/likelihood F_j, the user's requested file index M is equal to j." The server's estimate of the user requested file index is then $\hat{M} = \arg\max_{j\in[\mathsf{M}]} F_j$.

A. Learning Algorithm

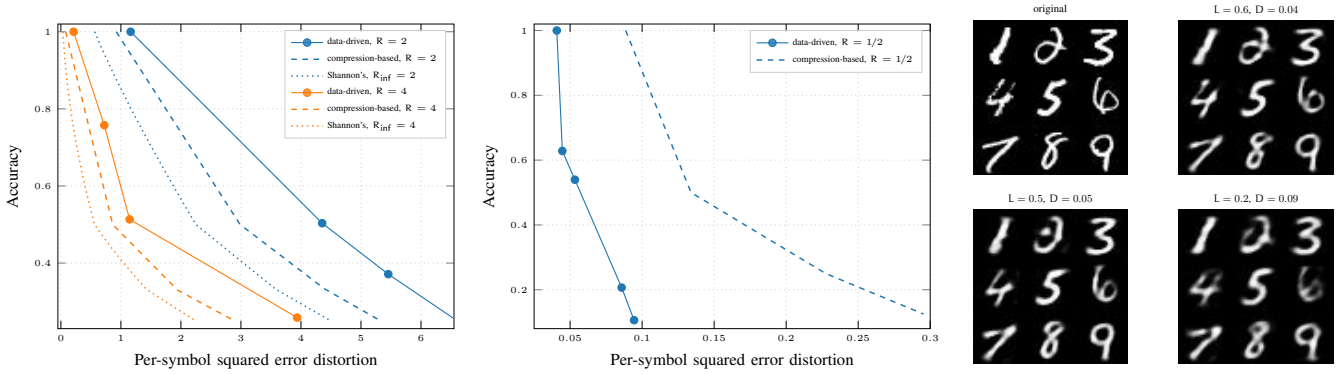
Let $\boldsymbol{X}^{(M_l)} = \big(X_1^{(M_l)},\ldots,X_\beta^{(M_l)}\big)$, $l \in [\mathfrak{n}]$, denote the l-th requested file during training and $\hat{\boldsymbol{X}}_l$ the corresponding user reconstructed estimate, where $\mathfrak{n}$ is the number of training samples. Training the deep neural networks representing the functions f_{Q}, f_{A}, $f_{\hat{\mathrm{X}}}$, and $f_{\hat{\mathrm{M}}}$ is done by first fixing the download rate R and then solving the minimax optimization problem

$$\min_{(f_{\mathrm{Q}},f_{\mathrm{A}},f_{\hat{\mathrm{X}}})}\max_{f_{\hat{\mathrm{M}}}} \frac{1}{\mathfrak{n}}\sum_{l=1}^{\mathfrak{n}}\Big[-f_{\text{XE-Loss}}(\boldsymbol{Y}_l,\boldsymbol{F}_l) + \eta d\big(\boldsymbol{X}^{(M_l)},\hat{\boldsymbol{X}}_l\big)\Big], \tag{8}$$

where $f_{\text{XE-Loss}}(\boldsymbol{Y}_l,\boldsymbol{F}_l) = -\sum_{j=1}^{\mathsf{M}} Y_{l,j}\log F_{l,j} = -\log F_{l,M_l}$ measures the categorical cross-entropy between the vectors $\boldsymbol{Y}_l = (Y_{l,1},\ldots,Y_{l,\mathsf{M}})$ and $\boldsymbol{F}_l = (F_{l,1},\ldots,F_{l,\mathsf{M}})$. Here, $\boldsymbol{Y}_l$ and $\boldsymbol{F}_l$ is the $\boldsymbol{Y}$-vector and $\boldsymbol{F}$-vector for the l-th sample. The parameter η is a tradeoff coefficient that is increased in every epoch. The solution to the minimax optimization problem in (8) is found using an iterative algorithm employing stochastic gradient decent similar to Algorithm 1 in [16]. The solution is found by first maximizing by the objective function of (8) to determine the optimal $f_{\hat{\mathrm{M}}}$ for a fixed initial triple $(f_{\mathrm{Q}},f_{\mathrm{A}},f_{\hat{\mathrm{X}}})$. Next, the optimal triple $(f_{\mathrm{Q}},f_{\mathrm{A}},f_{\hat{\mathrm{X}}})$ is found for the given $f_{\hat{\mathrm{M}}}$ from the previous step by minimizing the same objective function. This iterative process is continued until convergence or the maximum number of iterations is exceeded. Note that, although training is done based on a log-loss function, we evaluate the performance more intuitively using accuracy, i.e., $\Pr(M=\hat{M})$, in Section IV below.

IV. Numerical Results

We demonstrate the application of our proposed data-driven approach to a synthetic Gaussian dataset and also for the MNIST dataset, showing that guaranteeing a certain privacy level leads to a higher rate-distortion tradeoff curve, and hence a sacrifice in either download rate or distortion. We also compare the data-driven approach with both a compression-based scheme (for all datasets) and a scheme that we refer to as Shannon's scheme (for the Gaussian dataset). Here, both the compression-based scheme and Shannon's scheme are constructed from a general achievable scheme combining source coding with the download of multiple files. In particular,



(a) (b) (c)

Fig. 2: Accuracy versus per-symbol squared error distortion for both the data-driven approach, and the schemes based on compression and Shannon's theoretical approach. (a) Synthetic Gaussian dataset. (b) MNIST. (c) MNIST reconstructed digits for $\mathsf{R} = 1/2$.

the scheme achieves a certain level of privacy by downloading (using lossy compression) a subset of the dataset. The Gaussian dataset consists of $\mathsf{M} = 4$ files, each of dimension $\beta = 3$ and drawn independently according to $\mathcal{N}(\boldsymbol{\mu}^{(m)}, \sigma^2 I)$, where $\boldsymbol{\mu}^{(1)} = (-3.0, -3.0, 3.0)^\intercal$, $\boldsymbol{\mu}^{(2)} = (3.0, 3.0, 3.0)^\intercal$, $\boldsymbol{\mu}^{(3)} = (3.0, -3.0, -3.0)^\intercal$, $\boldsymbol{\mu}^{(4)} = (-3.0, 3.0, -3.0)^\intercal$, and $\sigma = 3$. For MNIST, the training set is comprised of $\mathfrak{n} = 6000$ 28×28 images for each digit, and testing is performed on randomly chosen images from 10000 test images. The distortion between a requested file $\boldsymbol{X}^{(m)}$ and its estimate $\hat{\boldsymbol{X}}$ is measured as the per-symbol squared error.

In Fig. 2, we plot the accuracy, or $\Pr(M = \hat{M})$, as function of per-symbol squared error distortion for different download rates R for the data-driven approach (solid curves) and the compression-based scheme (dashed curves). For MNIST a symbol is a grey-scale pixel, while for the Gaussian dataset it is a one-dimensional Gaussian random variable. As a comparison, for the Gaussian dataset, we also plot the performance of Shannon's scheme (dotted curves), assuming $\beta \to \infty$. For Shannon's scheme, we use the *information rate*, denoted by $\mathsf{R}_{\mathsf{inf}}$ and defined as $\mathsf{H}(\boldsymbol{A}|\boldsymbol{Q})/\beta$. It is well-known that the information rate is a true lower bound on the *operational* rate from (3) (cf. [21]). Although $\mathsf{R}_{\mathsf{inf}}$ is a true lower bound on R, their numerical values appear to be close, e.g., for the Gaussian dataset and $\mathsf{D} = 4.35$, $\mathsf{R} = 2$ while $\mathsf{R}_{\mathsf{inf}} \approx 1.97$.

As expected, one can have a higher privacy level (i.e., smaller leakage) for a given distortion at the expense of a higher download rate. For the MNIST dataset, the data-driven approach significantly outperforms the compression-based scheme, while for the Gaussian dataset it performs close to the compression-based scheme using a variant of the generalized Lloyd's algorithm [14], [15] for the source code. For MNIST we combined JPEG-like compression (including discrete-cosine transform), run-length encoding, and other entropy coding techniques for the compression-based scheme (cf. [22, Sec. 8.2]). However, due to a very small size of images, the overhead (e.g., for storing the Huffman codebook) turned out to be unacceptably high. We thus opted for scalar quantization. In Fig. 2(c), we also show the reconstructed digits for three levels of distortion and accuracy for a download rate of $\mathsf{R} = 1/2$ for the MNIST dataset.

So far only the download cost of the proposed schemes has been considered, neglecting both the query upload cost and the cost of distributing the trained neural networks. First, the cost of distributing the trained networks can be neglected since training is usually done beforehand on a dedicated training server and can hence be seen as a one-time cost. This cost vanishes as the protocol can in principle run for a very long time serving a large number of users while the dataset grows continuously. Second, the query upload cost is in most cases much smaller than the download cost of the answers. As we will show below, this is also the case here for MNIST. For completeness, the one-time cost (in bits) of distributing the networks to the user is 465479×32 and 1259135×32 bits for the Gaussian and MNIST datasets, respectively, while the corresponding one-time cost of distributing the answer generation network to the server is 532230×32 and 7336888×32 bits, respectively.

	Download cost (R)	Upload cost	Download dataset
Gaussian	$6\,(2), 12\,(4)$	128	384
MNIST	$392\,(1/2)$	320	17825

The number of neurons of the output layer of the query network is always equal to the number of files M. Thus, the upload cost is at most $32 \cdot \mathsf{M}$ (assuming 32-bits floats). For MNIST, this yields $32{\cdot}10 = 320$ bits, while the download cost is $28{\cdot}28{\cdot}1/2 = 392$ bits (assuming $\mathsf{R} = 1/2$), which is higher. Downloading the entire dataset requires an overall communication cost of $28{\cdot}28{\cdot}8{\cdot}10 = 62720$ bits uncompressed, which is significantly higher. Moreover, the average losslessly compressed sizes for different digits range from 1351 (digit 1) to 1988 (digit 0) and leads to a total average size of 17825 bits, which is again significantly higher than the download cost. In general, the upload cost scales linearly with the number of files and is independent of the file size, while the download cost increases with the file size. In general, the file size is much larger than the number of files which is the standard argument for not considering the upload cost. In the embedded table we summarize the different costs (in bits) for both the Gaussian and MNIST datasets and compare with the cost (in bits) of downloading the entire dataset using lossless source coding.

REFERENCES

[1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," in *Proc. 36th Annu. IEEE Symp. Found. Comp. Sci. (FOCS)*, Milwaukee, WI, USA, Oct. 23–25, 1995, pp. 41–50.

[2] K. Banawan and S. Ulukus, "The capacity of private information retrieval from coded databases," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1945–1956, Mar. 2018.

[3] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk, "Private information retrieval from coded databases with colluding servers," *SIAM J. Appl. Algebra Geom.*, vol. 1, no. 1, pp. 647–664, Nov. 2017.

[4] S. Kopparty, S. Saraf, and S. Yekhanin, "High-rate codes with sublinear-time decoding," in *Proc. 43rd Annu. ACM Symp. Theory Comput. (STOC)*, San Jose, CA, USA, Jun. 06–08, 2011, pp. 167–176. [Online]. Available: http://doi.acm.org/10.1145/1993636.1993660

[5] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4075–4088, Jul. 2017.

[6] R. Tajeddine, O. W. Gnilke, and S. El Rouayheb, "Private information retrieval from MDS coded data in distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7081–7093, Nov. 2018.

[7] S. Yekhanin, "Private information retrieval," *Commun. ACM*, vol. 53, no. 4, pp. 68–73, Apr. 2010.

[8] H.-Y. Lin, S. Kumar, E. Rosnes, A. Graell i Amat, and E. Yaakobi, "The capacity of single-server weakly-private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 21–26, 2020.

[9] I. Samy, R. Tandon, and L. Lazos, "On the capacity of leaky private information retrieval," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 7–12, 2019, pp. 1262–1266.

[10] R. R. Toledo, G. Danezis, and I. Goldberg, "Lower-cost ϵ-private information retrieval," in *Proc. Privacy Enhancing Technol. Symp. (PETS)*, Darmstadt, Germany, Jul. 19–22, 2016, pp. 184–201.

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 28th Int. Conf. Neural Inf. Proc. Syst. (NeurIPS)*, Montreal, Canada, Dec. 8–13, 2014, pp. 2672–2680.

[12] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," in *Proc. Int. Conf. Learn. Representations (ICLR)*, San Juan, Puerto Rico, May 2–4, 2016.

[13] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[14] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[15] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[16] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, "Context-aware generative adversarial privacy," *Entropy*, vol. 19, no. 12, Dec. 2017. [Online]. Available: https://www.mdpi.com/1099-4300/19/12/656

[17] ——, "Generative adversarial privacy," in *Proc. ICML'18 Workshop: Privacy Mach. Learn. Artif. Intell.*, Stockholm, Sweden, Jul. 15, 2018.

[18] A. Tripathy, Y. Wang, and P. Ishwar, "Privacy-preserving adversarial networks," in *Proc. 57th Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Sep. 24–27, 2019, pp. 495–505.

[19] B.-W. Tseng and P.-Y. Wu, "Compressive privacy generative adversarial network," *IEEE Trans. Inf. Forens. & Secur.*, vol. 15, pp. 2499–2513, 2020.

[20] Y. Blau and T. Michaeli, "Rethinking lossy compression: The rate-distortion-perception tradeoff," in *Proc. Int. Conf. Mach. Learn. (ICML)*. Long Beach, CA, USA: PMLR, Jun. 9–15, 2019.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY, USA: Wiley, 2006.

[22] I. Bocharova, *Compression for multimedia*. Cambridge, UK: Cambridge University Press, 2010.