DATA-DRIVEN ALGORITHMS FOR GAUSSIAN MEASUREMENT MATRIX DESIGN IN COMPRESSIVE SENSING

Yang Sun[1] *and Jonathan Scarlett*[1,2]

[1]Department of Computer Science, National University of Singapore
[2]Department of Mathematics & Institute of Data Science, National University of Singapore

ABSTRACT

In this paper, we provide two data-driven algorithms for learning compressive sensing measurement matrices with Gaussian entries. In contrast to the ubiquitous i.i.d. Gaussian design, we associate different variances with different signal entries, so that we may utilize training data to focus more energy on the "most important" parts of the signal. Our first algorithm is based on simple variance-proportional sampling (i.e., place more energy at locations where the signal tends to vary more), and our second overcomes limitations of the first by iteratively up-weighing and down-weighing the variance values according to reconstructions performed on the training signals. Our algorithms enjoy the advantages of being simple and versatile, in the sense of being compatible with a diverse range of signal priors and/or decoding rules. We experimentally demonstrate the effectiveness of our algorithms under both generative priors with gradient-based recovery and sparse priors with ℓ_1-minimization based recovery.

***Index Terms*—** Compressive sensing, measurement matrix design, data-driven techniques, generative priors.

1. INTRODUCTION

Compressive sensing (CS) is a fundamental problem in signal processing and high-dimensional statistics, consisting of recovering an unknown vector from an underdetermined set of linear measurements by assuming low-dimensional structure such as sparsity. Recently, there has been a push towards incorporating *data-driven* techniques into several aspects of CS, including the decoding procedure [1, 2], the measurement matrix [3, 4], and the signal prior itself [5, 6]. See [7, 8] for detailed surveys.

In this paper, our focus is on measurement matrix design. Training data can be used for measurement matrix design in many ways, including tuning a parametrized design [9], selecting a subsampling pattern [4], and training a deep neural network to learn a the matrix directly [3]. These approaches naturally exhibit a trade-off between simplicity, training cost, effectiveness for reconstruction, and so on.

This work was supported by the Singapore National Research Foundation (NRF) under grant number R-252-000-A74-281.

We provide two data-driven algorithms for measurement matrix design, adopting the simple (but to our knowledge, novel) approach of considering matrices with independent Gaussian entries and using training data to optimize *non-identical variances* associated with each column. Our proposed algorithms are simple, versatile with respect to the choice of prior and/or decoder, and efficient in terms of computation and the required amount of training data. We evaluate our algorithms experimentally under both generative priors with gradient-based recovery [5] and non-uniform sparse priors with ℓ_1-minimzation based recovery [3].

1.1. Related Work

The related work is extensive, and we only seek to provide a representative sample of the most related works.

Perhaps closest in spirit to our work is that of optimizing subsampling patterns in [4]. The idea therein is that instead of uniform random subsampling in the sampling domain (e.g., Fourier or wavelet), the subsampling pattern can be optimized based on training data, with significant performance improvements even given very few training samples. (See also [10, 11] for extensions and applications in MRI.) Our work analogously seeks to improve the i.i.d. Gaussian design by using data to optimize the variances in an i.-non-i.d. Gaussian design. Optimized Gaussian measurements are complementary to optimized subsampling-based measurements, and may be preferable in applications where finding a good basis for subsampling is difficult.

Another popular approach has been to optimize the constant of the Restricted Isometry Property (RIP) restricted to the training data, e.g., see [12, 13, 14]. Such techniques are focused on preserving geometry among a given set of training signals, rather than directly optimizing recovery performance. The former can be a reasonable proxy for the latter, but may also require fairly large amounts of data to generalize well.

Finally, recent advances in deep learning have naturally led to learning measurement matrices using deep learning techniques [3, 15, 16]. Such techniques often attain state-of-the-art performance, but with the caveat of usually requiring a large amount of training data, and sometimes also requiring large amounts of computation to train.

2. SETUP

We consider the problem of recovering a target signal $\mathbf{x} \in \mathbb{R}^n$ via linear measurements of the form $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, where $\mathbf{y} \in \mathbb{R}^\ell$ is the measurement vector, $\mathbf{A} \in \mathbb{R}^{\ell \times n}$ is the measurement matrix, and $\mathbf{z} \in \mathbb{R}^\ell$ is additive noise. Given knowledge of both $\mathbf{A}$ and $\mathbf{y}$, a decoder recovers an estimate $\hat{\mathbf{x}}$ which ideally closely approximates $\mathbf{x}$. For concreteness, we consider the per-entry mean squared error (MSE) criterion:

$$\mathrm{MSE}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{n}\|\mathbf{x} - \hat{\mathbf{x}}\|^2. \tag{1}$$

The CS problem typically focuses on $\ell \ll n$, which is only well-posed when $\mathbf{x}$ is suitably structured according to some prior. The techniques of this paper can, in principle, be applied under any prior, but are perhaps most naturally paired with a generative prior [5].

We are interested in designing $\mathbf{A}$ given access to a set of representative training signals $\mathbf{x}_1, \dots, \mathbf{x}_m$. We restrict our attention to Gaussian measurement designs, where each entry of $\mathbf{A}$ is drawn independently from a Gaussian distribution. The ubiquitous i.i.d. design uses all-equal variances, whereas we seek to improve on this using unequal variances. Using the training signals to optimize these variance can be viewed as a form of *power allocation*.

Specifically, we consider each row of $\mathbf{A}$ as being independently drawn from the same distribution, with that distribution being in the form of n independent zero-mean Gaussians with possibly different variances denoted by $(v_1, \dots, v_n)$. To avoid the trivial solution of increasing every v_i to infinity to drown out the noise, we assume a constraint of the form

$$\sum_{i=1}^{n} v_i \le nP \tag{2}$$

for some "power level" P. In the following, we provide two algorithms for choosing $(v_1, \dots, v_n)$ via the training data.

3. PROPOSED ALGORITHMS

3.1. Variance-Proportional Sampling

Intuitively, if the signals of interest always take near-identical values at certain entries, then it may be wasteful to allocate a large amount of power to them, particularly when the number of measurements is very limited. Accordingly, our first strategy simply computes the empirical variance $\hat{\sigma}_i^2$ of each signal entry (with respect to the m values in $\mathbf{x}_1, \dots, \mathbf{x}_m$), and chooses $v_i \propto \hat{\sigma}_i^2$, where the constant of proportionality is selected such that (2) holds with equality. This yields

$$v_i = nP \cdot \frac{\hat{\sigma}_i^2}{\sum_{j=1}^{n} \hat{\sigma}_j^2}. \tag{3}$$

While variance-proportional sampling is natural and can be effective, it is easy to find examples where it is unsuitable. For instance, we will see a success case in Section 4 for the MNIST data set (consisting of handwritten images), but we can highlight a slight variant of MNIST where this would no longer be the case: Simply duplicate every image, and invert the colors in each duplicate (i.e., we now have both black text on a white background *and* vice versa).

For this "MNIST with inversion" data set, a given pixel will be black in half the images and white in half the images, and thus, $\hat{\sigma}_i^2$ will be roughly identical for every i, thus reducing (3) to the i.i.d. Gaussian design. Hence, even though focusing the energy on the more important parts of the image (i.e., stroke locations instead of background locations) would be preferable, this strategy fails to do so in this example.

3.2. Iteratively-Learned Power Allocation

Our second and main algorithm is designed to overcome the limitation discussed in the preceding paragraphs. In this approach, we *incorporate the decoder* into the learning procedure, by iteratively applying it to randomly-chosen signals in the data set. Given the estimate $\hat{\mathbf{x}}$, we up-weigh the v_i whose corresponding entries were estimated the least accurately, and down-weigh those that were estimated the most accurately.

In more detail, the algorithm is as follows, with parameters $\alpha \in \left(0, \frac{1}{2}\right)$ (proportion updated), $\{\lambda_i\}_{i=1}^T$ with $\lambda_i > 0$ (update weight sequence), $B > 0$ (mini-batch size), and $T > 0$ (number of iterations):

1. Initialize $(v_1, \dots, v_n) = (P, \dots, P)$
2. Perform the following for iterations $t = 1$ to T:
   - Generate a random matrix $\mathbf{A}$ according to the current variance parameters $(v_1, \dots, v_n)$.
   - For B randomly-chosen training signals $\{\mathbf{x}_j\}$, produce $\mathbf{y}_j = \mathbf{A}\mathbf{x}_j + \mathbf{z}_j$ and apply the decoder to produce $\hat{\mathbf{x}}_j$. Using $(\mathbf{x}_j, \hat{\mathbf{x}}_j)$, calculate the MSE of each signal entry averaged over its B values.
   - For the entries with the top αn MSE values, set $v_i = v_i e^{-\lambda_t}$; for those with the bottom αn values, set $v_i = v_i e^{\lambda_t}$; for the rest, leave v_i unchanged.
   - Rescale $(v_1, \dots, v_n)$ such that $\sum_{i=1}^n v_i = nP$, i.e., multiply each entry by $\frac{nP}{\sum_{i=1}^n v_i}$.

We observe that the update rule can be viewed as a form of the popular *multiplicative weights* rule [17], used by several research communities. It has the benefit of allowing (exponentially) fast increases or decreases in weights so that they can be adapted quickly. Overall, our algorithm naturally favors placing more energy on parts of the signal that are harder to estimate. As an extreme example, if some part of the image is trivial to estimate (e.g., always zero), its associated weight will continually shrink. There are several possible choices for $\{\lambda_t\}_{t=1}^T$, and we will shortly choose one that slowly decays to zero, so that larger updates are performed earlier on.

Returning to the "MNIST with inversion" example above, even with a small number of measurements, it is easy to identify whether the unknown image is black text on a white background or vice versa. Hence, the areas near the image boundary (which are always part of the background) will be estimated correctly, and the variance at those locations will be down-weighed accordingly.

4. EXPERIMENTS

In this section, we evaluate our algorithms in two settings that utilize distinct priors and decoding rules. We refer to our two algorithms as PROPORTIONAL and ITERATIVE.

4.1. Experimental Setup

We separate the available data into train/test differently in each experiment, described below, and compute the MSE averaged over the test set. We set $P = 1$, so that the baseline strategy is $\mathbf{A}$ having i.i.d. standard Gaussian entries. For PROPORTIONAL (Section 3.1), the training size is denoted by m, and will be specified below. For ITERATIVE (Section 3.2), the training size is BT, and we set $\alpha = \frac{1}{3}$, and use the decaying sequence $\lambda_t = \lambda_0 \cdot \eta^{t-1}$ with $\lambda_0 = 0.1$ and $\eta = 0.95$. We set $T = 20$, and specify B below. There choices were made based on very minimal manual tuning, and detailed fine-tuning may help further improve the performance.

4.2. MNIST data set

We first consider the MNIST data set of handwritten digits with image size 28×28, i.e., $n = 784$. Since pre-trained generative models are readily available, we adopt the powerful decoding strategy of [5] based on a generative prior. For training PROPORTIONAL we use a large number of images $m = 50000$ (though much smaller m behaves similarly), and for ITERATIVE we set $B = 20$ and $T = 20$. Both algorithms use the same test set of size 300. We let the additive noise $\mathbf{z}$ be i.i.d. $N(0, \sigma^2)$ for some noise variance $\sigma^2 > 0$, considering various choices of $\sigma^2 \in \{0.1, 3.0, 8.0, 15.0\}$.

We observe in Figure 1 that our data-driven methods can visibly reduce the MSE compared to i.i.d. Gaussian measurements, particularly when at higher noise levels and the number of measurements is not too low.

The final variance maps are shown in Figure 2. Both of our approaches naturally place more energy at the locations where the MNIST number strokes lie. Unlike PROPORTIONAL, the maps for ITERATIVE depend on the noise level σ^2 and number of measurements ℓ. In particular, as we increase ℓ, the energy map appears to become more "fine-grained", i.e., less uniform around the center of the image.

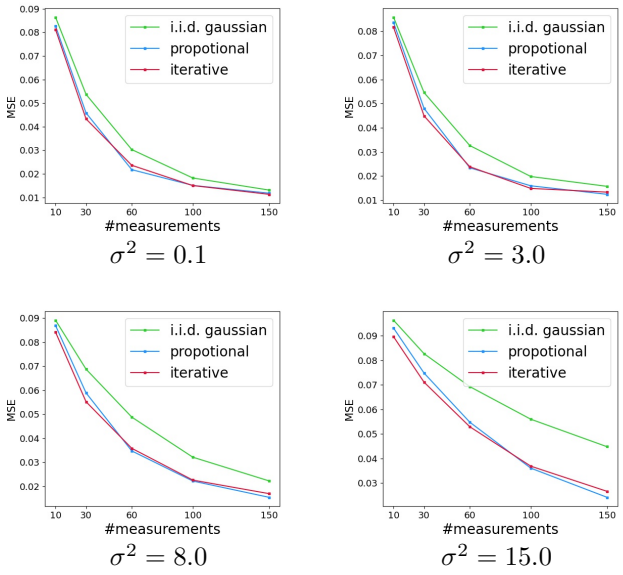

Fig. 1: Recovery performance on MNIST data.

4.3. Non-uniform sparse signals

Next, we replicate an experiment that was used to test the deep learning approach in [3]. This is a one-dimensional synthetic vector data set, in which the i-th entry of the vector is non-zero with probability $\min\left\{1, \frac{c}{i}\right\}$ for some $c > 0$. Conditioned on being non-zero, the value itself is uniform in $[0, 1]$. We set the vector length as $n = 1000$, and choose $c = 10$.

We compare against a deep learning based autoencoder design, called ℓ_1-AE, proposed in [3]. All algorithms share the same training and test sets, with $m = 6000$ and $BT = 300 \times 20 = 6000$, and the test set size is 2000. Unlike the previous example, we consider noiseless recovery here, and following [3], we fix the decoder to be ℓ_1-minimization (subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$).

We observe in Figure 3 that our learned measurement matrices are again effective and outperform i.i.d. measurements. In most cases, the performance falls only slightly short of the more sophisticated deep learning approach ℓ_1-AE, which can even be marginally outperformed by PROPORTIONAL when the number of measurements is very low. Overall, at least in this example, our algorithms are competitive despite being significantly simpler.

We again plot the generated variance maps in Figure 4, truncating to highlight the main differences.

4.4. CelebA-HQ data set

Finally, we consider the CelebA-HQ data set [18] using a pre-trained Glow generative model [19], again adopting generative model based decoding [5]. We crop the original $256 \times 256 \times 3$ images to $128 \times 128 \times 3$ such that only human face in the center is focused. For PROPORTIONAL we set $m = 2000$,

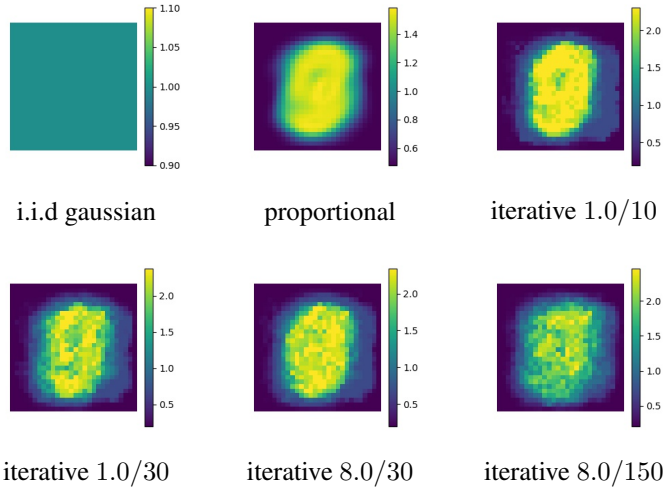

Fig. 2: Variance map generated for MNIST. For each plot of ITERATIVE, we write σ^2/ℓ in the caption, i.e., noise variance and number of measurements.

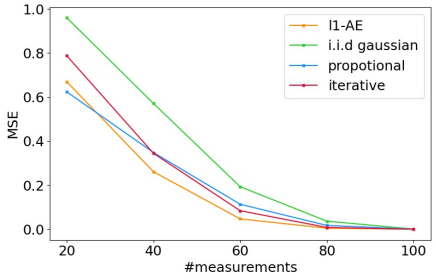

Fig. 3: Recovery performance on synthetic data.

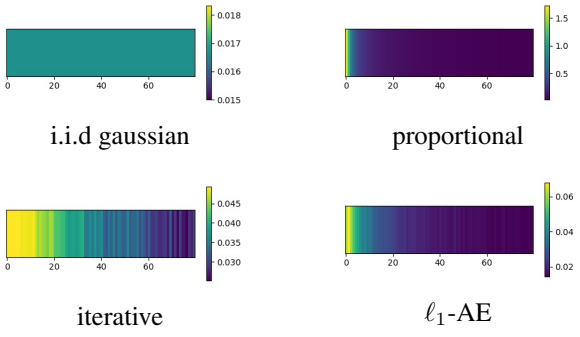

Fig. 4: Variance map for synthetic data. The length of each signal is 1000, and we only plot the first 80 entries.

and due to its more expensive training, we only set the training set size of ITERATIVE as $BT = 1 \times 20 = 20$ and test set size as 3. We consider the noise levels $\sigma^2 = 8.0$ and $\sigma^2 = 16.0$.

We observe in Figure 5 that our data-driven methods again produce lower MSE then i.i.d. Gaussian measurements. The variance maps in Figure 6 bottom show that compared to PROPORTIONAL, our main algorithm ITERATIVE avoids placing too much energy around the corners, and instead focuses on the most "ambiguous" regions, e.g., the eyes, noses, and lips.

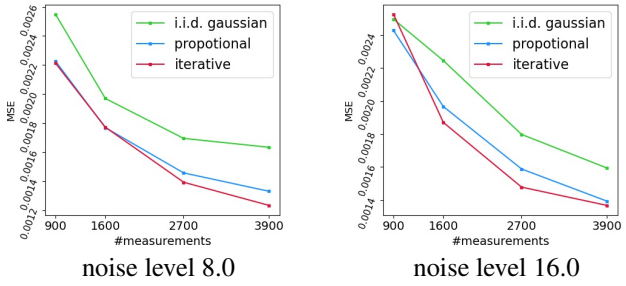

Fig. 5: Recovery performance on cropped CelebA data.

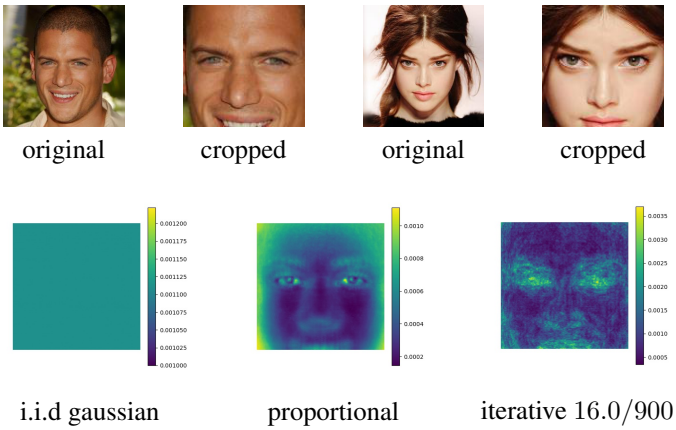

Fig. 6: Original and cropped CelebA data (top); Variance map generated for CelebA with $\sigma^2 = 16$ and $\ell = 900$ (bottom).

5. CONCLUSION

We have introduced two data-driven algorithms for learning power allocations in Gaussian compressive sensing measurement matrices. We observed experimentally that these algorithms can improve over standard i.i.d. Gaussian measurements under both sparse and generative priors, and on both synthetic and real data. In future work, it may be of interest to identify further areas for improvement and move towards application-driven experimental settings.

Acknowledgment: This work was supported by the Singapore National Research Foundation (NRF) under grant number R-252-000-A74-281.

6. REFERENCES

[1] Hemant K Aggarwal, Merry P Mani, and Mathews Jacob, "Modl: Model-based deep learning architecture for inverse problems," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 394–405, 2018.

[2] Jonas Adler and Ozan Öktem, "Learned primal-dual reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1322–1332, 2018.

[3] Shanshan Wu, Alex Dimakis, Sujay Sanghavi, Felix Yu, Daniel Holtmann-Rice, Dmitry Storcheus, Afshin Rostamizadeh, and Sanjiv Kumar, "Learning a compressed sensing measurement matrix via gradient unrolling," in *Int. Conf. Mach. Learn. (ICML)*. PMLR, 2019, pp. 6828–6839.

[4] Luca Baldassarre, Yen-Huan Li, Jonathan Scarlett, Baran Gözcü, Ilija Bogunovic, and Volkan Cevher, "Learning-based compressive subsampling," *IEEE J. Selected Topics in Sig. Proc.*, vol. 10, no. 4, pp. 809–822, 2016.

[5] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis, "Compressed sensing using generative models," in *Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 537–546.

[6] Viraj Shah and Chinmay Hegde, "Solving linear inverse problems using GAN priors: An algorithm with provable guarantees," in *IEEE Int. Conf. Acoust. Sp. Sig. Proc. (ICASSP)*, 2018, pp. 4609–4613.

[7] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett, "Deep learning techniques for inverse problems in imaging," *IEEE J. Selected Topics in Sig. Proc.*, vol. 1, no. 1, pp. 39–56, 2020.

[8] Alice Lucas, Michael Iliadis, Rafael Molina, and Aggelos K Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods," *IEEE Sig. Proc. Magazine*, vol. 35, no. 1, pp. 20–36, 2018.

[9] Michael Lustig, David Donoho, and John M Pauly, "Sparse mri: The application of compressed sensing for rapid mr imaging," *Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[10] Baran Gözcü, Rabeeh Karimi Mahabadi, Yen-Huan Li, Efe Ilıcak, Tolga Cukur, Jonathan Scarlett, and Volkan Cevher, "Learning-based compressive MRI," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1394–1406, 2018.

[11] Thomas Sanchez, Baran Gözcü, Ruud B van Heeswijk, Armin Eftekhari, Efe Ilıcak, Tolga Çukur, and Volkan Cevher, "Scalable learning-based sampling optimization for compressive dynamic MRI," in *IEEE Int. Conf. Acoust. Sp. Sig. Proc. (ICASSP)*, 2020, pp. 8584–8588.

[12] Ali Sadeghian, Bubacarr Bah, and Volkan Cevher, "Energy-aware adaptive bi-Lipschitz embeddings," in *Int. Conf. Samp. Theory and Apps. (SampTA)*, 2013.

[13] Chinmay Hegde, Aswin C Sankaranarayanan, Wotao Yin, and Richard G Baraniuk, "Numax: A convex approach for learning near-isometric linear embeddings," *IEEE Trans. Sig. Proc.*, vol. 63, no. 22, pp. 6109–6121, 2015.

[14] Kai Xu, Yixing Li, and Fengbo Ren, "A data-driven compressive sensing framework tailored for energy-efficient wearable sensing," in *IEEE Int. Conf. Acoust. Sp. Sig. Proc. (ICASSP)*, 2017, pp. 861–865.

[15] Shuaichao Li, Wanqing Zhang, Ying Cui, Hei Victor Cheng, and Wei Yu, "Joint design of measurement matrix and sparse support recovery method via deep autoencoder," *IEEE Sig. Proc. Letters*, vol. 26, no. 12, pp. 1778–1782, 2019.

[16] Iris A.M. Huijben, Bastiaan S. Veeling, and Ruud J.G. van Sloun, "Deep probabilistic subsampling for task-adaptive compressed sensing," in *Int. Conf. Learn. Repr. (ICLR)*, 2020.

[17] Sanjeev Arora, Elad Hazan, and Satyen Kale, "The multiplicative weights update method: a meta-algorithm and applications," *Theory of Computing*, vol. 8, no. 1, pp. 121–164, 2012.

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *Int. Conf. Computer Vision (ICCV)*, 2015.

[19] Durk P Kingma and Prafulla Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Conf. Neur. Inf. Proc. Sys. (NeurIPS)*, 2018.