Generative Modeling for Context-Aware Local Privacy: Final Report

Wei-Ning Chen (wnchen) **Dennis Rich (denrich)** **Harry Chen (weichen9)**

1 Introduction

As more personal data is entrusted to social networks, insurance conglomerates, and countless other aggregators, finding ways to maintain user privacy remain critical. Many current such methods have two key disadvantages. First, they are centralized, requiring difficult-to-earn user trust in the data aggregator. Second, they may protect even non-sensitive data from analysis, limiting utility while providing no benefit to the user.

In this work, we presented a framework for understanding and implementing deep learning methods to protect privacy. Critically, this framework is *local* — implemented even at the level of individual user profiles — and *context-aware* — focused on retaining important patterns while obscuring more individual features.

1.1 Related Work

The notion of differential privacy has been explored thoroughly. [3] mathematically analyzes databases that, by the introduction of noise, can trade off between utility and privacy. Many authors [1, 7, 14] continue to build on this idea, developing new ways of implementing these methods and proving their efficacy. Indeed, work is ongoing to scale to the largest of datasets [6]. Furthermore, a subfield of differential privacy — local differential privacy — lags not too far behind [5]. Local differential privacy has been shown to be useful in practical deployment with applications like Google's RAPPOR [8], a database technology for preserving both privacy and utility.

Another thread of research [2, 9, 11] focuses on "context-aware" privacy and aims to improve the poor privacy-utility trade-off in local differential privacy. This is done by erasing sensitive information directly from user profiles while retaining the rest (that is, the utility of the profile). Privacy and utility in this setting are defined in an information-theoretic way and the optimal privatization schemes are obtained by solving stochastic optimization problems. However, the main drawback of this approach is the assumption that data distributions are known beforehand, which, in most piratical applications, is not valid.

In our work, we draw direct inspiration from [12], which proposes a data-driven approach according to generative networks and thus circumvents the need of knowledge about data distributions. [12] analyzes the generative method on low-dimensional parametric model and evaluates on synthetic data, such as Bernoulli or Gaussian mixture models. In this work, We improve their results by giving a statistical foundation to general high-dimensional models and validate it on real-world datasets.

1.2 Our Approach

We focus initially on photographs, expanding upon [11] by not only removing sensitive data but replacing it with privatized but plausible data that reveals nothing about the user profile. We use a generative adversarial network (GAN) [10] architecture, with distortion and privacy as components of the loss function. To keep outputs of the generator plausible, we perturb an encoding of the image instead of the image itself. For the discriminator, we use a convolutional neural net.

In our milestone, we laid out our plan to 'throw applications at the wall and see what sticks'. Below are the applications that stuck.

1.3 Applications

The application targeted in [11] is compelling because it is a simple and easy-to-understand demonstration of the algorithm's effectiveness. We deploy our improved model on the same application, modifying images of faces to privatize whether or not they're smiling. Our results are given in later sections.

In our second application, we more realistically target photos with information users might actually want to keep private. We modify photos of houses, which might disclose house prices and therefore give clue's to the owner's income.

2 Problem Formulation

2.1 Privacy

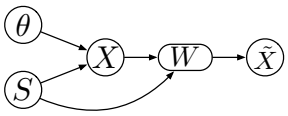

Figure 1: Dependency of random variables

We consider n samples with sensitive information being $S_i \in \mathcal{S}$ and the content being $X_i \in \mathcal{X}$, where $S_i \overset{\text{i.i.d.}}{\sim} P_S(s)$, $X_i \overset{\text{i.i.d.}}{\sim} P_{X|S}(x|S_i;\theta)$ and θ is a latent (insensitive) variable. A channel (i.e. privatizing mechanism) $W : \mathcal{X} \times \mathcal{S} \to \tilde{\mathcal{X}}$ generates and releases perturbed samples $\tilde{X}_i \sim W(\cdot|X_i, S_i)$ to data analysts (as well as adversaries). The goal of the channel is to hide sensitive information in X_i while preserve as much " nonsensitive information", which is defined as our utility, as possible. The graphic model in Figure 1 shows the dependencies between random variables.

In general, the privacy constraint restricts adversaries to infer S_i from $\tilde{X}_i$ for each user i. Mathematically speaking, let $\ell : \mathcal{S} \times \mathcal{S} \to \mathbb{R}^+$ be the loss function, we require the statistical risk (i.e. expected loss) of inferring S_i, after observing $\tilde{X}_i$, is closed to random guess:

$$\inf_{\hat{S}:\tilde{\mathcal{X}}\to\mathcal{S}} \mathbb{E}\left[\ell\left(\hat{S}(\tilde{X}_i), S_i\right)\right] \geq \inf_{s\in\mathcal{S}} \mathbb{E}\left[\ell\left(s, S_i\right)\right] - \epsilon, \quad (1)$$

where the expectations are taken with respect to

$$\left(\tilde{X}_i, S_i\right) \sim P_{\tilde{X},S|\theta} = P_{S,X|\theta} \circ W.$$

Note that the right-hand side in (1) is the risk of "blind guess", which means inferring S_i from the perturbed sample $\tilde{X}_i$ has Bayes's error closed to random guess.

2.2 Utility

Intuitively, the utility of perturbed data depicts how much "information" is preserved, and hence we measure it in two senses:

1. the distortion between original and perturbed data $\mathbb{E}\left[d\left(X, \tilde{X}\right)\right]$, for some distortion function $d(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$,
2. how well we can infer the latent variables $\theta \in \Theta$ (i.e. trying to minimize the statistical risk

$$R(\theta) \triangleq \mathbb{E}\left[\ell\left(\hat{\theta}(\tilde{X}^n), \theta\right)\right].$$

Moreover, in many scenario we can also combine both distortion and statistical risk on θ in our objective. However, due to the limitation of space, we only outline the learning algorithms for minimizing distortion, and it is straightforward to incorporate the statistical risk into the loss function.

3 Learning Method

To minimize the distortion, the optimization problem becomes

$$\begin{aligned} \min_{W} \quad & \mathbb{E}_{P_{X,\tilde{X}}} d(X, \tilde{X}) \\ \text{s.t.} \quad & \|P_{X|s_1} \circ W - P_{X|s_2} \circ W\|_{\mathsf{TV}} \leq \epsilon. \end{aligned} \quad (2)$$

The Variational Method The major difficulty to solve (2) is the lack of knowledge on distributions $P_{S,X}$ and $Q_{S,X}$. To overcome this issue, a standard approach is applying *variational inference* to information divergence, which was originally used in training generative models (GAN) [10]. Variational inference was further studied in f-GAN [15] for general information divergences. Fortunately, total-variation distance also has a simple variational expression

$$\|P - Q\|_{\mathsf{TV}} = \sup_{\|f\|_\infty \leq 1} \mathbb{E}_P f(X) - \mathbb{E}_Q f(X), \quad (3)$$

which allows us to circumvent the direct use of P and Q in solving (2).

Hence we can reformulate the problem as a minimax game:

$$\begin{aligned} \min_{W} \mathbb{E}_{P_{X,\tilde{X}}} d(X, \tilde{X}) + \\ \lambda\left(\max_{f_\omega} \mathbb{E}_{P_{\tilde{X}|s_1}} f_\omega(\tilde{X}) - \mathbb{E}_{P_{\tilde{X}|s_2}} f_\omega(\tilde{X})\right), \end{aligned} \quad (4)$$

and the training approach is similar to auto-encoder, except that now we incorporate both distortion and privacy into the loss function. The stochastic optimization (4) again can be approximated by empirical optimization:

$$\begin{aligned} &\min_{W} \max_{f_\omega} \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} d(X, \tilde{X}) \\ &+\lambda\left(\frac{1}{|\mathcal{D}_{s_1}|} \sum_{\mathcal{D}_{s_1}} f_\omega(W(x)) - \frac{1}{|\mathcal{D}_{s_2}|} \sum_{\mathcal{D}_{s_2}} f_\omega(W(x))\right), \end{aligned}$$

where $\mathcal{D}_{s_1}$ and $\mathcal{D}_{s_1}$ are the collection of x_i whose s-labels are s_1 and s_2 respectively.

Figure 2 and Algorithm 1 show our detailed training approaches.

3.1 Architecture

We implemented generators' (i.e. privatization channels) networks with auto-encoders and use convolutional neural networks for discriminators as shown in Figure 3.

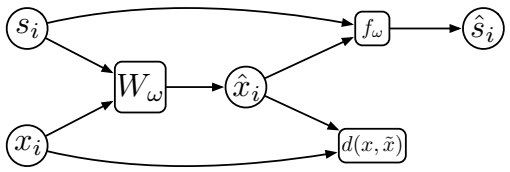

Figure 2: Training Architecture

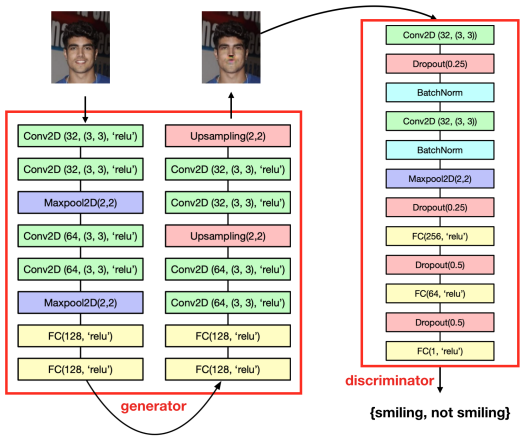

Figure 3: Smiling image privatization

Algorithm 1: Training Privatization Channel

for *number of iterations* **do**

  for k *steps* **do**

    Sample m samples $\{x_1, ..., x_m\}$;

    Perturb samples:

$$\{\tilde{x}_1, ..., \tilde{x}_m\} \leftarrow \{W(x_1), ..., W(x_m)\}$$

    Update discriminators by their stochastic gradients (w.r.t. cross entropy):

$$f_\omega \leftarrow \nabla_\omega \frac{1}{m} \sum_{\tilde{x}_i} (s_i \log(g_\omega(\tilde{x}_i)) + (1 - s_i) \log(1 - g_\omega(\tilde{x}_i))),$$

  end

  Sample m samples $\{z_1, ..., z_m\}$;

  Perturb samples:

$$\{\tilde{z}_1, ..., \tilde{z}_m\} \leftarrow \{W(z_1), ..., W(z_m)\}$$

  Update generator by its stochastic gradient:

$$W_\omega \leftarrow \nabla_\omega \frac{1}{m} \sum_{\tilde{z}_i} d(z_i, \tilde{z}_i) + \lambda(s_i \log(f_\omega(\tilde{z}_i)) + (1 - s_i) \log(1 - f_\omega(\tilde{z}_i))).$$

end

As mentioned in Section 2.2, we can also modify the loss function so that the perturbed sample $\hat{x}_i$ preserves other information we are interested in (which can be characterized by θ_i).

4 Results

We used this algorithm to attack multiple problems. Each required its own dataset and hyperparameter tuning, since hyperparameters were found to be data-dependent.

4.1 Smiling Privatization: Dataset and Hyperparameters

We evaluate our model on CelebA [13], a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. Due to the time and computation power and we have, we train our models on 35K of pre-shuffled samples and validate and test on 10K of the remaining. We do not do any preprocessing such as PCA or LDA. Examples can be found in the first rows of Figure 4 and Figure 6.

As an illustrative example, we set the sensitive variable S_i being the binary label that indicates whether the person in image X_i smiling or not. Though the scenario may not fully capture real-world applications, this experiment allows us to easily visualize results and see whether our algorithm works or not.

4.2 Smiling Privatization: Results

We first run our algorithm with different privacy levels (i.e. λ in (4)) and discuss results [1] in low privacy and high privacy regimes respectively. Each regime involves a different set of hyperparameters.

Though there are a great numbers of hyperparameters in our models (i.e. typologies of generators and discriminators/ dropout probabilities/ batch sizes...), we elaborate some that significantly affect the privacy levels.

- Number of epochs in pre-training discriminator and generator (an auto-encoder): `epochs_pre_d`, `epochs_pre_g`
- Number of iteration in adversarial training: `epochs_train`
- Number of updates on discriminator and generator in each single iteration : `itr_d`, `itr_g`

Note that in the pre-train stage, in each epoch we sweeps all $35K$ training images and update the dis-

[1] Source codes available in `https://github.com/WeiNingChen/GAPP-image`.

criminator and the generator. However in adversarial training stage, each epoch we only update the discriminator (generator) `itr_d`(`itr_g`) times, each time with `minibatch_size` = 128 training samples.

Interestingly, through the long and painful process of tuning them, we found some rule of thumbs that differ vastly from training traditional GANs and list them in Table 1. Such differences are mainly due to the discrepancy between goals of GAN and generative privacy: GAN aims to generate plausible examples *from pure random noise*, while generative privacy *modifies a given sample* in order to erase sensitive information.

GAN	our work
Do not pre-train discriminator	**Do** pre-train discriminator
Update generators more frequently	Depends on different privacy guarantees
-	Use different training sets to train D and G

Table 1: Differences between GAN and our generative adversarial privacy

Low privacy regime

In low privacy regime, we set hyperparameters as follows:

$$\texttt{epochs\_pre\_d} = 25,\ \texttt{epochs\_pre\_g} = 25,$$
$$\texttt{epochs\_train} = 300,\ \texttt{minibatch\_size} = 128,$$
$$\texttt{itr\_d} = 15,\ \texttt{itr\_g} = 20.$$

The privatized results are given in Figure 4.

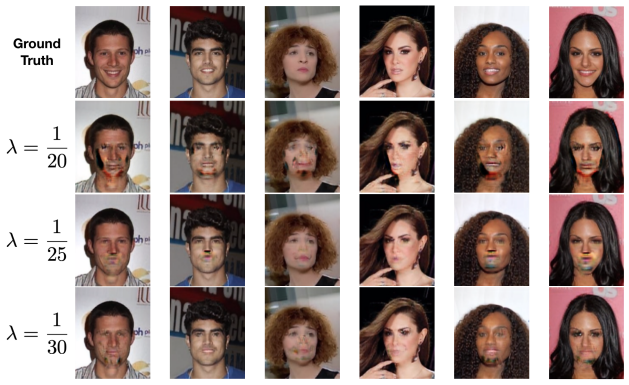

Figure 4: Smiling image privatization in low privacy regime. Larger λ gives stronger privacy guarantee.

Though the privatized images seem able to fool human eyes, one can use a CNN to attack the privacy (i.e. train on the perturbed images with correct labels) and can get roughly 85% accuracy. Compared to the original unperturbed smile detection task , in which one can achieve 90% accuracy, we only increase the privacy (i.e. the indistinguishableness) by 5% or so. The unsatisfactory results motive us to reinforce the discriminator by increasing `itr_d` and hope to get a better privacy guarantee.

High privacy regime

We thus set hyperparameters as follows:

$$\texttt{epochs\_pre\_d} = 10,\ \texttt{epochs\_pre\_g} = 10,$$
$$\texttt{epochs\_train} = 50,\ \texttt{minibatch\_size} = 128,$$
$$\texttt{itr\_d} = 800,\ \texttt{itr\_g} = 500,$$

and the results are in Figure 5. From Figure 5 we observe a significantly larger distortion, but this also gives us a better privacy guarantee. Indeed, the accuracy of a CNN attacker, under $\lambda = 0.3$ and $\lambda = 0.1$, are about 57% and 79% respectively. Of course we can keep increasing λ and obtain better privacy.

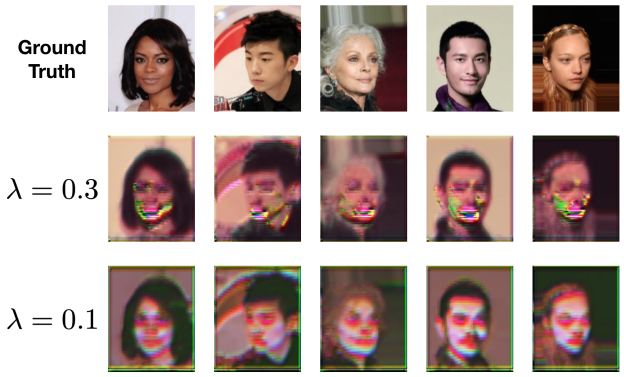

Figure 5: Smiling image privatization in high privacy regime.

4.3 Housing Privatization: Dataset and Hyperparameters

Having proven and refined our model, we attempted to extend it to a more useful case. Any publicly available photos of homes could disclose information about the price of those homes. With this, potential employers could learn about their candidates' socioeconomic status. With our algorithm, we could privatize this information in photos on social media or news articles while preserving the usefulness of the photo.

We used price-labeled home images localized to SoCal and scraped from Weichert (a real estate conglomerate) [4]. We train on 12K images and test on 2K. Due to the limitation of GPU memory, we resize each image to 208x106 pixels. The prices are also normalized by 100k, which restricts the range of prediction to $[0, 20]$. Examples can be found in Figure 6.

We didn't have enough time to adjust hyperparameters for this new application, using the low privacy regime

hyperparameters from the smiling dataset. However, house price is no longer a binary label, so we now use MSE to implement the cost function defining privacy, and a ReLU final layer output instead of a sigmoid.

4.4 Housing Privatization: Results

Using original hyperparameters, we find little success, as shown in Figure 6. The perturbing 'generator' appears to just add noise by blurring the house images. Oddly, the problem isn't with our generator, but our discriminator. We output very high MSE even on our unperturbed data: 14.9 for training error and 19.9 for testing error (recall that we normalize the prices by 100k so the range of outputs is in $[0, 20]$). Obviously our model does not generalize well in the task.

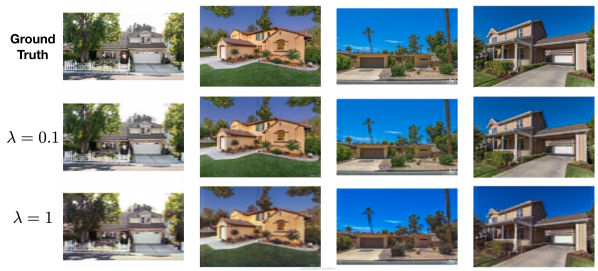

Figure 6: SoCal House Price Prediction

That's not to say this problem isn't possible. There are a number of things we could try to get better results if we had time. First, we could use a pre-trained discriminator. The dataset we downloaded also reports a discriminator performing the same function as ours, with accuracy within 7 percent [4]: much better than ours. This is because [4] uses more information, such as sizes, locations, or number of bedrooms, in prediction. Our suspicion is that the house prices are highly uncorrelated to their appearances, so predicting prices only based on images is fundamentally inaccurate. However, an easier version could be possible: relabelling the training set to be more representative of the use case. Instead of defining privacy with MSE, we could re-label houses as high, medium, and low cost, and train accordingly. Those concerned about house price privacy would probably not care exactly how far the 'guessed' price was from the true price, as long as it was far enough.

If it were possible to get the discriminator working, we could try tuning our hyperparameters with the log-distributed random clustering method taught in class. The results of the final tuned network would be interesting. We hypothesize they would be related to the occlusion sensitivity maps we discussed in lecture 6: the most important pixels for the discriminator would be most distorted by the generator.

5 Conclusion

5.1 Analysis

Results of our main application (smiling faces) show that our algorithm is functional, and improves upon [11] by perturbing encodings of data instead of merely obscuring pixels. We demonstrate a privacy-distortion tradeoff as expected, increasing λ to 0.3 and observing a decrease in discriminator performance to 57 percent: barely better than guessing. Our attempt to extend this application to a more useful case fails, because the information we try to privatize was, we hypothesize, never there in the first place. Perturbing the encoding, therefore, just results in the addition of random noise.

5.2 Discussion and Future Work

We can compare the privacy-distortion tradeoff we observe in this work and compare to other related works, such as [11] and [5]. For each of our applications, this trade-off informs how strongly particular information is conveyed from sets of data, which informs whether that data is appropriate to share. In the future, we could also evaluate our algorithm on synthetic parametric datasets. From these results, we could analytically derive the fundamental lower bound on the trade-off and see how far our scheme is from optimal.

Additionally, in our current work, we only aim to minimize distortion subject to privacy constraints. In many realistic applications, distortion may not be the only target to minimize. For example, in the CelebA dataset [13] there are 40 labeled attributes (gender, age, hair color, etc.). We could extend our methods by casting the problem as *multi-task* learning, in which we want to predict all non-private attributes from the perturbed data while keeping a sensitive attribute private. We would incorporate the error of predicting these target attributes into the loss function by introducing additional classifiers and doing adversarial training on the discriminator, classifiers and the generator simultaneously.

Finally, we could extend the model beyond photos. Social media profiles can give away income data as well to a trained discriminator. We could therefore use our architecture to perturb these profiles and determine how best to keep high user data utility while preventing these attacks. Furthermore, we could extend the method to NLP by using the same architecture, but using an NLP autoencoder as a generator component instead of an image autoencoder. One application of this architecture could be an anti-spam filter, where spammers write initial email drafts and run them through a network to perturb them and avoid a spam filter (the discriminator).

6 Contribution and Acknowledgements

6.1 Contributions

Wei-Ning — As the most experienced in the field among us, Wei-Ning spearheaded the architectural innovations. Harry and Dennis learned a lot from him.

Dennis — The leader of efforts towards a 'practical' application (although ours didn't turn out too well). Did most of the writing.

Harry — Debugging and some hyperparameter tuning. Best ping-pong player among us.

6.2 Thanks!

This project was a great learning experience, and we appreciate your efforts to make CS 230 a good time.

References

[1] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005.

[2] F. P. Calmon and N. Fawaz. Privacy against statistical inference. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1401–1408, Oct 2012.

[3] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

[4] Ted Dogan. House Price Prediction via Computer Vision, 2019.

[5] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Local privacy and statistical minimax rates. In *IEEE 54th Annual Symposium on Foundations of Computer Science*, Oct. 2013.

[6] Cynthia Dwork. Differential privacy. *Proceedings of the 33rd international conference on Automata, Languages and Programming*, pages 1–12, July 2006.

[7] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[8] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 1054–1067, New York, NY, USA, 2014. ACM.

[9] Joseph Geumlek and Kamalika Chaudhuri. Profile-based privacy for locally privacy computations. In *IEEE Symposium on Information Theory*, 2019.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[11] Hsiang Hsu, Shahab Asoodeh, and Flavio P Calmon. Discovering information-leaking samples and features. *NeurIPS Workshop on Privacy and Machine Learning*, 2019.

[12] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Context-aware generative adversarial privacy. *Entropy*, 19(12):656, Dec 2017.

[13] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[14] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[15] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 271–279. Curran Associates, Inc., 2016.