Batch Denoising via Blahut-Arimoto

Qing Li
Western Digital Research
Milpitas, CA, 95035
Qing.Li7@wdc.com

Cyril Guyot
Western Digital Research
Milpitas, CA, 95035
Cyril.Guyot@wdc.com

Abstract

In this work, we propose to solve batch denoising using Blahut-Arimoto algorithm (BA). Batch denoising via BA (BDBA), similar to Deep Image Prior (DIP), is based on an untrained score-based generative model. Theoretical results show that our denoising estimation is highly likely to be close to the best result. Experimentally, we show that BDBA outperforms DIP significantly.

1 Introduction

Our primary goal is to perform batch denoising using Blahut-Arimoto algorithm (BA). Denoising is the process of removing noise from a noisy observation in order to recover the true data. Denoising is essential in modern image processing because images are always contaminated by noise during acquisition, compression, and transmission. Blahut-Arimoto [1, 2] is a well-known information theory algorithm for computing either the information theoretic capacity of a channel or the rate-distortion function of a source numerically.

In this paper, we propose a method for solving batch denoising by sampling from the rate-distortion posterior computed by BA. Batch denoising via BA (BDBA), similar to Deep Image Prior (DIP) [3], is based on an *untrained* score-based generative model (SBM) [4, 5]. Theoretical results show that our denoising estimation is highly likely to be close to the best result. Experimentally, we show that BDBA outperforms DIP significantly.

2 Background

2.1 Batch Denosing

Given a noisy observation (OB) dataset $\{\mathbf{y} : \mathbf{y} \in \mathbb{R}^n\}$ consisting of n-dimensional i.i.d. samples, that is, $\mathbf{y} = \mathbf{x}^* + \mathbf{e}$, where $\mathbf{x}^* \in \mathbb{R}^n$ is the ground truth data, $\mathbf{e} \sim \mathcal{N}(0, \sigma)$, and $\mathbf{e} \in \mathbb{R}^n$, the objective is to learn a denoising model $p(\mathbf{x}|\mathbf{y})$ such that

$$D^* := \mathbb{E}_{\mathbf{y}}[\|\mathbf{x} - \mathbf{x}^*\|_2^2], \tag{1}$$

approaches zero, where $\mathbf{x} \sim p(\mathbf{x}|\mathbf{y})$.

2.2 Rate Distortion

Given a distribution $p(\mathbf{y})$ and a rate-distortion trade-off $\beta \in \mathbb{R}^+$ associated with a distortion metric $\rho(\cdot)$, the objective of *rate-distortion* is

$$R(\beta) := \min_{p(\mathbf{x}), p(\mathbf{x}|\mathbf{y})} I(\mathbf{x}; \mathbf{y}) + \beta \mathbb{E}[\rho(\mathbf{x}, \mathbf{y})], \tag{2}$$

where $I(\mathbf{x}; \mathbf{y})$ denotes the mutual information, and β controls the trade-off between rate, i.e., $I(\mathbf{x}; \mathbf{y})$, and distortion, i.e., $D := \mathbb{E}[\rho(\mathbf{x}, \mathbf{y})]$.

NeurIPS 2022 Workshop on Score-Based Methods.

Let denote *optimized* $p(\mathbf{x})$ and $p(\mathbf{x}|\mathbf{y})$ achieving (2) by $p^*(\mathbf{x})$ and $p^*(\mathbf{x}|\mathbf{y})$.

That is,

$$p^*(\mathbf{x}|\mathbf{y}) := \underset{\{p(\mathbf{x}|\mathbf{y}):\mathbf{y},\mathbf{x}\sim p(\mathbf{y})p(\mathbf{x}|\mathbf{y})\}}{\arg\min} I(\mathbf{x};\mathbf{y}) + \beta\mathbb{E}[\rho(\mathbf{x},\mathbf{y})], p^*(\mathbf{x}) = \int p(\mathbf{y})p^*(\mathbf{x}|\mathbf{y})d\mathbf{y}. \tag{3}$$

$p^*(\mathbf{x}|\mathbf{y})$ and $p^*(\mathbf{x})$ are characterized by [6, chapter 10, pp. 330]

$$p^*(\mathbf{x}|\mathbf{y}) = \frac{1}{Z_\beta(\mathbf{y})}p^*(\mathbf{x})\exp[-\beta\rho(\mathbf{y},\mathbf{x})], Z_\beta(\mathbf{y}) := \int p^*(\mathbf{x})\exp[-\beta\rho(\mathbf{x},\mathbf{y})]d\mathbf{x}. \tag{4}$$

2.3 Score-Based Generative Models

Diffusing data to noise with an SDE Let $p(\mathbf{x})$ denote the unknown distribution of a dataset. Score-Based Generative Models (SBM [4, 5]) employs a stochastic differential equation (SDE) to diffuse $p(\mathbf{x})$ towards a noise distribution. The SDEs are of the form

$$d\mathbf{x} = \mathbf{f}(\mathbf{x},t)dt + g(t)d\mathbf{w}, \tag{5}$$

where $\mathbf{f}(\mathbf{x},t):\mathbb{R}^n\to\mathbb{R}^n$ is the drift coefficient, $g(t)\in\mathbb{R}$ is the diffusion coefficient, and $\mathbf{w}\in\mathbb{R}^n$ denotes a standard Wiener process (a.k.a., Brownian motion). Intuitively, we can interpret $d\mathbf{w}$ as infinitesimal Gaussian noise. The solution to (5) is a diffusion process $\{\mathbf{x}(t)\}_{t\in[0,T]}$, where $[0,T]$ is a fixed time horizon.

That is, an SDE smoothes the data distribution by adding noise and gradually removing structure until little of the original remains.

Generating samples with the reverse SDE Consider the following reverse SDE,

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x},t) - g(t)^2\nabla_{\mathbf{x}}\log p_t(\mathbf{x})]dt + g(t)d\tilde{\mathbf{w}}, \tag{6}$$

where $\tilde{\mathbf{w}}$ is a standard Wiener process in the reverse-time direction, and the above SDE must be solved from $t=T$ to $t=0$. The solution to (6) is the same diffusion process $\{\mathbf{x}(t)\}_{t\in[0,T]}$ as (5), assuming it is initialized with $\mathbf{x}(T)\sim p_T(\mathbf{x})$.

That is, starting with samples of $\mathbf{x}(T)\sim p_T(\mathbf{x})$ and reversing the process of (5), we gradually remove noise to obtain samples $\mathbf{x}(0)\sim p_0(\mathbf{x})$, where $p_0(\mathbf{x}) = p(\mathbf{x})$.

As a result, the training objective of SBM is to learn $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$, the time-dependent score function via a neural network, $\mathbf{s}_\theta(\mathbf{x},t)$, such that $\mathbf{s}_\theta(\mathbf{x},t) = \nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ for $0\le t\le T$. The sampling is via Langevin dynamic [7, 8].

3 Methods

3.1 Proposed solution

Let $\rho(\mathbf{x},\mathbf{y}) := \|\mathbf{y}-\mathbf{x}\|_2^2$. We propose batch denoising $\{\mathbf{y}\}$ by sampling $\mathbf{x}\sim p^*(\mathbf{x}|\mathbf{y})$, where $p^*(\mathbf{x}|\mathbf{y})$ is computed by Algorithm 1.

BDBA is presented in Algorithm 1, where θ^k denotes the trained SBM at the k^{th}-iteration. More specifically, the inputs of the algorithm are $\{\mathbf{y}\}$, $\rho(\mathbf{x},\mathbf{y})$, and a selected β; The algorithm is based on an *untrained* SBM, i.e., Line 2; The algorithm then alternately loops between two steps until θ^k converges: denoising $\mathbf{y}$ by sampling $\mathbf{x}\sim p_{\theta^k}(\mathbf{x}|\mathbf{y})$ via Langevin dynamic, i.e., Line 5, where $p_{\theta^k}(\mathbf{x}|\mathbf{y})$ is the same as (4) except $p^*(\mathbf{x})$ is replaced by $p_{\theta^k}(\mathbf{x})$; optimizing θ^k based on SBM training, i.e., Line 6; Finally, the denoised result $\{\mathbf{x}\}$ is returned, i.e., Line 10.

That is, an SBM, θ, is trained to model $\nabla_{\mathbf{x}}\log p^*(\mathbf{x})$ and correspondingly $\nabla_{\mathbf{x}}\log p^*(\mathbf{x}|\mathbf{y})$ due to Lemma 2. $p^*(\mathbf{x}|\mathbf{y})$ is our denoising model, and the denoising can be done by Langevin dynamic.

Algorithm 1 BDBA

procedure BDBA($\{\mathbf{y}\}, \beta, \rho(\cdot)$)
  $k \leftarrow 0$ and initialize θ^k arbitrarily ▷ untrained oSBM
  while not converged **do**
    for $\mathbf{y} \in \{\mathbf{y}\}$ **do**
      sample $\mathbf{x} \sim p_{\theta^k}(\mathbf{x}|\mathbf{y})$ via Langevin dynamic ▷ refer to Lemma 2
      update θ^k with $\mathbf{x}$ s.t. $\mathbf{s}_{\theta^k}(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ for $0 \leq t \leq T$ ▷ refer to SBM training, i.e.,Sec 2.3
    end for
    $k \leftarrow k + 1$
  end while
  return $\{\mathbf{x}\}$
end procedure

3.2 Theoretical analysis

Lemma 1. *Assuming θ has enough capacity to fully represent any score function, $(p_{\theta^k}(\mathbf{x}), p_{\theta^k}(\mathbf{x}|\mathbf{y})) \rightarrow (p^*(\mathbf{x}), p^*(\mathbf{x}|\mathbf{y}))$ when $k \rightarrow \infty$.*

Lemma 1 states that $(p_{\theta^k}(\mathbf{x}), p_{\theta^k}(\mathbf{x}|\mathbf{y}))$ learned by BDBA converges to rate-distortion posterior $(p^*(\mathbf{x}), p^*(\mathbf{x}|\mathbf{y}))$ regardless of how θ is initialized (see Fig. 2b). The proof is deferred to Appendix A.

Lemma 2. *Assume $\nabla_{\mathbf{x}} \log p_t^*(\mathbf{x})$ is represented by one SBM, $S_\theta(\mathbf{x}, t)$, i.e., $S_\theta(\mathbf{x}, t) = \nabla_{\mathbf{x}} \log p_t^*(\mathbf{x})$ (refer to Sec 2.3). $\nabla_{\mathbf{x}} \log p_t^*(\mathbf{x}|\mathbf{y})$ can be represented by $S_\theta(\mathbf{x}|\mathbf{y}, t) := S_\theta(\mathbf{x}, t) - \nabla_{\mathbf{x}} \beta \rho(\mathbf{x}, \mathbf{y})$.*

As a result of Lemma 2, sampling from $p^*(\mathbf{x}|\mathbf{y})$ is possible with Langevin dynamics. The proof is deferred to Appendix B.

Theorem 1.

$$Pr[\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : \|\mathbf{x} - \mathbf{x}^*\|_2^2 \leq \frac{(1+\alpha)}{(1-\alpha)} \left(\frac{R(\beta)}{\beta} - \sigma^2 \right) + \frac{2 \ln \frac{1}{\eta}}{n\beta(1-\alpha)}] \geq 1 - 2\eta, \quad (7)$$

where $\eta > 0$, $0 < h\beta < 1$, $\alpha := \frac{2\sigma^2}{\frac{1}{\beta} - h}$, and $\beta > h + 2\sigma^2$.

The proof is deferred to Appendix C. Theorem 1 states that our denoising estimation is highly likely to be close to the best result.

How to select β? Based on (4), when $\beta \rightarrow \infty$, $p^*(\mathbf{x}|\mathbf{y}) = \mathbb{1}_{\mathbf{y}=\mathbf{x}}$ (i.e., $p^*(\mathbf{x}|\mathbf{y}) = 1$ if $\mathbf{y} = \mathbf{x}$ otherwise 0), that is sampling from $p^*(\mathbf{x}|\mathbf{y})$ tends to reproduce $\mathbf{y}$ exactly; when $\beta = 0$, $p^*(\mathbf{x}|\mathbf{y}) = p^*(\mathbf{x})$, that is sampling from $p^*(\mathbf{x}|\mathbf{y})$ tends to produce a random $\mathbf{x}$ which is independent of $\mathbf{y}$. That is, β controls the distortion between $\mathbf{x}$ and $\mathbf{y}$. Therefore, a *properly* selected β gives a *good* denoising result shown by Theorem 2.

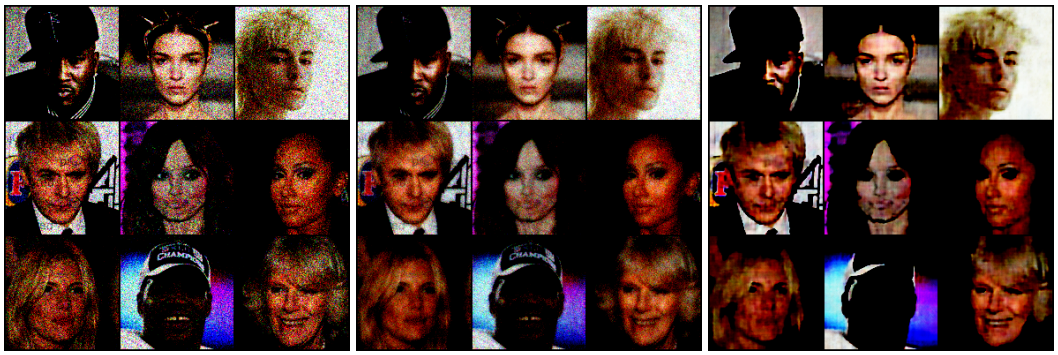
Theorem 2. *Given the batch denoising problem, when β is selected such that $D := \mathbb{E}\|\mathbf{y} - \mathbf{x}\|_2^2 = \sigma^2$, we have $H(\mathbf{x}) \rightarrow H(\mathbf{x}^*)$ when $n \rightarrow \infty$.*

The proof is deferred to Appendix D. Theorem 2 suggests that if β satisfies $D = \sigma^2$, i.e., the empirical MSE average of $\mathbf{x}$ and $\mathbf{y}$ is equal to the noise magnitude of $\mathbf{x}^*$ and $\mathbf{y}$, then the entropy of $\mathbf{x}$ is equal to that of $\mathbf{x}^*$ asymptotically. The experimental demonstration is shown in Fig. 2a.

4 Experiment

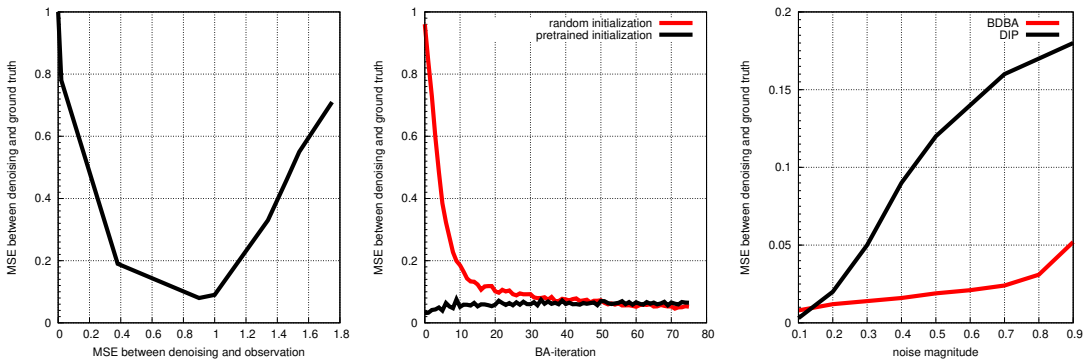
Since BDBA is an untrained method, we now compare it with state of the art untrained denoising methods, such as the DIP method (as opposed to learned methods in [9]). We compare denoising results with 171 randomly selected images of CelebA-HQ [10]. Appendix E contains the details of the experiment.

The denoising examples are presented in Fig. 1 and quantitative results are in Fig. 2. Fig. 2a shows D-D^* curve when denoising via BDBA with $\mathbf{e} \sim \mathcal{N}(0, 1)$. Fig. 2a shows that when $D = 0.9$, D^* obtains its minimum value at 0.05. The discrepancy between this and Theorem 2 which predicts when $D = 1$, D^* obtains its minimum value at 0 is because Theorem 2 holds asymptotically. Fig. 2b compares the D^* evolution when θ is initialized randomly or by a pretrained model. Fig. 2b shows that both converge to the same D^* eventually. However, BDBA with a pretrained initialization converges faster and gives a lower D^* initially. As shown by Fig. 2c, BDBA significantly outperforms DIP especially for large σ.



(a) OB ($\mathbf{e} \sim \mathcal{N}(0, 0.25)$) ; (b) BDBA (MSE = 0.02); (c) DIP (MSE = 0.06).

Figure 1: Denoising example.



(a) D vs D^* ($\mathbf{e} \sim \mathcal{N}(0, 1)$); (b) D^* evolution ($\mathbf{e} \sim \mathcal{N}(0, 1)$); (c) Comparison with DIP.

Figure 2: Quantitative results.

5 Related Work

The Bayesian posterior sampling method was proposed in [9], and it was shown to be nearly optimal. Our rate-distortion posterior sampling differs from [9] in two ways: first, we use the rate-distortion posterior rather than the Bayesian posterior; second, BDBA, similar to DIP [3], is based on an untrained SBM [4, 5] rather than a pretrained SBM as [9].

Some work has been done for inverse problems using pretrained SBMs via unsupervised methods, for example, [11, 12, 13, 14]. Song et al. has presented a framework for all inverse problems [4]; Kawar et al. [11, 13] presented a denoising method by sampling from a posterior distribution based on [15]; Choi et al. [12] demonstrated super-resolution methods with SBMs; Compressive-sensing methods via SBM has been presented by Song et al. [16]. However, no work on solving the denoising problem with an untrained SBM has been presented.

6 Conclusions and Future work

We propose to solve the batch denoising problem by sampling from the rate-distortion posterior computed using Blahut-Arimoto. Theoretical results show that our estimate is highly likely to be close to the best result. We demonstrate empirically that our method outperforms Deep Image Prior.

The proposed framework can theoretically be generalized to other noises, such as Laplacian noise, by replacing a noise-corresponding distortion. Our empirical results, however, show that the denoising performance is inferior to other denoising methods. We leave the investigation as future work.

References

[1] Suguru Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.

[2] Richard Blahut. Computation of channel capacity and rate-distortion functions. *IEEE transactions on Information Theory*, 18(4):460–473, 1972.

[3] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[4] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

[6] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

[7] Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, pages 341–363, 1996.

[8] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[9] Ajil Jalal, Sushrut Karmalkar, Alexandros G Dimakis, and Eric Price. Instance-optimal compressed sensing via posterior sampling. *arXiv preprint arXiv:2106.11438*, 2021.

[10] 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.

[11] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Stochastic image denoising by sampling from the posterior distribution. *arXiv preprint arXiv:2101.09552*, 2021.

[12] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14367–14376, 2021.

[13] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Snips: Solving noisy inverse problems stochastically. *arXiv preprint arXiv:2105.14951*, 2021.

[14] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.

[15] Zahra Kadkhodaie and Eero P Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640*, 2020.

[16] Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. *arXiv preprint arXiv:2111.08005*, 2021.

[17] Cecil C Craig. On the tchebychef inequality of bernstein. *The Annals of Mathematical Statistics*, 4(2):94–102, 1933.

[18] Abbas El Gamal and Young-Han Kim. *Network Information Theory*. Cambridge University Press, Cambridge, United Kingdom, 2012.

[19] Andrew R Barron. Complexity regularization with application to artificial neural networks. In *Nonparametric functional estimation and related topics*, pages 561–576. Springer, 1991.

[20] Qing Li and Yang Chen. Denoising Deep Botlzmann Machines. In *2020 Data Compression Conference (DCC)*, pages 13–22. IEEE, 2020.

[21] Qing Li, Yang Chen, and Yongjune Kim. Compression by and for deep boltzmann machines. *IEEE Transactions on Communications*, 68(12):7498–7510, 2020.

[22] Chayne Planiden and Xianfu Wang. Most convex functions have unique minimizers. *arXiv preprint arXiv:1410.1078*, 2014.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

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

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

A Proof to Lemma 1

Proof. The proof is based on [6, Theorem 10.5] and the assumption that θ has enough capacity to fully represent any score function. □

B Proof to Lemma 2

Proof. Based on (4), we have $\nabla_{\mathbf{x}} \ln p^*(\mathbf{x}|\mathbf{y}) = \nabla_{\mathbf{x}} \ln p^*(\mathbf{x}) - \beta \nabla_{\mathbf{x}} \rho(\mathbf{x}, \mathbf{y})$. Thus replacing $\nabla_{\mathbf{x}} \ln p^*(\mathbf{x}|\mathbf{y})$ with $S_\theta(\mathbf{x}, t)$ finishes this part. □

C Proof to Theorem 1

Its proof breaks down to prove the following parts:

- $\|\mathbf{x} - \mathbf{x}^*\|_2^2$ satisfies

$$Pr\{\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : \|\mathbf{x} - \mathbf{x}^*\|_2^2 \leq \frac{(1+\alpha)}{(1-\alpha)}[\frac{R(\beta)}{\beta} - \sigma^2] + \frac{2 \ln \frac{1}{\eta}}{n\beta(1-\alpha)}\} \leq 1 - 2\eta, \quad (8)$$

  where $\eta > 0$, $0 < h\beta < 1$, $\alpha := \frac{2\sigma^2}{\frac{1}{\beta} - h}$, $R(\beta) := \min_{p(\mathbf{x}), p(\mathbf{x}|\mathbf{y})}[I(\mathbf{x}; \mathbf{y}) + \beta \mathbb{E}(\|\mathbf{y} - \mathbf{x}\|_2^2)]$, and $\beta > h + 2\sigma^2$.

- The expectation of $\|\mathbf{x} - \mathbf{x}^*\|_2^2$ is upper-bounded by

$$\mathbb{E}[\|\mathbf{x} - \mathbf{x}^*\|_2^2 - \frac{(1+\alpha)}{(1-\alpha)}(\frac{R(\beta)}{\beta} - \sigma^2)] \leq \frac{4}{\beta(1-\alpha)n}. \quad (9)$$

The main technique is Craig–Bernstein inequality [17, p.96] and the rate-distortion function for lossy compression. We present Craig–Bernstein inequality [17, p.96].

Lemma 3. *Let W_i ($i = 1, 2, \cdots, n$) be a set of independent random variables with average $\bar{W} = \frac{1}{n} \sum_{i=1}^{n} W_i$. For all $\tau \geq 0$ and $\epsilon, c \in (0, 1)$, we have*

$$Pr[\bar{W} - \mathbb{E}\bar{W} \geq \frac{\tau}{n\epsilon} + \frac{n\epsilon Var(\bar{W})}{2(1-c)}] \leq e^{-\tau}. \quad (10)$$

C.1 Proof to (8)

Proof. Let us consider U_i associated with $\ln \frac{p_e(\mathbf{y}_i|\mathbf{x}_i)}{p_e(\mathbf{y}_i|\mathbf{x}_i^*)}$. That is

$$U_i := |\mathbf{y}_i - \mathbf{x}_i|^2 - |\mathbf{y}_i - \mathbf{x}_i^*|^2 = [\mathbf{x}_i - \mathbf{x}_i^*]^2 - 2[\mathbf{y}_i - \mathbf{x}_i^*][\mathbf{x}_i - \mathbf{x}_i^*]. \quad (11)$$

its mean is $\mathbb{E}U_i = [\mathbf{x}_i - \mathbf{x}_i^*]^2$, and its variance is $Var(U_i) = 4\sigma^2[\mathbf{x}_i - \mathbf{x}_i^*]^2$ due to the Gaussian assumption.

Now, let us consider $\bar{U} = \frac{1}{n}\sum_i U_i = \frac{1}{n}\sum_i |\mathbf{y}_i - \mathbf{x}_i|^2 - \frac{1}{n}\sum_i |\mathbf{y}_i - \mathbf{x}_i^*|^2 = \|\mathbf{y} - \mathbf{x}\|_2^2 - \|\mathbf{y} - \mathbf{x}^*\|_2^2$. Its mean is $\mathbb{E}\bar{U} = \|\mathbf{x} - \mathbf{x}^*\|_2^2 = \mathbb{E}[\|\mathbf{y} - \mathbf{x}\|_2^2] - \sigma^2$, and its variance is $Var(\bar{U}) = \frac{4\sigma^2}{n}(\|\mathbf{x} - \mathbf{x}^*\|_2^2)$.

Applying the Craig-Bernstein inequality to $-\bar{U}$ with $\tau = nI(\mathbf{x}; \mathbf{y}) + \ln \frac{1}{\eta}$, $\epsilon = \beta$, and $c = h\beta \in (0, 1)$ yields

$$Pr[\|\mathbf{x} - \mathbf{x}^*\|_2^2 - \bar{U} \geq \frac{I(\mathbf{x}; \mathbf{y})}{\beta} + \frac{\ln \frac{1}{\eta}}{n\beta} + \frac{2\sigma^2\|\mathbf{x} - \mathbf{x}^*\|_2^2}{\frac{1}{\beta} - h}] \leq e^{-nI(\mathbf{x};\mathbf{y})}\eta. \quad (12)$$

Let $\alpha := \frac{2\sigma^2}{\frac{1}{\beta} - h}$ and apply (12) to all $\mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y})$ and all $\mathbf{y}$ with union found, thus we have

$$Pr[\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : (1-\alpha)\|\mathbf{x} - \mathbf{x}^*\|_2^2 \geq \bar{U} + \frac{I(\mathbf{x}; \mathbf{y})}{\beta} + \frac{\ln \frac{1}{\eta}}{n\beta}] \leq \eta, \quad (13)$$

where RHS of (13) is due to the number of $(\mathbf{y}, \mathbf{x})$ satisfying the above constraint is bounded by $e^{nI(\mathbf{x};\mathbf{y})}$ asymptotically [18].

Applying the Craig-Bernstein inequality to $\bar{U}$ with $\tau = \ln\frac{1}{\eta}$, $\epsilon = \beta$, and $c = h\beta$ and the fact that $\mathbb{E}\bar{U} = \mathbb{E}[\|\mathbf{y}-\mathbf{x}\|_2^2] - \sigma^2$, we have

$$Pr[\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : \bar{U} \geq (1+\alpha)\left(\mathbb{E}[\|\mathbf{y}-\mathbf{x}\|_2^2] - \sigma^2\right) + \frac{\ln\frac{1}{\eta}}{n\beta}] \leq \eta. \tag{14}$$

Together with (13) and (14), we have

$$Pr[\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : (1-\alpha)\|\mathbf{x}-\mathbf{x}^*\|_2^2 \geq (1+\alpha)\left(\mathbb{E}[\|\mathbf{y}-\mathbf{x}\|_2^2] - \sigma^2\right) + \frac{I(\mathbf{x};\mathbf{y})}{\beta} + \frac{2\ln\frac{1}{\eta}}{n\beta}] \leq 2\eta. \tag{15}$$

As $\alpha > 0$, it also holds that

$$Pr[\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : (1-\alpha)\|\mathbf{x}-\mathbf{x}^*\|_2^2 \geq (1+\alpha)\left(\mathbb{E}[\|\mathbf{y}-\mathbf{x}\|_2^2] - \sigma^2 + \frac{I(\mathbf{x};\mathbf{y})}{\beta}\right) + \frac{2\ln\frac{1}{\eta}}{n\beta}] \leq 2\eta, \tag{16}$$

and

$$Pr[\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : (1-\alpha)\|\mathbf{x}-\mathbf{x}^*\|_2^2 \geq (1+\alpha)\min_{p(\mathbf{x}),p(\mathbf{x}|\mathbf{y})}\left(\mathbb{E}[\|\mathbf{y}-\mathbf{x}\|_2^2] - \sigma^2 + \frac{I(\mathbf{x};\mathbf{y})}{\beta}\right) + \frac{2\ln\frac{1}{\eta}}{n\beta}] \leq 2\eta. \tag{17}$$

Define $R(\beta) := \min_{p(\mathbf{x}),p(\mathbf{x}|\mathbf{y})}[I(\mathbf{x};\mathbf{y}) + \beta\mathbb{E}(\|\mathbf{y}-\mathbf{x}\|_2^2)]$, we have

$$Pr[\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : (1-\alpha)\|\mathbf{x}-\mathbf{x}^*\|_2^2 \geq (1+\alpha)(\frac{R(\beta)}{\beta} - \sigma^2) + \frac{2\ln\frac{1}{\eta}}{n\beta}] \leq 2\eta, \tag{18}$$

that is

$$Pr[\forall \mathbf{x} \sim p^*(\mathbf{x}|\mathbf{y}), \mathbf{y} : \|\mathbf{x}-\mathbf{x}^*\|_2^2 \geq \frac{(1+\alpha)}{(1-\alpha)}(\frac{R(\beta)}{\beta} - \sigma^2) + \frac{2\ln\frac{1}{\eta}}{n\beta(1-\alpha)}] \leq 2\eta. \tag{19}$$

C.2 Proof to (9)

The technique is based on [19]. That is if X is a random variable such that $\mathbb{E}[X] < \infty$, then integration by parts yields

$$\mathbb{E}[X] \leq \int_0^\infty Pr[X \geq t]dt. \tag{20}$$

Let $X := \|\mathbf{x}-\mathbf{x}^*\|_2^2 - \frac{(1+\alpha)}{(1-\alpha)}(\frac{R(\beta)}{\beta} - \sigma^2)$, $\eta = e^{-\frac{tn\beta(1-\alpha)}{2}}$ such that $\frac{2\ln\frac{1}{\eta}}{n\beta(1-\alpha)} = t$. Based on (20), we have

$$\begin{aligned}
\mathbb{E}[\|\mathbf{x}-\mathbf{x}^*\|_2^2 - \frac{(1+\alpha)}{(1-\alpha)}(\frac{R(\beta)}{\beta} - \sigma^2)] &\leq \int_0^\infty Pr[\|\mathbf{x}-\mathbf{x}^*\|_2^2 - \frac{(1+\alpha)}{(1-\alpha)}(\frac{R(\beta)}{\beta} - \sigma^2) \geq t]dt, &(21)\\
&\leq 2\int_0^\infty \exp^{-\frac{tn\beta(1-\alpha)}{2}} dt, &(22)\\
&= \frac{4}{\beta(1-\alpha)n}.
\end{aligned}$$

□

D Proof to Theorem 2

The proof to Theorem 2 is based on [20, 21], and for completeness we present it here.

Proof. First, we address the following theorem and then complete the proof of Theorem 2.

Theorem 3. *Let* $\rho(\mathbf{y},\mathbf{x}) = -\log_2 p_{\mathbf{e}}(\mathbf{z})$, $\mathbf{z} = \mathbf{x} - \mathbf{y}$, $d = H(\mathbf{e})$, *and* $R(d)$ *be defined as follows:*

$$R(d) := \min_{\{p(\mathbf{x}),p(\mathbf{x}|\mathbf{y}):\mathbb{E}[\rho(\mathbf{x},\mathbf{y})]\leq d\}} I(\mathbf{x};\mathbf{y}). \tag{23}$$

Then if $\mathbf{z}$ *and* $\mathbf{e}$ *follow the same distribution,* $R(d) = H(\mathbf{x}) - H(\mathbf{e})$.

Proof. Let the distributions associated with $(\mathbf{x},\mathbf{y})$, $(P^*(\mathbf{x}|\mathbf{y}), P^*(\mathbf{x}))$, achieve $R(d)$. Then

$$\begin{aligned}
R(d) &= I(\mathbf{x};\mathbf{y}), && (24)\\
&= H(\mathbf{x}) - H(\mathbf{x}|\mathbf{y}),\\
&= H(\mathbf{x}) - H(\mathbf{x}-\mathbf{y}|\mathbf{y}),\\
&\geq H(\mathbf{x}) - H(\mathbf{x}-\mathbf{y}), && (25)\\
&= H(\mathbf{x}) - H(\mathbf{z}), && (26)\\
&= H(\mathbf{x}) - H(\mathbf{e}), && (27)
\end{aligned}$$

where

(24) $H(\mathbf{x})$ and $H(\mathbf{x}|\mathbf{y})$ are defined upon $(p^*(\mathbf{x}|\mathbf{y}), p^*(\mathbf{y}))$;

(25) is due to the fact that conditioning reduces entropy, i.e., $H(\mathbf{x}|\mathbf{y}) \leq H(\mathbf{x})$;

(26) $H(\mathbf{x})$ is defined upon $P^*(\mathbf{x})$ and $H(\mathbf{z})$ is defined upon $p_{\mathbf{e}}(\mathbf{z})$;

(27) is due to the assumption that $\mathbf{z}$ and $\mathbf{e}$ follow the same distribution.

Next, we prove the above lower abound is achievable: As $\mathbb{E}[\rho(\mathbf{x}^*,\mathbf{y})] = H(\mathbf{e}) = H(\mathbf{z})$, the derivations from (24) to (27) hold for the distribution associated with $(\mathbf{x}^*,\mathbf{y})$, i.e., $p(\mathbf{y}|\mathbf{x}^*)$ and $p(\mathbf{x}^*)$. Therefore, if $(p^*(\mathbf{x}|\mathbf{y}), p^*(\mathbf{x}))$ and $(p(\mathbf{y}|\mathbf{x}^*), p(\mathbf{x}^*))$ are identical (see below for the identical proof), then the above lower bound is achievable. This finishes this part of proof as $R(d)$ is defined in (23). □

Now, we prove Theorem 2: $R(d)$ is achieved by both $(p^*(\mathbf{x}|\mathbf{y}), p^*(\mathbf{y}))$ and $(p(\mathbf{y}|\mathbf{x}), p(\mathbf{x}^*))$. As $R(d)$ a strictly convex function of associated conditional distributions [6, Section 13.7, pp. 362], there is only one minimizer for a strictly convex function [22]. Therefore, $(p^*(\mathbf{x}|\mathbf{y}), p^*(\mathbf{x}))$ and $(p(\mathbf{y}|\mathbf{x}), p(\mathbf{x}^*))$ must be identical and this leads to $H(\mathbf{x}) \to H(\mathbf{x}^*)$ with proper β chosen such that $d = H(\mathbf{z}) = H(\mathbf{e})$. □

E Experimental details

We performed all our experiments on a single GPU. The running time of BDBA is comparable to DIP.

BDBA is in PyTorch [23]. SBM is from https://github.com/cloneofsimo/minDiffusion. The image size of dataset is 128×128. We set T to 1000 for Langevin MCMC. The initialization of $\mathbf{x}_0$ is from $\mathbf{y}$. Adam [24] optimizer with an initial learning rate of $1e^{-4}$ is used.

DIP is from the open-source implementation: https://github.com/DmitryUlyanov/deep-image-prior. For DIP, we use the U-NET [25] with skip connections. Adam optimizer with an initial learning rate of $1e^{-3}$ is used. The loss function is MSE. The DIP optimization iterations is 1000.