

NF-ULA

Cai, Ziruo; Tang, Junqi; Mukherjee, Subhadip; Li, Jinglai; Schönlieb, Carola-Bibiane; Zhang, Xiaoqun

DOI:

[10.1137/23M1581807](https://doi.org/10.1137/23M1581807)

License:

Creative Commons: Attribution (CC BY)

Document Version

Peer reviewed version

Citation for published version (Harvard):

Cai, Z, Tang, J, Mukherjee, S, Li, J, Schönlieb, C-B & Zhang, X 2024, 'NF-ULA: Normalizing Flow-Based Unadjusted Langevin Algorithm for Imaging Inverse Problems', *SIAM Journal on Imaging Sciences*, vol. 17, no. 2, pp. 820-860. <https://doi.org/10.1137/23M1581807>

[Link to publication on Research at Birmingham portal](#)

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

NF-ULA: Normalizing flow-based unadjusted Langevin algorithm for imaging inverse problems

Ziruo Cai*, Junqi Tang†, Subhadip Mukherjee‡, Jinglai Li§, Carola-Bibiane Schönlieb¶, and Xiaoqun Zhang||

Abstract. Bayesian methods for solving inverse problems are a powerful alternative to classical methods since the Bayesian approach offers the ability to quantify the uncertainty in the solution. In recent years, data-driven techniques for solving inverse problems have also been remarkably successful, due to their superior representation ability. In this work, we incorporate data-based models into a class of Langevin-based sampling algorithms for Bayesian inference in imaging inverse problems. In particular, we introduce NF-ULA (Normalizing Flow-based Unadjusted Langevin algorithm), which involves learning a *normalizing flow* (NF) as the image prior. We use NF to learn the prior because a tractable closed-form expression for the log prior enables the differentiation of it using *autograd* libraries. Our algorithm only requires a normalizing flow-based generative network, which can be pre-trained independently of the considered inverse problem and the forward operator. We perform theoretical analysis by investigating the well-posedness and non-asymptotic convergence of the resulting NF-ULA algorithm. The efficacy of the proposed NF-ULA algorithm is demonstrated in various image restoration problems such as image deblurring, image inpainting, and limited-angle X-ray computed tomography (CT) reconstruction. NF-ULA is found to perform better than competing methods for severely ill-posed inverse problems.

Key words. Bayesian inference, Langevin algorithms, normalizing flows, inverse problems.

MSC codes. 62F15, 49N45, 92C55

1. Introduction. Imaging inverse problems can be formulated as $y = Ax + n$, where $y \in \mathbb{R}^m$ is the indirect noisy observation, $A : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is the observation operator, n is the measurement noise, and $x \in \mathbb{R}^d$ represents the unknown image that one aims to recover. In the classical variational framework, the reconstruction problem is formulated as the minimization of an energy functional $J(x) = L(y, Ax) + \alpha g(x)$, where L measures data-consistency and g is a regularizer that penalizes undesirable images. Following the surge of deep learning, data-driven regularization methods have become ubiquitous in imaging inverse problems [7, 10, 72], leading to state-of-the-art results which significantly outperform classical hand-crafted regularization schemes such as the total-variation [13] or sparsity-based regularizers (see [10] and references therein). Starting from the plug-and-play methods [96] which combine proximal-splitting optimization algorithms [17] with learned denoisers [45, 83, 103], researchers have made considerable progress in this direction. Current popular trends in this

*School of Mathematical Sciences, Shanghai Jiao Tong University, China (sjtu_caiziruo@sjtu.edu.cn).

†School of Mathematics, University of Birmingham, UK (j.tang.2@bham.ac.uk).

‡Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology (IIT) Kharagpur, India. (smukherjee@ece.iitkgp.ac.in).

§School of Mathematics, University of Birmingham, UK (j.li.10@bham.ac.uk).

¶Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK (cbs31@cam.ac.uk).

||School of Mathematical Sciences, MOELSC and Institute of Natural Sciences, Shanghai Jiao Tong University, China(xqzhang@sjtu.edu.cn).

line of research include the studies in improving practical performances and theoretical guarantees [33,38,47,81,90,94], the development of deep unrolling networks [1,67], deep equilibrium models [34], the studies on the image prior by specific networks structures [59], the extension of generative models in imaging applications [8,73,87,99], operator regularization methods [77], learning explicitly the regularization functional such as a gradient-step denoiser [42], total deep variation [53], adversarial regularizers [63,69,78] and the learned convex regularizer [70] with input-convex neural networks [5].

While the previously mentioned approaches treat x deterministically, another alternative framework for solving inverse problems is to do it within a Bayesian setting [46,93,95]. Different from the functional-analytic methods, Bayesian methods model the image x as a random variable and usually seek to approximate the posterior distribution $p(x|y)$ based on Bayes' formula. The methods based on Bayesian inference can not only give a point estimator (e.g., the maximum a posteriori probability (MAP) estimator) but also describe the uncertainty in the solution in a probabilistic way in terms of variance and credible intervals. The capability of uncertainty quantification is particularly helpful for decision-making and reliability assessment. Typical examples of Bayesian imaging schemes include the classical approach using the total variation prior [62,76], the works on Markov random fields [11], and more recently the patch-based models [2,41,102,105].

In Bayesian inference, one explores the posterior distribution to generate samples from it, typically using the Markov Chain Monte Carlo (MCMC) methods [32]. Among these sampling algorithms, the Langevin Monte Carlo (LMC) algorithms [71,80], also referred to as the *Unadjusted Langevin Algorithms* (ULA), stand out as an increasingly popular tool, since they bridge the gap between theoretical guarantees of nonasymptotic convergence analysis [20,22,28] and practical performance [29,56]. Note that ULA is subject to bias related to the stepsize, ULA can also be modified into Metropolis-adjusted Langevin algorithm (MALA) [80], a non-biased version, by adding a Metropolis-Hastings (MH) accept-reject step. Apart from the MCMC-based methods, there are also other kinds of sampling methods worth mentioning: methods based on variational inference [12,40,61] posit a family of densities and then attempt to find a member of that family which is close to the target density. Variational auto-encoders (VAEs) [52] approximate the posterior by learning deep encoders and decoders. Generative adversarial networks (GAN) [19,35] learn the generator to sample from the training distribution through adversarial learning. More recently, diffusion models [39,88,101] have been shown to be a powerful tool for image generation. They learn the target distribution by transforming an image into a Gaussian noise and then by reversing the noising process.

In recent years, the theoretical analysis and nonasymptotic convergence of ULA [20,28] have opened a new direction of research. Besides convex and smooth potentials [20,21,27,28], ULA for non-convex or non-smooth potentials has also seen great progress. While ULA requires evaluating the score, ULA for non-smooth distributions [29,58,64,68,76] draw samples from a smoothed proxy by borrowing the tools such as proximity operators from non-smooth optimization literature, or consider potential splitting [85]. For non-convex potentials, ULA also has convergence guarantees [14,22,31,65] if some conditions, (e.g., contractivity condition on the drift) are satisfied.

Incorporating data-based approaches into classical algorithms is a trending topic in ULA and Bayesian methods for solving inverse problems. More specifically, one aims to utilize

79 an over-parameterized model learned on given data, such as a neural network, instead of
 80 [handcrafted](#) prior. Recently, Langevin Monte Carlo using Plug and Play Prior (PnP-ULA) [56]
 81 was shown to yield promising results for Bayesian imaging problems. PnP-ULA leverages an
 82 implicit image prior learned via a Lipschitz-continuous image denoiser [84]. Since the true
 83 image prior is not assumed to be convex or smooth, PnP-ULA convergence was established
 84 for non-convex potentials.

85 Besides PnP priors [96], normalizing flow (NF)-based approaches [25, 74, 79] also lead
 86 to impressive performance on imaging problems [25, 50] and have the potential of learning
 87 the prior in the Bayesian imaging framework. In this work, we attempt to integrate an
 88 image prior that is learned by NF into the Langevin algorithms. Notably, the resulting
 89 negative log posterior in our case is non-convex. To make the model well-defined in the
 90 Bayesian setting and to ensure that the algorithm is numerically stable, we make minor
 91 changes to the standard ULA to add [a regularization on the posterior](#), akin to PnP-ULA [56].
 92 As some studies of normalizing flows have shown [25, 50, 74, 79], training a normalizing flow
 93 prior for natural images generally requires utilizing larger networks, larger training dataset,
 94 more computational resources and more time than training a PnP denoiser, our proposed
 95 method is more efficient if the normalizing flow prior is pre-trained and available.

96 The idea of interlacing NF with MCMC algorithms has been considered previously in
 97 the literature, but these methods had significant conceptual differences from our approach.
 98 For instance, [100] proposed stochastic NF, an arbitrary sequence of deterministic invertible
 99 functions and stochastic sampling blocks, to sample from target density. The authors of [36, 91]
 100 considered stochastic NF from a Markov chain point of view and replaced the transition
 101 densities with general Markov kernels. [15] utilized NF to sample from the target distribution in
 102 the latent domain before transporting it back to the target domain relying on MALA. There are
 103 some studies combining other generative models with non-Langevin Monte Carlo algorithms,
 104 e.g., [16] introduced a stochastic PnP sampling algorithm leveraging variable splitting to
 105 efficiently sample from a posterior distribution using diffusion-based generative models [23].
 106 To summarize, all the above mentioned approaches are different from ours, mainly because
 107 they do not directly utilize the log gradient density of NF in Langevin algorithms.

108 **1.1. Our contributions.** The main contributions of this work are:

- 109 1. We propose NF-ULA, a novel framework of sampling by Langevin Monte Carlo-based
 110 algorithms while leveraging a pre-trained normalizing flow induced prior. Since both
 111 the density and the log gradient of the density of normalizing flows can be evaluated,
 112 NF-ULA can potentially be extended to a Metropolis-adjusted version.
- 113 2. We give a sufficient condition to ensure the Lipschitz gradient of the log density of the
 114 normalizing flows since the Lipschitz gradient is one of the most essential conditions
 115 to guarantee the convergence of ULA. This might also be useful in the future when an
 116 NF-based prior is used in methods other than Langevin algorithms.
- 117 3. We show that the Bayesian solution of NF-ULA is well-defined and well-posed and
 118 establish that NF-ULA admits an [unique](#) invariant distribution. We also give a non-
 119 asymptotic bound on the bias.
- 120 4. We demonstrate that NF-ULA yields high-quality results in applications such as image
 121 deblurring, image inpainting, and limited-angle X-ray computed tomography (CT) re-

122 construction. For more ill-posed problems, NF-UULA demonstrates stronger regulariza-
 123 tion than competing methods. We also provide experimental evidence that enhanced
 124 training of the NF prior results in improved sampling and reconstruction, especially
 125 for severely ill-posed problems (such as limited-angle CT).

126 The rest of the paper is organized as follows: Sec. 2 gives a brief review of both Langevin
 127 Monte Carlo and normalizing flow, leading to the proposed NF-UULA method. Sec. 3 presents
 128 a theoretical analysis of the Bayesian solution obtained using NF-UULA. In Sec. 4, we evaluate
 129 NF-UULA on image deblurring, image inpainting, and limited-angle CT reconstruction. Final
 130 conclusions are summarized in Sec. 5. The proofs and extra experiments are in the Appendix.

131 **2. Mathematical background and the proposed method.** We begin by giving some back-
 132 ground on Langevin Monte Carlo (LMC) algorithms and normalizing flow. Subsequently, we
 133 propose NF-UULA, an LMC algorithm that utilizes a pre-trained normalizing flow network.

134 **2.1. LMC for Non-smooth Potentials.** In Bayesian inference, there is a broad class of
 135 problems where we seek to draw samples $\{X_k\}_{k=1}^K$, $X_k \in \mathbb{R}^d$, from a target posterior distribu-
 136 tion $p(x|y)$, given the observation $y \in \mathbb{R}^m$. Using Bayes' formula, we have that

$$137 \quad (2.1) \quad p(x|y) = \frac{p(y|x)p(x)}{\int p(y|\tilde{x})p(\tilde{x})d\tilde{x}}.$$

138 Under some assumptions on the likelihood $p(y|x)$ and the prior $p(x)$, the posterior distribu-
 139 tion $p(x|y)$ is well-posed; meaning that it is well-defined ($\int p(y|\tilde{x})p(\tilde{x})d\tilde{x}$ is finite), unique,
 140 and varies continuously in y with respect to appropriate distance metrics for probability dis-
 141 tributions [55, 89]. The well-known LMC approach [71, 80], also referred to as the *unadjusted*
 142 *Langevin algorithm* (ULA), can efficiently sample from $p(x|y)$ using the following Markov
 143 chain:

$$144 \quad (2.2) \quad \begin{aligned} X_{k+1} &= X_k + \delta \nabla \log p(X_k|y) + \sqrt{2\delta} Z_{k+1} \\ &= X_k + \delta \nabla \log p(y|X_k) + \delta \nabla \log p(X_k) + \sqrt{2\delta} Z_{k+1}, \end{aligned}$$

145 where $\{Z_k\}_k \sim \mathcal{N}(0, I^d)$ is a family of i.i.d. standard Gaussian random variables. The ULA
 146 approach in (2.2) is based on the Euler-Maruyama (EM) discretization with step-size δ of the
 147 over-damped Langevin stochastic differential equation (SDE) given by

$$148 \quad (2.3) \quad dX_t = \nabla \log p(X_t|y) dt + \sqrt{2} dB_t,$$

149 where B_t is a Brownian motion. It has been shown in [20, 28] that when $-\log p(x|y)$ is contin-
 150 uously differentiable and has Lipschitz gradient, the convergence of ULA can be guaranteed
 151 if the convexity of $-\log p(x|y)$ [20] or contractivity in the tails [28] is satisfied. The conver-
 152 gence is subject to a bias related to the step-size δ . In general, smaller δ leads to a smaller
 153 bias and larger δ leads to faster convergence of the Markov Chain. The non-asymptotic
 154 bias and convergence analysis of ULA have remained relatively under-explored until the last
 155 few years [20, 21, 27, 28]. Notably, the bias of ULA in (2.2) can be removed by adding a
 156 Metropolis-Hastings (MH) accept-reject step, leading to the so-called Metropolis-adjusted
 157 Langevin algorithm (MALA) [80]. In this paper, we will focus on ULA without any MH
 158 adjustments.

159 When the potential $-\log p(x)$ is convex but non-smooth, [29] uses a smooth proxy utilizing
 160 the Moreau envelope $U^{(\lambda)}(x)$ of $U(x) = -\log p(x)$ in (2.2). The Moreau envelope $U^{(\lambda)}(x)$ and
 161 the proximity operator $\text{prox}_{\lambda,U}$ of $U(x)$ are defined as

$$162 \quad U^{(\lambda)}(x) := \inf_{z \in \mathbb{R}^d} \left(U(z) + \frac{1}{2\lambda} \|x - z\|_2^2 \right), \quad \text{and} \quad \text{prox}_{\lambda,U}(x) := \arg \min_{z \in \mathbb{R}^d} \left(U(z) + \frac{1}{2\lambda} \|x - z\|_2^2 \right).$$

163 For a convex function U , $\text{prox}_{\lambda,U}(x)$ is unique and well-defined.

164 Since the Moreau envelope $U^{(\lambda)}(x)$ is always continuously differentiable [9, 18] even if
 165 $U(x)$ is not, the authors of [29] replace $\nabla U(x)$ by $\nabla U^{(\lambda)}(x) = (x - \text{prox}_{\lambda,U}(x)) / \lambda$, resulting
 166 in Moreau-Yoshida regularized ULA (referred to as MYULA), which requires the proximal
 167 operator of $U(x)$ in each iteration of (2.2).

168 In a more general case where the prior $p(x)$ is not available in closed form, the authors
 169 of [56] propose a plug-and-play (PnP) denoising-based approach for learning the prior [84, 96].
 170 This is achieved by training a Lipschitz-continuous Gaussian denoiser $D_\varepsilon(x)$. More precisely,
 171 $D_\varepsilon(x)$ is trained on a given dataset $\{x_n\}_{n=1}^N$ by learning to remove Gaussian noise of zero-
 172 mean and ε variance added to the clean images x_n , which are i.i.d. samples of $p(x)$. The ideal
 173 minimum mean-squared-error (MMSE) denoiser takes the form

$$174 \quad (2.4) \quad D_\varepsilon(x) = (2\pi\varepsilon)^{-d/2} \int_{\mathbb{R}^d} \tilde{x} \exp[-\|x - \tilde{x}\|^2 / (2\varepsilon)] p(\tilde{x}) d\tilde{x}.$$

176 The noisy data follows the Gaussian-smoothed prior

$$177 \quad p_\varepsilon(x) = (2\pi\varepsilon)^{-d/2} \int_{\mathbb{R}^d} \exp[-\|x - \tilde{x}\|_2^2 / (2\varepsilon)] p(\tilde{x}) d\tilde{x},$$

178 which is the convolution of the non-explicit prior $p(x)$ with a Gaussian smoothing kernel.
 179 Similar to the Moreau envelope [9, 18], p_ε is always differentiable and satisfies Tweedie's
 180 identity [30]: $\varepsilon \nabla \log p_\varepsilon(x) = D_\varepsilon(x) - x$. While computing $\nabla \log p(x)$ could be intractable, one
 181 can use $\nabla \log p_\varepsilon(x)$ as a surrogate in (2.2), leading to the PnP-ULA approach [56]:

$$182 \quad (\text{PnP-ULA}) : \quad X_{k+1} = X_k + \delta \nabla \log p(y|X_k) \\ + \frac{\delta\alpha}{\varepsilon} (D_\varepsilon(X_k) - X_k) + \frac{\delta}{\lambda} (\Pi_C(X_k) - X_k) + \sqrt{2\delta} Z_{k+1},$$

183 where $\alpha > 0$ is a regularization parameter associated with the PnP prior and $\{Z_k\}_k$ are i.i.d.
 184 drawn from $\mathcal{N}(0, I^d)$. A projection $\Pi_C(X_k)$ onto a convex and compact set C is added in each
 185 iteration to enable the theoretical analysis for PnP-ULA. $\lambda > 0$ is a parameter associated with
 186 the operator $\Pi_C - \text{Id}$. Moreover, the Lipschitz continuity of the denoiser $D_\varepsilon(x)$ is required for
 187 convergence. A detailed convergence analysis of (2.5) is available in [56].

188 **2.2. Normalizing Flow.** Similar to a PnP prior, a flow-based model can also serve as a
 189 prior. A flow-based model seeks to express $x \in \mathbb{R}^d$ as

$$190 \quad (2.6) \quad x = T(z),$$

191 where $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is an invertible transformation applied to $z \in \mathbb{R}^d$, where $z \sim q_z(z)$. Here,
 192 $q_z(z)$ is the input (or, latent) distribution of the flow-based model and is generally chosen to
 193 be a distribution that can be sampled easily, such as a multivariate Gaussian [51, 54, 74, 79].
 194 Apart from $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ being invertible, both T and T^{-1} must be differentiable [74, 79].
 195 The flow-based model is also called *normalizing flow* since T^{-1} implicitly transforms $q(x)$, the
 196 distribution of x , into a normal distribution. In practice, T is typically implemented with an
 197 invertible neural network [25, 50]. By a change of variables in (2.6), the distribution of x can
 198 be written as

$$199 \quad (2.7) \quad q(x) = q_z(z) |\det J_T(z)|^{-1} = q_z(T^{-1}(x)) |\det J_{T^{-1}}(x)|,$$

200 where $z = T^{-1}(x)$ and $J_T(z)$ is the $d \times d$ Jacobian matrix of T . Many normalizing flows
 201 [50, 51, 74, 75, 79] use specific network architectures such that T^{-1} is a triangular mapping, that
 202 is, the Jacobian $J_{T^{-1}}(x)$ is a triangular matrix, which simplifies the calculation of $|\det J_{T^{-1}}(x)|$.
 203 Note that T is used to generate x from z , and T^{-1} is needed for evaluating the density $q(x)$.

204 Some works on normalizing flow use coupling layers in the network to make T^{-1} a tri-
 205 angular mapping [24, 25, 50, 51, 75]. Denote $G(x) = T^{-1}(x)$, $G : \mathbb{R}^d \rightarrow \mathbb{R}^d$. Let x_j be the
 206 j -th element of x and $x_{<j}$ be the elements before x_j , i.e. x_1, \dots, x_{j-1} . Then, for one-layer
 207 network, [44] summarizes the coupling layer-based flows as $G_j(x_j, x_{<j}) = \varphi_j(x_{<j})x_j + \eta_j(x_{<j})$,
 208 where G_j is the j -th element of the vector $G(x)$ and the functions φ_j and η_j map $x_{<j}$ to a
 209 real number. The Jacobian $J_G(x)$ is triangular since G_j only depends on x_j and $x_{<j}$.

210 Assume that the unknown prior distribution that we aim to learn is $p(x)$. Then, the
 211 forward KL divergence between the target distribution $p(x)$ and the output distribution $q(x)$
 212 of the NF model [54, 74, 79] can be written as

$$213 \quad (2.8) \quad D_{\text{KL}}(p, q) = -\mathbb{E}_{p(x)} [\log q(x)] + \text{const.} \\ 214 \quad \quad \quad = -\mathbb{E}_{p(x)} [\log q_z(T^{-1}(x)) + \log |\det J_{T^{-1}}(x)|] + \text{const.}$$

216 When the transformation T is parameterized by an invertible neural network T_θ with param-
 217 eters $\theta \in \Theta$, we denote the parameterized density of x as $q_\theta(x)$ and the optimization problem
 218 of learning T_θ reads:

$$219 \quad (2.9) \quad \min_{\theta \in \Theta} D_{\text{KL}}(p, q_\theta).$$

220 Given samples $\{x_n\}_{n=1}^N$ drawn i.i.d. from $p(x)$, we can estimate the expectation in (2.8) by
 221 Monte Carlo averaging over the training samples $\{x_n\}_{n=1}^N$. Correspondingly, the loss function
 222 for training the NF model becomes

$$223 \quad (2.10) \quad \mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \left(\log q_z(T_\theta^{-1}(x_i)) + \log \left| \det J_{T_\theta^{-1}}(x_i) \right| \right) + \text{const.} \\ 224$$

225 Generally, it is reasonable to assume that the data samples $\{x_i\}_i^N$ lie within a compact set
 226 $C_R \subset \mathbb{R}^d$. In particular, when the flow-based model is learned on imaging data, it is common
 227 to set $C_R = [0, 1]^d$. Knowing the set where the data samples lie will give us the intuition to

228 select some parameters in the next section. From the numerical observations, the networks also
 229 partially know C_R while trained from the data - the knowledge of C_R is implicitly encapsulated
 230 in a well-trained flow model, meaning that most generated samples using a well-trained NF
 231 model fall within C_R .

232 **2.3. ULA with NF-prior** . In this section, we propose a framework for sampling using
 233 the LMC algorithm based on a pre-trained normalizing flow network. Given data samples
 234 $\{x_n\}_{n=1}^N$ drawn i.i.d. from $p(x)$, one can approximate $p(x)$ by learning a flow-based model
 235 $x = T_\theta(z)$, with output distribution $q_\theta(x) = q_z(T_\theta^{-1}(x)) \left| \det J_{T_\theta^{-1}}(x) \right|$. Once $q_\theta(x)$ is learned,
 236 $\log q_\theta(x)$ is always differentiable since T_θ and T_θ^{-1} are differentiable. By replacing $p(x)$ with
 237 $q_\theta(x)$ in (2.2), the ULA scheme boils down to

$$238 \quad X_{k+1} = X_k + \delta \nabla \log p(y|X_k) + \delta \nabla \log q_\theta(X_k) + \sqrt{2\delta} Z_{k+1}.$$

239 Since convexity of $-\log q_\theta(x)$ and the Lipschitz continuity of its gradient are not guaranteed to
 240 be satisfied, one does not yet have the sufficient conditions to infer convergence and numerical
 241 stability similar to the cases in [20, 28]. In this work, we follow [56] to impose a projection
 242 $\Pi_C(X_k)$ onto a convex and compact set C to ensure that the posterior distribution is well-
 defined and propose the resulting NF-ULA algorithm (c.f. Algorithm 2.1). The parameter

Algorithm 2.1 Normalizing Flow-based Unadjusted Langevin algorithm (NF-ULA)

Input: $y \in \mathbb{R}^m$, $X_0 \in \mathbb{R}^d$, $\alpha > 0$, $\lambda > 0$, $K \in \mathbb{N}$, $C \subset \mathbb{R}^d$

L_y : Lipschitz constant of $\nabla \log p(y|x)$.

L : Lipschitz constant of $\nabla \log q_\theta(x)$.

Output: $\{X_k\}_{k=1}^K$

Set: $k = 0$, $\delta < (1/6)(L_y + \alpha L + 1/\lambda)^{-1}$.

Initialize X_0 according to the considered problems.

while $k < K$ **do**

$Z_{k+1} \sim \mathcal{N}(0, I^d)$

$X_{k+1} = X_k + \delta \nabla \log p(y|X_k) + \delta \alpha \nabla \log q_\theta(X_k) + \frac{\delta}{\lambda} (\Pi_C(X_k) - X_k) + \sqrt{2\delta} Z_{k+1}$

$k = k + 1$

end while

243 $\alpha > 0$ controls how strongly the regularization of q_θ is imposed and λ controls the amount of
 244 the projection $(\Pi_C - \text{Id})$ enforced. Theoretical analysis of NF-ULA is presented in Sec. 3, while
 245 in Sec. 4, we provide some general guidelines for selecting the hyper-parameters involved in
 246 NF-ULA. One can efficiently compute $\nabla \log q_\theta(x)$ using the automatic differentiation libraries
 247 in the standard deep learning frameworks (such as PyTorch).

249 **Remark:** Algorithm 2.1 only requires evaluating the $\nabla \log q_\theta(x)$ and its Lipschitz constant.
 250 Our theoretical analysis in Sec. 3 depends on the properties of $q_\theta(x)$ and holds even when q_θ
 251 does not arise from a normalizing flow. This is essential since in our CT experiments in Sec.
 252 4.3, we utilize *patchNR* [3], a normalizing flow-based regularizer which cannot generate x by
 253 (2.6) but is able to evaluate the log gradient $\nabla \log q_\theta(x)$. Moreover, since $q_\theta(x)$ can also be

254 evaluated given x , Algorithm 2.1 can be extended to a Metropolis-adjusted version by adding
 255 an accept-reject step. We leave this as a possible future work.

256 It is imperative to understand why the projection $(\Pi_C - \text{Id})$ is necessary for the convergence
 257 of NF-UULA. Let $\iota_C^{(\lambda)}(x)$ be the λ -Moreau envelope [9] of the indicator function

$$258 \quad \iota_C(x) = \begin{cases} 0, & x \in C, \\ +\infty, & x \notin C. \end{cases}$$

259 Then, we have that

$$260 \quad \iota_C^{(\lambda)}(x) := \inf_{u \in \mathbb{R}^d} \left(\iota_C(u) + \frac{1}{2\lambda} \|x - u\|_2^2 \right) = \frac{1}{2\lambda} \|x - \Pi_C(x)\|_2^2,$$

$$\text{and } \nabla \iota_C^{(\lambda)}(x) = \frac{x - \text{Prox}_{\iota_C}(x)}{\lambda} = \frac{x - \Pi_C(x)}{\lambda},$$

261 where Π_C is the projection operator on the convex and compact (i.e., closed and bounded)
 262 set $C \subset \mathbb{R}^d$. Define $p_\lambda(x|y)$ as

$$263 \quad (2.11) \quad p_\lambda(x|y) = \frac{p(y|x)q_\theta^\alpha(x) \exp(-\iota_C^{(\lambda)}(x))}{\int_{\mathbb{R}^d} p(y|\tilde{x})q_\theta^\alpha(\tilde{x}) \exp(-\iota_C^{(\lambda)}(\tilde{x}))d\tilde{x}},$$

264 where the exponent $\alpha > 0$. The subscript λ in p_λ underlines the distinction from the posterior
 265 $p(x|y) = p(y|x)p(x)/p(y)$. Since θ is fixed if the NF is pre-trained and α is adjusted in the
 266 experiments section, they are not in the notation of p_λ for brevity. We show in Sec. 3.2 that
 267 $p_\lambda(x|y)$ is well-defined and therefore the projection term is necessary for NF-UULA, without
 268 which, $p(y|x)q_\theta^\alpha(x)/\int_{\mathbb{R}^d} p(y|\tilde{x})q_\theta^\alpha(\tilde{x})d\tilde{x}$ is not guaranteed to be well-defined in our settings.
 269 Denote by $\pi_{\lambda,y}$ (which we will write as π_λ for brevity) the probability measure whose density
 270 is $p_\lambda(x|y)$ in (2.11), i.e.,

$$271 \quad (2.12) \quad \frac{d\pi_\lambda}{d\pi_{\text{leb}}}(x) = p_\lambda(x|y),$$

272 where π_{leb} denotes the Lebesgue measure. Then, NF-UULA in Algorithm 2.1 is essentially
 273 equivalent to

$$274 \quad (2.13) \quad X_{k+1} = X_k + \delta \nabla \log p_\lambda(X_k|y) + \sqrt{2\delta} Z_{k+1}.$$

275 For standard ULA (2.2), the tail-decay condition ($-\log p(x|y)/\|x\|^2$ converges to a positive
 276 constant when $x \rightarrow \infty$) was first studied in [80,92] and was shown to imply the convergence of
 277 ULA. For NF-UULA (2.13), we want to emphasize that in most of our experiments, NF-UULA is
 278 convergent while using a well-pre-trained normalizing flow, even without the projection term.
 279 This is presumably because the density q_θ of a well-trained normalizing flow already satisfies
 280 the tail-decay condition [80,92] and most of the probability mass lies within C . For the cases
 281 where the normalizing flow is poorly trained, one should select a smaller C , without which
 282 the samples generated by NF-UULA will go far beyond our expected region (for imaging it is
 283 $C_R = [0, 1]^d$).

284 **3. Theoretical Analysis.** We define some useful notations for our analysis in Sec. 3.1 and
 285 present a theoretical analysis (well-definedness and well-posedness) of the Bayesian posterior
 286 $p_\lambda(x|y)$ in Sec. 3.2. Subsequently, we prove the convergence and non-asymptotic bias of
 287 NF-ULA in Sec. 3.3.

288 **3.1. Notations.** Denote by $\mathcal{B}(\mathbb{R}^d)$ the Borel σ -field of \mathbb{R}^d . Let μ be a probability measure
 289 on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ and f be a μ -integrable function. Denote by $\mu(f)$ the integral of f w.r.t. μ .
 290 For measurable $f : \mathbb{R}^d \rightarrow \mathbb{R}$ and measurable $V : \mathbb{R}^d \rightarrow [1, \infty)$, the V -norm of f is defined
 291 as $\|f\|_V = \sup_{\tilde{x} \in \mathbb{R}^d} |f(\tilde{x})|/V(\tilde{x})$. Let ξ be a finite signed measure on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$. Then the
 292 V -total variation norm of ξ is defined as

$$293 \quad (3.1) \quad \|\xi\|_V = \sup_{\|f\|_V \leq 1} \left| \int_{\mathbb{R}^d} f(\tilde{x}) d\xi(\tilde{x}) \right|.$$

294 Note that if $V = 1$, then $\|\cdot\|_V$ is the total variation $\|\cdot\|_{\text{TV}}$. $\|\cdot\|_V$ is weaker than $\|\cdot\|_{\text{TV}}$
 295 and from the definitions one has $\|\xi\|_{\text{TV}} \leq \|\xi\|_V$. $\|\cdot\|_V$ has been used a lot in the studies of
 296 ULA [22, 28, 56].

297 We denote by $\mathcal{P}(\mathbb{R}^d)$ the set of probability measures over $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ and for any
 298 $m \in \mathbb{N}$, $\mathcal{P}_m(\mathbb{R}^d) = \{\nu \in \mathcal{P}(\mathbb{R}^d) : \int_{\mathbb{R}^d} \|\tilde{x}\|^m d\nu(\tilde{x}) < +\infty\}$. Denote by \mathbf{W}_p as Wasserstein- p
 299 metric:

$$300 \quad (3.2) \quad \mathbf{W}_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu, \nu)} \mathbf{E}_{(x, y) \sim \gamma} \|x - y\|^p \right)^{1/p}, \quad p \geq 1,$$

301 where $\Gamma(\mu, \nu)$ is the set of all joint probability whose marginal distributions are μ and ν
 302 respectively.

303 Let $b \in C(\mathbb{R}^d, \mathbb{R}^d)$ where $C(\mathbb{R}^d, \mathbb{R}^d)$ stands for the set of all continuous functions from \mathbb{R}^d
 304 to \mathbb{R}^d . We consider the Markov chain $(X_k)_{k \in \mathbb{N}}$ given by the following recursion for any $k \in \mathbb{N}$
 305 and $x \in \mathbb{R}^d$, initialized at $X_0 = x$:

$$306 \quad X_{k+1} = X_k + \gamma b(X_k) + \sqrt{2\gamma} Z_k,$$

307 where $\gamma > 0$ and $\{Z_k : k \in \mathbb{N}\}$ a family of i.i.d. Gaussian random variables with zero mean and
 308 identity covariance matrix. We define its associated Markov kernel $R_\gamma : \mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d) \rightarrow [0, 1]$
 309 as follows for any $x \in \mathbb{R}^d$ and $A \in \mathcal{B}(\mathbb{R}^d)$:

$$310 \quad R_\gamma(x, A) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} \mathbf{1}_A(x + \gamma b(x) + \sqrt{2\gamma}z) \exp[-\|z\|^2/2] dz,$$

311 where $\mathbf{1}_A(x)$ is the function taking the value 1 if $x \in A$ or 0 if $x \notin A$. We say that R_γ satisfies a
 312 discrete drift condition $\mathbf{D}_d(W, \zeta_d, c)$ if there exist $\zeta_d \in [0, 1)$, $c \geq 0$ and a measurable function
 313 $W : \mathbb{R}^d \rightarrow [1, +\infty)$ such that for all $x \in \mathbb{R}^d$

$$314 \quad R_\gamma W(x) \leq \zeta_d W(x) + c,$$

315 where $R_\gamma W(x) := \int_{\mathbb{R}^d} R_\gamma(x, d\tilde{x}) W(\tilde{x})$. Note that this drift condition implies the existence
 316 of an invariant probability measure if R_γ is a Feller kernel and the level sets of W are compact,
 317 see [22] and Theorem 12.3.3 in [26].

318 Similarly, let $b \in C(\mathbb{R}^d, \mathbb{R}^d)$ such that for any $x \in \mathbb{R}^d$, the following SDE admits a unique
 319 strong solution

$$320 \quad (3.3) \quad \begin{aligned} d\mathbf{X}_t &= b(\mathbf{X}_t) dt + \sqrt{2} d\mathbf{B}_t, \\ \mathbf{X}_0 &= x, \end{aligned}$$

321 where $(\mathbf{B}_t)_{t \geq 0}$ is a d -dimensional Brownian motion. For any $x \in \mathbb{R}^d$ and $A \in \mathcal{B}(\mathbb{R}^d)$, equation
 322 (3.3) defines a Markov semi-group $(P_t)_{t \geq 0}$ by $P_t(x, A) = \mathbb{P}(\mathbf{X}_t \in A)$ where $(\mathbf{X}_t)_{t \geq 0}$ is the
 323 solution of (3.3) with $\mathbf{X}_0 = x$. For any $f \in C^2(\mathbb{R}^d, \mathbb{R})$, define the generator \mathcal{A} of $(P_t)_{t \geq 0}$ by
 324 $\mathcal{A}f = \langle \nabla f, b(x) \rangle + \Delta f$, where Δ is the Laplace operator. We say that $(P_t)_{t \geq 0}$ on $\mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d)$
 325 with extended infinitesimal generator $(\mathcal{A}, D(\mathcal{A}))$ (see e.g. [66] for the definition of $(\mathcal{A}, D(\mathcal{A}))$)
 326 satisfies a continuous drift condition $\mathbf{D}_c(W, \zeta, \beta)$ if there exist $\zeta > 0, \beta \geq 0$ and a measurable
 327 function $W : \mathbb{R}^d \rightarrow [1, +\infty)$ with $W \in D(\mathcal{A})$ such that for all $x \in \mathbb{R}^d$,

$$328 \quad \mathcal{A}W(x) \leq -\zeta W(x) + \beta.$$

329 This assumption is the continuous counterpart of the discrete drift condition $\mathbf{D}_d(W, \zeta_d, c)$,
 330 which will be used in Appendix A.7.

331 **3.2. Well-posedness of the Bayesian solution.** In this section, we first prove that the
 332 posterior distribution (2.11) is well-defined. Secondly, we prove the well-posedness for the
 333 Bayesian solution, i.e., the Lipschitz continuity of the posterior measure (2.12) with respect
 334 to changes in y . To start with, we give a lemma that will be used later.

335 **Lemma 3.1.** *Let $\lambda > 0$. For any convex and compact subset C of \mathbb{R}^d and for all $k \in \mathbb{N}$, it*
 336 *holds that*

$$337 \quad \int_{\mathbb{R}^d} \|x\|^k \exp\left(-\frac{\|x - \Pi_C(x)\|_2^2}{2\lambda}\right) dx < +\infty.$$

338 *Proof.* See Appendix A.1. ■

339 Lemma 3.1 implies that the integral of any polynomials multiplied by $\exp\left(-\iota_C^{(\lambda)}\right)$, where
 340 $\iota_C^{(\lambda)} = \frac{\|x - \Pi_C(x)\|_2^2}{2\lambda}$, is finite. To prove that $p_\lambda(x|y)$ and π_λ are well-defined, besides Lemma
 341 3.1, we need an assumption about the boundedness of the prior and the likelihood.

342 **Assumption 3.2.** The distribution learned by NF is bounded, i.e., $\sup_{x \in \mathbb{R}^d} q_\theta(x) < +\infty$. More-
 343 over, for any $y \in \mathbb{R}^m$, $\sup_{x \in \mathbb{R}^d} p(y|x) < +\infty$ and $p(y|\cdot) \in C^1(\mathbb{R}^d, (0, +\infty))$.

344 Since $q_\theta(x)$ is a distribution induced by normalizing flow and $q_\theta(x)$ is continuous on \mathbb{R}^d ,
 345 intuitively $\sup_{x \in \mathbb{R}^d} q_\theta(x)$ is bounded and Assumption 3.2 is easily satisfied. To give rigorous proof,
 346 we state the following proposition which assumes a similar triangular network architecture as
 347 mentioned in Sec. 2.2.

348 **Proposition 3.3.** Assume that the input distribution $q_z(z)$ to the normalizing flow network
 349 is the standard normal distribution. Assume that $T^{-1}(x) = G^{(k)} \circ \dots \circ G^{(1)}(x)$ is a composition
 350 of k coupling layers and each of the layer $G^{(i)} : \mathbb{R}^d \rightarrow \mathbb{R}^d, x^{(i)} \mapsto x^{(i+1)}$ is given by

$$351 \quad (3.4) \quad G_j^{(i)}(x_j^{(i)}, x_{<j}^{(i)}) = \varphi_j^{(i)}(x_{<j}^{(i)})x_j^{(i)} + \eta_j^{(i)}(x_{<j}^{(i)}), \quad j = 1, \dots, d.$$

352 Denote $x^{(1)} = x$ and $x^{(k+1)} = z$. If $\varphi_j^{(i)}$ s are bounded, then $\log q_\theta(x)$ is upper bounded on \mathbb{R}^d .

353 *Proof.* See Appendix A.2. ■

354 Using Lemma 3.1, we can then prove that the normalizing constant in the expression for
 355 $p_\lambda(x|y)$ in (2.11) is finite.

356 **Corollary 3.4.** Suppose Assumption 3.2 holds. Let $\lambda > 0$. Then, for any convex and com-
 357 pact set C and $\alpha > 0$, we have

$$358 \quad \int_{\mathbb{R}^d} p(y|x) q_\theta^\alpha(x) \exp\left(-\frac{\|x - \Pi_C(x)\|_2^2}{2\lambda}\right) dx < +\infty.$$

359 Hence, $p_\lambda(x|y)$ in (2.11) is well-defined.

360 *Proof.* Letting $k = 0$ in Lemma 3.1 and using Assumption 3.2, we conclude the proof. ■

361 **Remark:** Although $\int_{\mathbb{R}^d} q_\theta(x) dx = 1$, $\int_{\mathbb{R}^d} q_\theta^\alpha(x) dx$ may not be finite in rare cases. This
 362 depends on how heavy the tail of $q_\theta(x)$ is. Corollary 3.4 shows that multiplying $q_\theta^\alpha(x)$ with
 363 $\exp\left(-\iota_C^{(\lambda)}(x)\right)$ always leads to a finite integral, regardless of the tail behavior of $q_\theta(x)$.

364 Now, we establish the well-posedness of the posterior measure π_λ in the following propo-
 365 sition. Note that the local Lipschitz stability of posterior distribution in the observation has
 366 been studied in [55, 89] and applied to posterior sampling with PnP prior [56] and generative
 367 models in [4]. Apart from the considered $\iota_C^{(\lambda)}(\tilde{x})$, Proposition 3.5 and Proposition 3 in [56] are
 368 based on similar ideas.

369 **Proposition 3.5.** Suppose Assumption 3.2 holds and that there exist continuous functions
 370 $\Phi_1 : \mathbb{R}^d \rightarrow [0, +\infty)$ and $\Phi_2 : \mathbb{R}^m \rightarrow [0, +\infty)$ such that for any $x \in \mathbb{R}^d$ and $y_1, y_2 \in \mathbb{R}^m$, the
 371 following are satisfied:

$$372 \quad \left| \log(p(y_1|x)) - \log(p(y_2|x)) \right| \leq (\Phi_1(x) + \Phi_2(y_1) + \Phi_2(y_2)) \|y_1 - y_2\|,$$

$$373 \quad \text{and} \quad \int_{\mathbb{R}^d} (1 + \Phi_1(\tilde{x})) \exp\left[c_0 \Phi_1(\tilde{x}) - \iota_C^{(\lambda)}(\tilde{x})\right] d\tilde{x} < +\infty,$$

374
 375 for all $c_0 > 0$. Then, $y \mapsto \pi_{\lambda,y}$ defined in (2.12)) is locally Lipschitz w.r.t. the total-variation
 376 (TV) norm $\|\cdot\|_{\text{TV}}$, i.e., for any compact set K , there exists $M_K \geq 0$ such that for any
 377 $y_1, y_2 \in K$, $\|\pi_{\lambda,y_1} - \pi_{\lambda,y_2}\|_{\text{TV}} \leq M_K \|y_1 - y_2\|$.

378 *Proof.* See Appendix A.3. ■

379 For Gaussian likelihood $p(y|x)$, the conditions in Proposition 3.5 are satisfied when $\Phi_1(x) =$
 380 $c_1 \|x\|_2$ and $\Phi_2(y) = c_2 \|y\|_2$ with positive constants c_1 and c_2 .

381 **3.3. Convergence of NF-ULA.** Most of the existing works on ULA for non-convex poten-
 382 tials [14, 28, 31, 56, 65] assume Lipschitz-continuity of the **score**. If the drift term $\nabla \log p_\lambda(x|y)$
 383 is not Lipschitz, from [43, 48], it cannot generally be guaranteed that the SDE (2.3) will
 384 have a unique strong solution. This is why one must investigate the Lipschitz continuity
 385 of $\nabla \log p_\lambda(x|y)$ before studying the convergence of NF-ULA. First, we make an assumption
 386 about the Lipschitz-continuity of $\nabla \log(p(y|\cdot))$:

387 *Assumption 3.6.* $\nabla \log(p(y|x))$ is L_y -Lipschitz continuous in x , where $L_y > 0$ is a constant.

388 Note that Assumption 3.6 is generally satisfied for common imaging inverse problems.
 389 One example is the popular Gaussian likelihood where $p(y|x) \propto \exp\left(-\|y - Ax\|_2^2/(2\sigma^2)\right)$, for
 390 which $L_y = \|A^\top A\|/\sigma^2$.

391 *Lemma 3.7.* *Under Assumption 3.6, $\nabla \log p_\lambda(x|y)$ is Lipschitz continuous if and only if*
 392 *$\nabla \log q_\theta(x)$ is Lipschitz continuous.*

393 *Proof.* See Appendix A.4. ■

394 For convenience, we explicitly define the Lipschitz condition on the log gradient of $q_\theta(x)$ in
 395 the following assumption:

396 *Assumption 3.8.* There exist $L \geq 0$ such that for any $x_1, x_2 \in \mathbb{R}^d$,

$$397 \quad \|\nabla \log q_\theta(x_1) - \nabla \log q_\theta(x_2)\| \leq L \|x_1 - x_2\|.$$

398 It is therefore natural to ask how to enforce Assumption 3.8 on the NF-based image prior
 399 $q_\theta(x)$ during training or by the network architecture. There have been some studies about
 400 the Lipschitz continuity of the invertible transform T_θ [54, 74, 97], the Lipschitz constants of
 401 invertible neural networks by changing the latent distribution from a standard normal one to
 402 a Gaussian mixture model [37], the Lipschitz constants of other “push-forward” generative
 403 models [86]. However, to the best of our knowledge, there is no study about the Lipschitz
 404 continuity of $\nabla \log q_\theta(x)$ until now.

405 While the equivalent conditions on T_θ for Assumption 3.8 remain unknown, a sufficient
 406 condition on T_θ for Assumption 3.8 can be obtained easily. For instance, when T_θ is a linear
 407 transform mapping a Gaussian distribution $q_z(z)$ to another Gaussian distribution $q_\theta(x)$,
 408 Assumption 3.8 holds. However, this may not be true if T_θ is nonlinear.

409 As we have mentioned that Assumption 3.8 is necessary for the convergence of NF-ULA,
 410 we derive a sufficient condition on T_θ for Assumption 3.8 to hold. Intuitively, distributions
 411 with similar tail behaviors as Gaussian may have similar log gradients as Gaussian, if more
 412 conditions are satisfied. We thus refer to some studies on the tails of normalizing flow priors
 413 [44]. Theorem 4 in [44] shows that affine coupling layer-based flows (e.g., NICE [24], Real-
 414 NVP [25], MAF [75], IAF [51], and Glow [50]) can only map the base normal distribution $q_z(z)$
 415 to a light-tailed distribution $q_\theta(x)$. To be more specific, denote $G(x) = T^{-1}(x)$, where $G(x)$
 416 is a triangular mapping and the Jacobian $J_G(x)$ is a triangular matrix function. From [44],
 417 generally one can assume that for affine coupling layer-based flows, $G_j(x_j, x_{<j}) = \varphi_j(x_{<j})x_j +$
 418 $\eta_j(x_{<j})$, where G_j is the j -th element of the vector $G(x)$ and $x_{<j}$ indicate x_1, \dots, x_{j-1} . The
 419 condition they assume is heuristic: if φ_j is bounded above and η_j is Lipschitz, then $q_\theta(x)$
 420 is light-tailed. In Glow, [50] these conditions on φ and η are satisfied and even stricter.

421 Therefore, we are able to prove the Lipschitz continuity of $\nabla \log q_\theta(x)$ in the proposition
 422 below, by enforcing a stricter condition on φ and η .

423 **Proposition 3.9.** *Assume that the input distribution $q_z(z)$ to the normalizing flow network*
 424 *is the standard normal distribution, and that $T^{-1}(x) = G^{(k)} \circ \dots \circ G^{(1)}(x)$ is a composition of*
 425 *k coupling layers, where each of the layers $G^{(i)} : \mathbb{R}^d \rightarrow \mathbb{R}^d, x^{(i)} \mapsto x^{(i+1)}$ is given by*

$$426 \quad (3.5) \quad G_j^{(i)}(x_j^{(i)}, x_{<j}^{(i)}) = \varphi_j^{(i)}(x_{<j}^{(i)})x_j^{(i)} + \eta_j^{(i)}(x_{<j}^{(i)}), \quad j = 1, \dots, d.$$

427 Denote $x^{(1)} = x$ and $x^{(k+1)} = z$. If $\varphi_j^{(i)}$ is a constant function, $\eta_j^{(i)}$ is Lipschitz and for all
 428 $r < j$, $\frac{\partial \eta_j^{(i)}}{\partial x_r}$ is well-defined almost everywhere and piecewise constant on \mathbb{R} , then $\nabla \log q_\theta(x)$
 429 is Lipschitz continuous on \mathbb{R}^d .

430 *Proof.* See Appendix A.5. ■

431 The conditions on φ, η in Proposition 3.9 are satisfied in *Glow* [50] with *additive coupling*
 432 *layers* where each η is a five-layer sequential network with 2D convolutional layers (denoted
 433 as **Conv2d**) and ReLU activations:

$$434 \quad \eta(x) = \text{Conv2d}(\text{ReLU}(\text{Conv2d}(\text{ReLU}(\text{Conv2d}(x))))),$$

435 where $\text{ReLU}(x) := \max(0, x)$ (applied in an element-wise manner) and $\text{Conv2d}(x) := K_{\text{NF}} * x$
 436 denotes a 2D convolution layer acting on x with a kernel K_{NF} . Further, $\varphi = 1$ is used in
 437 the additive coupling layer. Note that in *Glow*, there is an option of using an *affine coupling*
 438 *layer* where φ is the sigmoid function $\varphi(x) = 1/(1 + e^{-x})$ element-wise. This leads to a
 439 more powerful network and can generate better human face images [50], but $\nabla \log q_\theta(x)$ is
 440 not guaranteed to be Lipschitz anymore. This theoretical observation is corroborated by our
 441 experiments in Sec. 4.1, as we found that NF-ULA with affine coupling layer did not converge.
 442 The conditions on φ and η might be relaxed if $q_z(z)$ is not Gaussian, but this requires re-
 443 training the network since most of the popular normalizing flows accept standard Gaussian
 444 base distribution as input. We leave these studies on the Lipschitz-continuity of $\nabla \log q_\theta(x)$
 445 for future work.

446 In order to prove the convergence of NF-ULA, we need one final assumption.

447 **Assumption 3.10.** There exists $m_y \in \mathbb{R}$ such that for all $x_1, x_2 \in \mathbb{R}^d$, we have

$$448 \quad \langle \nabla \log p(y|x_2) - \nabla \log p(y|x_1), x_2 - x_1 \rangle \leq -m_y \|x_2 - x_1\|_2^2.$$

450 This condition is called the *contractivity condition* of $\nabla \log p(y|x)$ and is used to prove
 451 the contractivity of the drift term $\nabla \log p_\lambda(x|y)$ at infinity (see proofs of Theorem 3.11 in
 452 Appendix A.6). Note that the influence of the drift's contractivity condition has been studied
 453 in ULA for non-convex potentials [14, 22, 65].

454 If Assumption 3.10 is satisfied with $m_y > 0$, then $x \mapsto -\log p(y|x)$ is m_y -strongly convex.
 455 If Assumption 3.6 is satisfied, then Assumption 3.10 holds for $m_y = -L_y$. However, we are
 456 interested to find $m_y > -L_y$ while Assumption 3.6 holds, since we will see in the proofs of
 457 Theorem 3.11 and Theorem 3.12 in Appendix A.6 and A.7 that a larger m_y is beneficial to
 458 the convergence of NF-ULA.

459 In what follows, we introduce the associated stochastic kernel $R_\delta : \mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d) \rightarrow [0, 1]$ of
 460 the NF-ULA (2.13) and the drift $b_\lambda \in C(\mathbb{R}^d, \mathbb{R}^d)$:

$$461 \quad (3.6) \quad \begin{aligned} R_\delta(x, A) &= (2\pi)^{-d/2} \int_{\mathbb{R}^d} \mathbf{1}_A \left(x + \delta b_\lambda(x) + \sqrt{2\delta}z \right) \exp[-\|z\|^2/2] dz, \\ b_\lambda(x) &= \nabla \log p_\lambda(x|y) = \nabla \log p(y|x) + \alpha \nabla \log q_\theta(x) + \frac{\Pi_C(x) - x}{\lambda}, \end{aligned}$$

462 where $x \in \mathbb{R}^d$ and $A \in \mathcal{B}(\mathbb{R}^d)$. Here b_λ has the subscript λ and is different from the b defined
 463 in Sec. 3.1 because of $(\Pi_C(x) - x)/\lambda$. Given X_k in NF-ULA (2.13), $R_\delta(X_k, \cdot)$ is actually a
 464 probability measure which defines the transition probability $p(X_{k+1}|X_k)$.

465 With all the previous four assumptions A 3.2, A 3.6, A 3.8, and A 3.10 holding, we can
 466 prove that NF-ULA (Algorithm 2.1) is convergent, or more precisely, the stochastic kernel
 467 R_δ admits a unique invariant distribution $\pi_{\delta, \lambda}$. We follow the proof in SM6.2 from [57] but
 468 our theorem and proof are slightly different, as we do not include the parameter ε of PnP
 469 denoisers in the condition. The first thing to prove is that R_δ defines a contractive mapping.

470 **Theorem 3.11.** *Assume A 3.2, A 3.6, A 3.8, and A 3.10. Assume $V(x) = 1 + \|x\|^2$, $x \in \mathbb{R}^d$.
 471 Let $\lambda, \alpha, C, L_y, L$ be the ones in NF-ULA (Algorithm 2.1). Let m_y be the parameter in A 3.10.
 472 Let $\lambda > 0$, such that $2\lambda(L_y + \alpha L - \min(m_y, 0)) \leq 1$ and let $\bar{\delta} = (1/6)(L_y + \alpha L + 1/\lambda)^{-1}$.
 473 Then for any convex and compact C with $0 \in C$, there exist $A_1 \geq 0$ and $\rho_1 \in [0, 1)$ such that
 474 for any $\delta \in (0, \bar{\delta}]$, $x_1, x_2 \in \mathbb{R}^d$, and $k \in \mathbb{N}$ we have*

$$475 \quad \begin{aligned} \left\| \delta_{x_1} R_\delta^k - \delta_{x_2} R_\delta^k \right\|_V &\leq A_1 \rho_1^{k\delta} (V^2(x_1) + V^2(x_2)), \text{ and} \\ \mathbf{W}_1 \left(\delta_{x_1} R_\delta^k, \delta_{x_2} R_\delta^k \right) &\leq A_1 \rho_1^{k\delta} \|x_1 - x_2\|_2. \end{aligned}$$

476 *Proof.* See Appendix A.6. ■

477 In the above theorem the Dirac measures $\delta_{x_1}, \delta_{x_2}$ can be extended to any measures $\nu_1, \nu_2 \in$
 478 $\mathcal{P}_1(\mathbb{R}^d)$:

$$479 \quad (3.7) \quad \begin{aligned} \left\| \nu_1 R_\delta^k - \nu_2 R_\delta^k \right\|_V &\leq A_1 \rho_1^{k\delta} \left(\int_{\mathbb{R}^d} V^2(\tilde{x}) d\nu_1(\tilde{x}) + \int_{\mathbb{R}^d} V^2(\tilde{x}) d\nu_2(\tilde{x}) \right), \\ \mathbf{W}_1 \left(\nu_1 R_\delta^k, \nu_2 R_\delta^k \right) &\leq A_1 \rho_1^{k\delta} \left(\int_{\mathbb{R}^d} \|\tilde{x}\| d\nu_1(\tilde{x}) + \int_{\mathbb{R}^d} \|\tilde{x}\| d\nu_2(\tilde{x}) \right). \end{aligned}$$

480 From Theorem 6.18 in [98], $(\mathcal{P}_1(\mathbb{R}^d), \mathbf{W}_1)$ is a complete metric space. For any measure
 481 $\nu \in \mathcal{P}_1(\mathbb{R}^d)$, define $f : \mathcal{P}_1(\mathbb{R}^d) \rightarrow \mathcal{P}_1(\mathbb{R}^d)$ as $f(\nu) = \nu R_{\varepsilon, \delta}$. Then for any $\delta \in (0, \bar{\delta}]$, there
 482 exists large enough $m_\delta \in \mathbb{N}^*$ such that f^{m_δ} is a contractive mapping. Therefore we can apply
 483 the Picard fixed point theorem and we obtain that R_δ admits a unique invariant probability
 484 measure $\pi_{\delta, \lambda} \in \mathcal{P}_1(\mathbb{R}^d)$. Since $\pi_{\delta, \lambda}$ is subject to bias comparing with the solution of the SDE
 485 $dX_t = b_\lambda(X_t)dt + \sqrt{2\delta} dB_t$, in the Theorem below, we follow the proof in SM6.3 from [57] and
 486 give a nonasymptotic bias analysis:

487 **Theorem 3.12.** *Assume A 3.2, A 3.6, A 3.8, A 3.10. Assume $V(x) = 1 + \|x\|^2$, $x \in \mathbb{R}^d$.
 488 Let $\lambda, \alpha, C, L_y, L$ be the ones in NF-ULA (Algorithm 2.1). Let m_y be the parameter in A 3.10.*

489 Let $\lambda > 0$ such that $2\lambda(L_y + \alpha L - \min(m_y, 0)) \leq 1$ and let $\bar{\delta} = (1/6)(L_y + \alpha L + 1/\lambda)^{-1}$.
 490 Then for any $\delta \in (0, \bar{\delta}]$ and C convex and compact, R_δ admits an *unique* invariant probability
 491 measure $\pi_{\delta, \lambda}$. In addition, there exists $B_1, B_2, B_3 \geq 0$, $\tilde{\rho}_1 \in [0, 1)$ such that for any $\delta \in (0, \bar{\delta}]$,
 492 $k \in \mathbb{N}^*$,

$$493 \quad \left\| \delta_x R_\delta^k - \pi_\lambda \right\|_V \leq B_1 \tilde{\rho}_1^{k\delta} V^2(x) + B_2 V(x) \sqrt{\delta^2 k \left(d + \frac{B_3 \delta}{3} \right)}.$$

494 *Proof.* See Appendix A.7. ■

495 **Remark:** Note that there is a trade-off of selecting the step-size δ . In order to achieve a
 496 small bias, one needs to set a large time interval $t = k\delta$, keep t fixed and use a small step size
 497 δ . However, larger k means drawing more samples, resulting in longer computation time. In
 498 practice, the burn-in period is incorporated in t , in which the Markov Chain is dramatically
 499 exploring the state space.

500 **4. Experiments in Bayesian Imaging.** We apply NF-ULA and PnP-ULA on three inverse
 501 problems: image motion deblurring, image inpainting, and limited-angle computed tomogra-
 502 phy (CT) reconstruction. We compare with PnP-ULA since, to the best of our knowledge, it
 503 is the state-of-the-art Langevin algorithm with data-driven non-convex regularizers.

504 **Choice of α :** For both NF-ULA and PnP-ULA on different problems, we fine-tune α such
 505 that the peak signal-to-noise ratio (PSNR) of the sample mean gets maximized. While in most
 506 cases $\alpha \in (0, 5]$ works well, for NF-ULA it is also related to the architecture of the normalizing
 507 flow. For CT reconstruction, we use the pre-trained patchNR, a NF-based regularizer learned
 508 on medical images, from the code provided in [3] and choose $\alpha = 5000$. Notably, in the original
 509 implementation, the maximum a posteriori estimator was considered, and $\alpha = 700$ was the
 510 best choice.

511 **Choices of C and λ :** We only perform the study of choosing different C and λ in the
 512 deblurring experiments. From [56], a projection term $(\text{Id} - \Pi_C)$ is introduced to PnP-ULA
 513 to make sure that the posterior satisfies the tail-decay condition. Therefore, for posterior
 514 distributions with a slower tail-decay, a smaller C is recommended. We found experimentally
 515 that NF-ULA was numerically stable when the NF prior was trained for more than 20 epochs,
 516 even with a large C . In this case, C is chosen to be large enough such that Π_C is never
 517 activated, since we do not expect to choose a small C to change the behaviors of NF-ULA if
 518 it already converges. For a normalizing flow that is not well trained (less than 5 epochs), it
 519 is recommended that C should be the same as the range C_R of the dataset. In the imaging
 520 problems, we have that $C_R = [0, 1]^d$. See Table 1 for details on the algorithm behaviors
 521 of NF-ULA with different choices of C and normalizing flow architectures. For well-trained
 522 normalizing flows in NF-ULA and denoiser in PnP-ULA, we set $C = [-100, 100]^d$. Actually
 523 all the samples generated in Tables 2, 3, and 4 never escaped $[-0.2, 1.2]^d$, indicating that the
 524 projection $\Pi_C(x)$ was never activated. We keep $\lambda = 5 \times 10^{-5}$, even though different λ makes
 525 no difference in most of our experiments.

526 **Choice of δ :** From the convergence analysis in Theorem 3.11 and Theorem 3.12, any $\delta <$
 527 $(1/6)(L_y + \alpha L + 1/\lambda)^{-1}$ should work. However, this upper bound is not a strict bound and
 528 in practice, it is not easy to know the Lipschitz constant L of $\nabla \log q_\theta(x)$. To give an upper

529 bound of L , we calculate the spectral norm of $\nabla^2 \log q_\theta(x)$ through power iteration when x
 530 is randomly chosen in C_R and the spectral norm are smaller than 2×10^5 . This upper
 531 bound for L is still too loose since we find that NF-ULA converges for many δ larger than
 532 the corresponding upper bound. Moreover, as different λ makes no difference in most of our
 533 experiments, we fine tune δ instead of precisely calculating the upperbound given by L and λ .
 534 In most of our experiments, δ is chosen to be smaller than $(1/10)L_y^{-1}$, to ensure convergence of
 535 different algorithms. Our choice of δ is slightly different from PnP-ULA because the Lipschitz
 536 parameter L of the PnP prior can be set to 1 during training.

537 **Implementations:** We implement all the experiments in Python and utilize PyTorch for im-
 538 plementing the ULA Markov chains. The numerical experiments are run on Intel(R) Xeon(R)
 539 Platinum 8358P CPU with four Nvidia Tesla A100 GPUs. Codes for NF-ULA are available
 540 at Github¹.

541 **4.1. Image Deblurring.** We first consider a non-blind motion deblurring problem on hu-
 542 man face images. The corresponding forward operator A applies a convolution on the image
 543 x with a 9×9 motion-blurring kernel of horizontal blurring direction, with all the elements
 544 in the fifth row of the kernel being $1/9$ and the other rows being 0. Both $x, y \in \mathbb{R}^d$, where
 545 $d = 3 \times 128 \times 128$ and the forward operator $A : \mathbb{R}^d \mapsto \mathbb{R}^d$ is linear. To describe the for-
 546 ward model (likelihood), we add Gaussian noise $n \sim \mathcal{N}(0, \sigma^2 I^d)$, leading to the following
 547 measurement equation and likelihood:

$$548 \quad y = Ax + n, \quad p(y|x) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{\|y - Ax\|^2}{2\sigma^2}\right).$$

549 **4.1.1. Networks and Parameters.** To realize NF-ULA, we train the well-known flow-
 550 based model, Glow [50], on the human face dataset FFHQ [49] without the first 20 images,
 551 which amounts to 69980 images in total. All the images are 3-channel images normalized
 552 to $C_R = [0, 1]^{3 \times 128 \times 128}$. We train Glow from scratch using the publicly available PyTorch
 553 implementation², however, NF-ULA can also use an appropriate pre-trained model. The
 554 architecture of Glow has five blocks with 32 flows in each block.

555 For PnP-ULA [56], we use the real spectral normalization DnCNN (realSN-DnCNN),
 556 which is a Lipschitz-continuous denoiser proposed in [84]. In order to see the behavior of the
 557 denoiser without the Lipschitz constraint, we train both the standard DnCNN [104] and
 558 realSN-DnCNN [84] on the image patches of a 980-image subset of FFHQ. To train the
 559 denoiser, we follow the same procedure reported in [56], i.e., we add Gaussian noise with
 560 the variance $\varepsilon = (5/255)^2$ on the training data batches. In fact, we also tested $\varepsilon = (15/255)^2$
 561 or $(25/255)^2$ but the generated samples get lower PSNR. To train the standard DnCNN, we
 562 directly use the code in the Image Restoration Toolbox³. We keep the default parameter
 563 settings to train a 17-layer DnCNN on image patches of size 40×40 . For realSN-DnCNN,
 564 the original implementation⁴ in [84] only supports training on grayscale images, therefore we

¹<https://github.com/caiziruo/NF-ULA>

²<https://github.com/rosinality/glow-pytorch>

³<https://github.com/cszn/KAIR>

⁴https://github.com/uclaopt/Provable_Plug_and_Play/

Table 1

The behavior of NF-ULA by different Glow and different choices of C . The algorithm does not converge for Glow with affine coupling layers. For Glow with additive coupling layers, the algorithm converges better when Glow is trained for more epochs.

| Deblurring | network: Glow. $\alpha = 1.5$ | | | |
|------------|-------------------------------|--------|-----------------|-----------|
| | coupling layers | epochs | C | PSNR |
| face1 | | | | |
| NF-ULA | affine | 100 | $[0, 1]^d$ | divergent |
| NF-ULA | additive | 5 | $[-100, 100]^d$ | divergent |
| NF-ULA | additive | 5 | $[0, 1]^d$ | 26.58 |
| NF-ULA | additive | 20 | $[-100, 100]^d$ | 29.84 |
| NF-ULA | additive | 100 | $[-100, 100]^d$ | 30.42 |

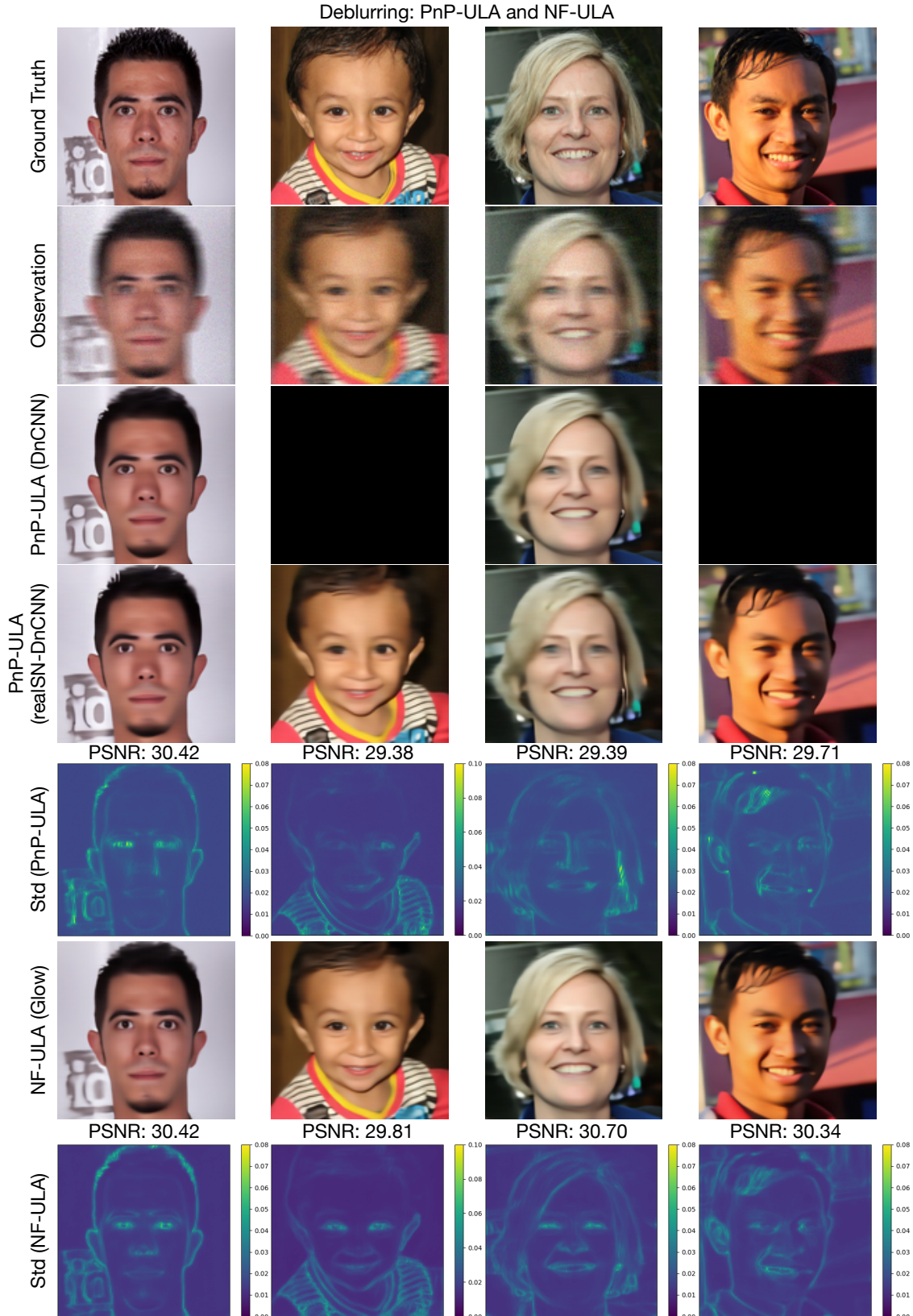
565 modified the code to make it applicable to color images. We also set up the number of network
 566 layers as 17 and preprocess the data to patches of size 40×40 , while setting the Lipschitz
 567 parameter to 1. Although DnCNN and realSN-DnCNN are trained on such a small dataset,
 568 they can still obtain a peak signal-to-noise ratio (PSNR) of more than 40 dB on the validation
 569 set. In fact, the original implementation in [84] trains the denoiser on a dataset consisting of
 570 only 400 images, and increasing the size of the dataset does not necessarily lead to a higher
 571 PSNR on the validation set.

572 The Glow network that we used for NF-ULA has 100870544 parameters in total, while
 573 DnCNN has 559363 parameters and realSN-DnCNN has 558336 parameters. To train 100
 574 epochs, Glow spent up to 100 hours, while DnCNN and realSN-DnCNN spent less than 5
 575 hours. The heavier network and the longer training time for Glow pay off when it comes to
 576 reconstruction performance and image quality.

577 **ULA parameters settings:** We set the standard deviation of the Gaussian noise n to
 578 $\sigma = 0.02$. To ensure that both PnP-ULA and NF-ULA are numerically stable, we select the
 579 step size $\delta = 5 \times 10^{-5}$. For Glow, DnCNN and realSN-DnCNN, $\alpha = 1.5$ leads to the highest
 580 PSNR. We initialize $X_0 = y$, the noisy blurred observation for both NF-ULA and PnP-ULA.

581 **4.1.2. Performance of the Algorithms.** To explore the state space thoroughly, all the
 582 experiments have burn-in iterations less than 5000. Since the first sample X_0 is initialized as
 583 the observation y , the PSNR of the samples X_n starts from around 22.78 dB and then keeps
 584 going up and finally stays in an interval, e.g. [29.0, 31.0]. After the burn-in time, we calculate
 585 the posterior mean by obtaining 10000 samples and compute the PSNR of the sample mean. To
 586 draw 10000 samples, NF-ULA spends around 3100 seconds, while PnP-ULA spends 30 seconds.
 587 For both algorithms, calculating the posterior mean by more samples, e.g. 10^6 samples, does
 588 not improve the PSNR. When generating equal samples, NF-ULA spends more time mainly
 589 because of the large network Glow uses - the Glow we use has approximately 100 times more
 590 parameters than realSN-DnCNN. In fact, we found that computing and forwarding the auto-
 591 gradient function of $q_\theta(x)$ takes 10% longer time than forwarding $q_\theta(x)$ itself. However, we
 592 believe that NF-ULA has great potential to leverage smaller and more advanced normalizing
 593 flows to reduce computational time. In Sec. 4.3, we use a lightweight NF-based regularizer
 594 and the resulting NF-ULA requires significantly less time.

Figure 1. Deblurring by PnP-ULA and NF-ULA. What each row represents is written on left of the rows. PSNR values corresponding to the sample mean are provided in Table 2. PnP-ULA with standard DnCNN does not converge on face2 and face4. On all four faces, NF-ULA (Glow) yields a higher PSNR (for the sample mean estimator) than PnP-ULA (realSN-DnCNN). The sample mean images also have a better visual quality for NF-ULA.



595 To examine the Lipschitz continuity of $\nabla \log q_\theta(x)$ for different kinds of coupling layers, we
 596 train two different Glow networks for 100 epochs each, with affine and additive coupling layers,
 597 respectively. Also, to verify our hypothesis that better training of the normalizing flow prior
 598 will imply better samples from NF-ULA, we trained Glow (additive coupling layers) for 5,
 599 20, and 100 epochs, and compared their performance when used in the NF-ULA framework.
 600 The PSNR values of the sample mean images corresponding to these variants of NF-ULA
 601 with different NF-based priors are reported in Table 1. With affine coupling layers in Glow,
 602 NF-ULA fails to converge because $\nabla \log q_\theta(x)$ is not Lipschitz continuous, which is consistent
 603 with Proposition 3.9. For Glow with additive coupling layers and also for the case where the
 604 Glow model is well-trained (more than 20 epochs), NF-ULA works well and the generated
 605 samples do not blow up, even in the case where $C = [-100, 100]^d$ is much bigger than C_R .
 606 This suggests that a well-trained prior $q_\theta(x)$ already satisfies the tail decay conditions, without
 607 imposing the projection $\text{Id} - \Pi_C$. However, it is still essential for the theoretical study. For
 608 poorly trained Glow (less than 5 epochs) and large C , NF-ULA does not work well - most of
 609 the samples go far beyond C_R and the PSNR of them are below 10 dB. If C is set to be a
 610 much smaller set, e.g., $C = C_R$, then the PSNR can be up to 26 dB, which is still considerably
 611 lower than what one can achieve with a well-trained Glow.

612 Intuitively, $q_\theta(x)$ is more *diffusive* when Glow is trained for only a few epochs. After
 613 training for some epochs, the normalizing flow is more suitable to serve as an image prior,
 614 and the density $q_\theta(x)$ is more concentrated. Moreover, the tail decay condition of $p(x|y)$ is
 615 also satisfied with a well-trained prior, even without the projection term.

616 To compare the performance of ULA with both PnP- and normalizing flow-induced priors,
 617 we run NF-ULA using Glow, PnP-ULA using DnCNN, and PnP-ULA with realSN-DnCNN
 618 on four human face images randomly selected from the first 20 images of FFHQ [49], which are
 619 the ones not used during training. In the following experiments, we use Glow with additive
 620 coupling layers. Glow, DnCNN, and realSN-DnCNN are all trained for 100 epochs for a fair
 621 comparison. The results are shown in Figure 1 and Table 2. From Table 2, we note that
 622 NF-ULA with Glow generates samples with the highest PSNR. We also present the standard
 623 deviation of the samples on the same channel in Fig 1. NF-ULA has richer details for the
 624 posterior mean and more variations for standard deviation, particularly on the eyes, mouths,
 625 and hair. This is probably due to a more accurate prior learned by the generative model. It is
 626 worth noting that PnP-ULA with DnCNN shows great performance on Face-1 and Face-3, but
 627 is divergent on Face-2 and Face-4. However, PnP-ULA with realSN-DnCNN converges on all
 628 images, albeit with lower PSNR than NF-ULA. [Moreover, we also performed the simulations](#)
 629 [of PnP-ULA using DRUnet \[103\], a newer denoiser than DnCNN, but the results are very](#)
 630 [comparable to the ones obtained with DnCNN - the algorithm is not convergent on Face-2](#)
 631 [and Face-4 due to DRUnet not being Lipschitz.](#)

632 We record the PSNR of the samples and the minimum mean square error (MMSE) es-
 633 timator in Figure 2. It’s about the deblurring experiments of face1 and the evolutions for
 634 face2, face3, and face4 are similar. In the left figure, we start from the burn-in period un-
 635 til 15000 samples. The MMSE estimator is approximated by the last 10000 samples. For
 636 both algorithms, the burn-in periods are less than 5000 samples. Regardless of the sampling
 637 time, NF-ULA shows a faster increase of PSNR, which means the convergence speed of the
 638 first-order moment for NF-ULA mildly outperforms PnP-ULA. However, in the right figure,

Table 2

Deblurring: Comparison of ULA with different priors for image deblurring. PnP-ULA with a standard DnCNN does not converge on face2 and face4. NF-ULA (Glow) generates samples with slightly higher PSNR than PnP-ULA.

| Deblurring | net_epochs = 100, $C = [-100, 100]^d$ | | |
|------------|---------------------------------------|----------------|-----------|
| | network | parameters | PSNR |
| face1 | | | |
| NF-ULA | Glow | $\alpha = 1.5$ | 30.42 |
| PnP-ULA | DnCNN | $\alpha = 1.5$ | 30.40 |
| PnP-ULA | realSN-DnCNN | $\alpha = 1.5$ | 30.42 |
| face2 | | | |
| NF-ULA | Glow | $\alpha = 1.5$ | 29.81 |
| PnP-ULA | DnCNN | $\alpha = 1.5$ | divergent |
| PnP-ULA | realSN-DnCNN | $\alpha = 1.5$ | 29.38 |
| face3 | | | |
| NF-ULA | Glow | $\alpha = 1.5$ | 30.70 |
| PnP-ULA | DnCNN | $\alpha = 1.5$ | 29.61 |
| PnP-ULA | realSN-DnCNN | $\alpha = 1.5$ | 29.39 |
| face4 | | | |
| NF-ULA | Glow | $\alpha = 1.5$ | 30.34 |
| PnP-ULA | DnCNN | $\alpha = 1.5$ | divergent |
| PnP-ULA | realSN-DnCNN | $\alpha = 1.5$ | 29.71 |

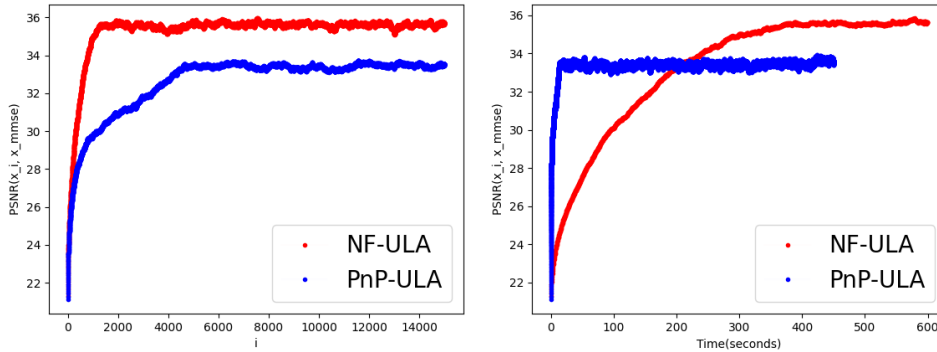


Figure 2. The evolution of $PSNR(x_i, x_{mmse})$ of deblurring (face1). The left figure is according to the number of the samples and the right one is according to elapsed time. A faster increase means a faster convergence speed.

639 we consider evolution w.r.t. the sampling time and NF-ULA has a slower increase of PSNR.
 640 NF-ULA has a burn-in time of about 400 seconds while PnP-ULA is less than 40 seconds.

641 One common approach to studying the convergence speed of a Markov chain is to calculate
 642 the d -dimensional auto-correlation function (ACF) of it. For samples $\{Y_i\}_{i=1}^N$ from a one-

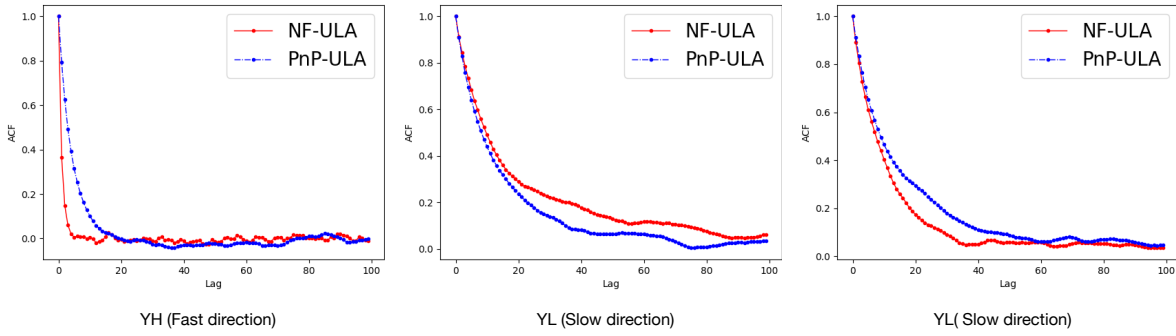


Figure 3. The autocorrelation function (ACF) of the samples (deblurring on face1). The definition of the ACF is given in (4.1). ACF is calculated by wavelet basis using the band-pass coefficients (YH) and the low-pass coefficients (YL). Faster decreasing ACF implies faster convergence of the Markov chain.

643 dimensional Markov chain, the sample auto-correlation function is given by

644 (4.1)
$$\omega(l) = \frac{\sum_{t=1}^{n-l} (Y_{t+l} - \bar{Y})(Y_t - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}, \quad \bar{Y} = \frac{1}{n} \sum_{t=1}^n Y_t,$$

645 where $l = 0, 1, \dots, n - 1$, is the lag between the samples. Since the samples generated
 646 by ULA are not strictly uncorrelated, faster decreasing ACF means that the samples are
 647 less correlated and generally implies faster convergence of the Markov chain to some extent.
 648 Notably, the calculation of ACF is not easy in high-dimensional problems. Therefore, we firstly
 649 transform the image samples using wavelet basis and obtain the band-pass coefficients (YH)
 650 and the low-pass coefficients (YL). YH contains the image details while YL captures the overall
 651 image structure. We consider the finest scale coefficients in YH. To characterize the Markov
 652 chain generated by NF-ULA (Glow) and PnP-ULA (realSN-DnCNN), we randomly select
 653 100 dimensions respectively from YH and YL, and calculate the ACF on those dimensions.
 654 It should be noted that the ACF can have different rates of decay in different directions,
 655 therefore it is time-consuming to analyze the ACF of all the image dimensions and calculate
 656 the fastest and slowest decreasing direction. However, ACF in YH mostly have faster decrease
 657 and ACF in YL will have slower decrease. In Fig 3, we show the convergence of ACF (face1),
 658 along one *fast direction* in YH and two *slow directions* in YL. In the fast direction, the ACF
 659 of PnP-ULA decreases from 1 to 0 within about 20 lags, while for NF-ULA it converges even
 660 faster (within approx. 10 lags). For slow directions, both NF-ULA and PnP-ULA hold a
 661 non-zero ACF until more than 40 lags, and it is not immediately clear which of these two
 662 methods has a faster decay of the ACF. ACF of face2, face3 and face4 are similar as face1
 663 and hence omitted here.

664 **4.2. Image Inpainting.** In this section, we present the experimental results on image in-
 665 painting. We still consider human face images and use the Glow and realSN-DnCNN networks
 666 trained as explained in Sec. 4.1. For inpainting, the forward operator A applies masking on
 667 x so that 80% of the pixels in x are missing. We choose different α to ensure both NF-
 668 ULA and PnP-ULA have the best performance: $\alpha = 2.0$ works well for NF-ULA, while for

Figure 4. Comparison of image inpainting performance of PnP-ULA and NF-ULA. The PSNR values of the sample mean images are reported in Table 3. NF-ULA (Glow) yields a higher PSNR (by approximately 2.5-3.0 dB) of the sample mean images than PnP-ULA with a realSN-DnCNN denoiser. This experiment underscores the importance of stronger regularization (which the Glow-based prior can achieve) when the forward operator is severely ill-posed.

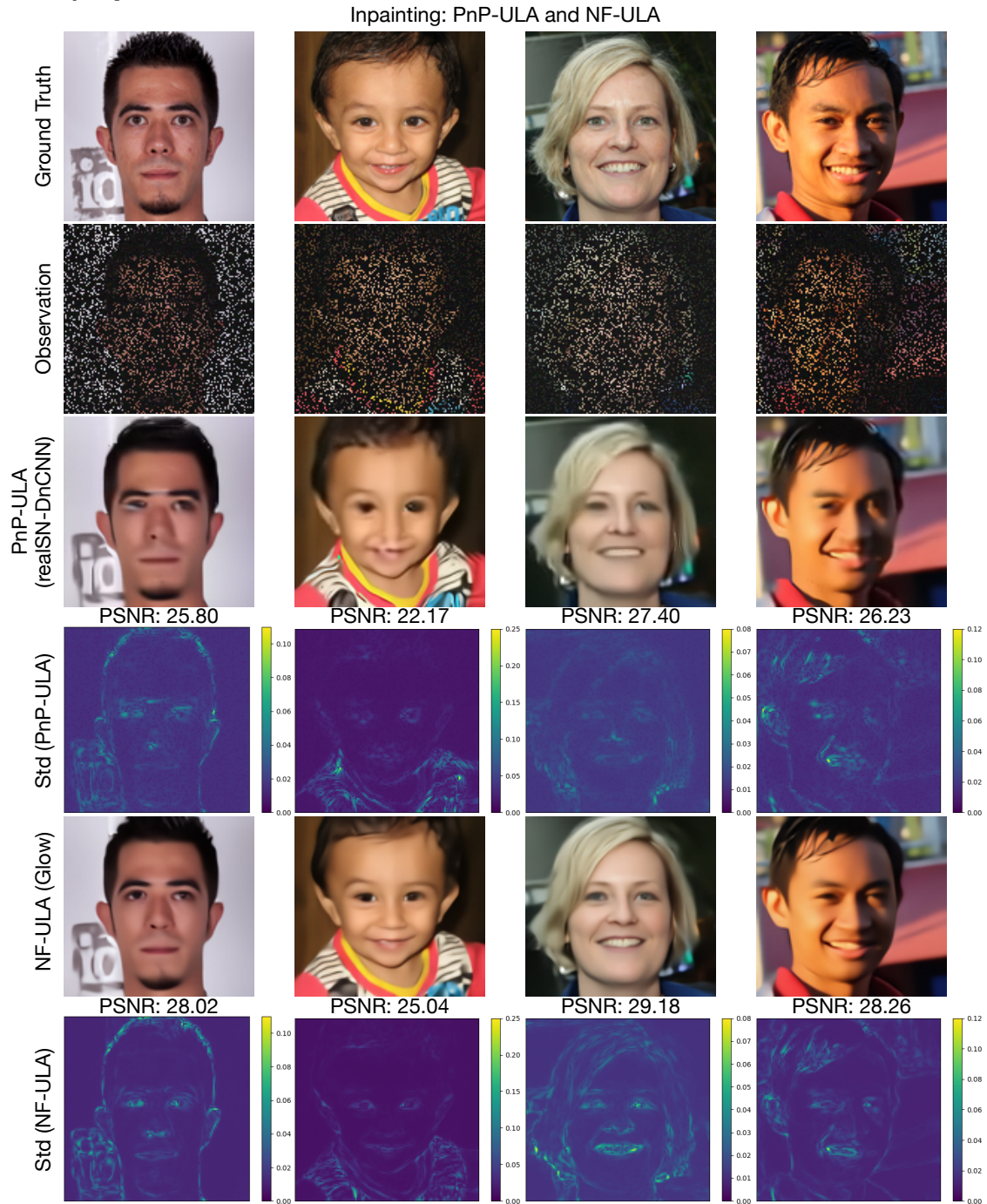


Table 3

Inpainting: Comparison of the ULA with different priors. The parameter α is fine-tuned to maximize the PSNR for both algorithms. Since inpainting relies more on the prior, NF-ULA has a higher PSNR for the sample mean as compared with PnP-ULA.

| Inpainting | net_epochs = 100, $C = [-100, 100]^d$ | | |
|------------|---------------------------------------|----------------|-------|
| | network | parameters | PSNR |
| face1 | | | |
| NF-ULA | Glow | $\alpha = 2$ | 28.02 |
| PnP-ULA | realSN-DnCNN | $\alpha = 2.5$ | 25.80 |
| face2 | | | |
| NF-ULA | Glow | $\alpha = 2$ | 25.04 |
| PnP-ULA | realSN-DnCNN | $\alpha = 2.5$ | 22.17 |
| face3 | | | |
| NF-ULA | Glow | $\alpha = 2$ | 29.18 |
| PnP-ULA | realSN-DnCNN | $\alpha = 2.5$ | 27.40 |
| face4 | | | |
| NF-ULA | Glow | $\alpha = 2$ | 28.26 |
| PnP-ULA | realSN-DnCNN | $\alpha = 2.5$ | 26.23 |

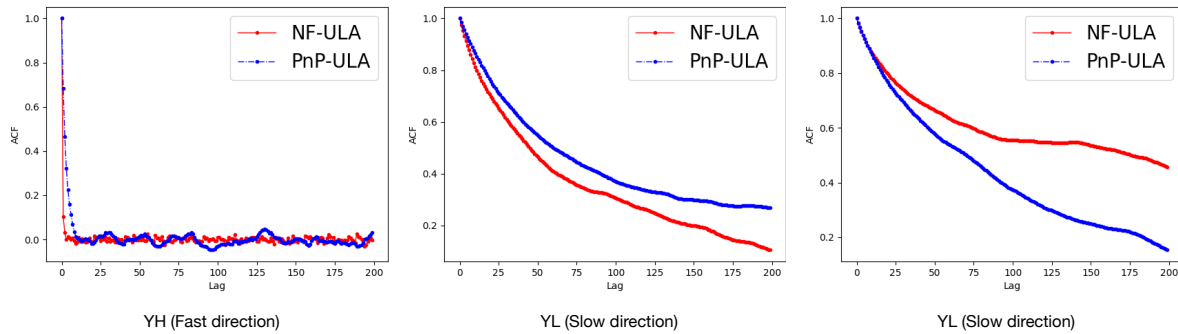


Figure 5. The auto-correlation function (ACF) of the samples (inpainting on face1). The definition of ACF is given in (4.1). ACF is calculated by wavelet basis using the band-pass coefficients (YH) and the low-pass coefficients (YL). Faster decreasing ACF implies faster convergence of the Markov chain.

669 PnP-ULA $\alpha = 2.5$ works the best. We maintain the same setting for the other important
 670 hyper-parameters of the experiment, such as the noise standard deviation $\sigma = 0.02$, the di-
 671 mension of image and observation $x, y \in \mathbb{R}^d = \mathbb{R}^{3 \times 128 \times 128}$, the step-size of both algorithms
 672 $\delta = 5 \times 10^{-5}$, the convex set $C = [-100, 100]^d$, and the initialization $X_0 = y$.
 673 **Performance of the algorithms:** In contrast with deblurring, we found that both NF-ULA
 674 and PnP-ULA have much longer burn-in times. We initialize X_0 with the measurement y ,
 675 whose PSNR is only 5.46 dB. NF-ULA has a burn-in iteration of 10000 until the PSNR of
 676 X_n grows more than 25 dB and becomes stable, while PnP-ULA takes about 80000-iterations
 677 (eight times larger than NF-ULA) for burn-in. The reason might be that Glow’s powerful

Table 4

Limited-angle CT reconstruction from Gaussian noise-corrupted limited-angle projection data. α is chosen to maximize the PSNR for both PnP-ULA and NF-ULA to make a fair comparison. NF-ULA leads to a higher sample mean PSNR than PnP-ULA.

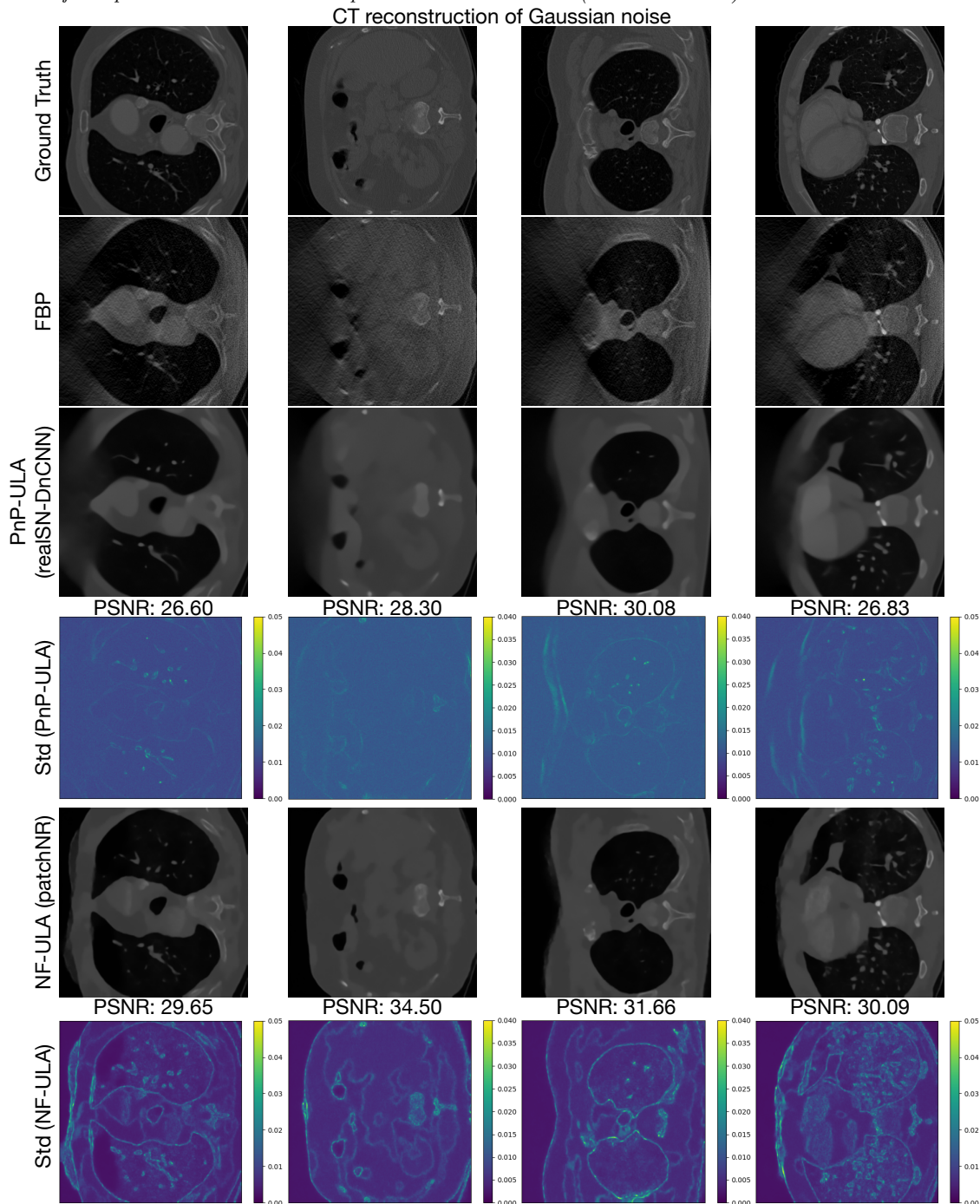
| CT | $C = [-100, 100]^d$ | | |
|---------|---------------------|-----------------|-------|
| | network | parameters | PSNR |
| Image-1 | | | |
| NF-ULA | PatchNR | $\alpha = 5000$ | 29.65 |
| PnP-ULA | realSN-DnCNN | $\alpha = 3$ | 26.60 |
| Image-2 | | | |
| NF-ULA | PatchNR | $\alpha = 5000$ | 34.50 |
| PnP-ULA | realSN-DnCNN | $\alpha = 3$ | 28.30 |
| Image-3 | | | |
| NF-ULA | PatchNR | $\alpha = 5000$ | 31.66 |
| PnP-ULA | realSN-DnCNN | $\alpha = 3$ | 30.08 |
| Image-4 | | | |
| NF-ULA | PatchNR | $\alpha = 5000$ | 30.09 |
| PnP-ULA | realSN-DnCNN | $\alpha = 3$ | 26.83 |

678 prior information accelerates the burn-in process, particularly on the pixels missing in the
 679 observation. After the burn-in time, we draw 10000 samples and compute the PSNR of
 680 the samples' mean. Drawing 10000 samples takes approximately the same time as in the
 681 deblurring experiment.

682 The sample mean images and the standard deviations are shown in Fig. 4. As compared
 683 with PnP-ULA, NF-ULA recovers more areas of the face and shows higher uncertainties on
 684 eyes, hairs, noses, and teeth. Those areas are easily distinguishable between different human
 685 faces and should have higher uncertainties than other areas, e.g., foreheads and cheeks. From
 686 Table 3, we observe that NF-ULA achieves a higher PSNR than PnP-ULA. For both NF-
 687 ULA and PnP-ULA, the PSNR of the posterior mean is lower than that of the deblurring
 688 experiment - the forward operator of masking 80% pixels is not invertible and the observation
 689 y in inpainting is ill-conditioned, which means that in the Bayesian setting, the samples rely on
 690 the prior than the likelihood. In such cases, NF-ULA provides a stronger and more informative
 691 prior as compared to PnP-ULA.

692 To calculate the ACF in this inpainting results, we use the same strategy as in deblurring:
 693 calculating the ACF respectively on 100 randomly selected dimensions of YH and YL. In
 694 Fig. 5, we show the ACF including one fast direction in YH and two slow directions in YL.
 695 Similar to Fig. 3, among those fast decreasing directions, the ACF of NF-ULA is slightly
 696 faster than PnP-ULA and they both decrease from 1 to 0 within 20 lags. For slow directions,
 697 both algorithms have slower decreasing ACF than the deblurring experiments and we cannot
 698 conclude for which method, the ACF decreases faster. ACF of face2, face3 and face4 are
 699 similar as face1 and omitted.

Figure 6. CT reconstruction of Gaussian noise (limited angles). What each column represents is written on top of the columns. PSNR of the samples mean are provided in Table 4. NF-ULA (patchNR) yields higher PSNR of samples mean and better samples Std than PnP-ULA (realSN-DnCNN).



700 **4.3. CT Reconstruction from limited-angle measurements.** We consider the classical
 701 ill-posed inverse problem of X-ray CT reconstruction from limited-angle projection data. We
 702 use the `torch_radon` library [82] to model the forward operator A that computes projections
 703 using a fan-beam acquisition geometry. Instead of considering the full angular range $[0, 2\pi]$,
 704 we only have projection data corresponding to an angular sweep over the range $[0.1\pi, 0.9\pi]$
 705 of angles. We set the number of detector elements to 144, and test the algorithms for both
 706 Gaussian noise and Poisson noise (see Appendix B). The noisy projection data is given by

$$707 \quad (4.2) \quad y = Ax + n \text{ or } y \sim P(Ax),$$

708 where n is used to denote additive Gaussian noise and $P(Ax)$ denotes adding a non-additive
 709 noise on Ax such as Poisson noise. The image to be recovered is $x \in \mathbb{R}^{362 \times 362}$ and the sinogram
 710 is $y \in \mathbb{R}^{144 \times 512}$. We calculate the norm of A and obtain that $\|A\| = \sup_{x: \|x\|=1} \|Ax\| \approx 100$.

711 **Network architecture:** The features and textures of medical images are more difficult
 712 to learn as compared with those in natural images. Hence, normalizing flows do not have
 713 comparable performance in generating semantically meaningful images for medical imaging
 714 applications, unlike applications involving natural images. Therefore, we utilize *patchNR* [3],
 715 which is analogous to normalizing flow, to apply NF-ULA for CT reconstruction. PatchNR is a
 716 powerful regularizer that involves Glow coupling layers learned on small patches extracted from
 717 very few images (only six images), which has shown promising results for CT reconstruction [3].
 718 PatchNR uses five GlowCoupling blocks and permutations in an alternating manner, where
 719 the coupling blocks are from the FrEIA package [6]. The three-layer subnetworks are fully
 720 connected with ReLU activation functions and 512 nodes, which overall result in a much
 721 smaller network than Glow. It should be noted that extracting the patches from an image is
 722 not a reversible process, therefore patchNR actually learns the prior over the image patches
 723 and cannot do unconditional sampling using $x = T(z)$. Even so, the log gradient is still
 724 computable and Lipschitz continuous, since its GlowCoupling blocks satisfy Proposition 3.9.

725 The patchNR we used is given by the pre-trained model⁵ trained on six images from the
 726 LoDoPaB dataset [60]. For PnP-ULA, we train the denoiser realSN-DnCNN on a 128-image
 727 subset of LoDoPaB, by adding Gaussian noise with the variance $\varepsilon = (5/255)^2$ on the training
 728 data batches. We train a 17-layer realSN-DnCNN on the preprocessed image patches with
 729 size 40×40 . The Lipschitz parameter of the realSN-DnCNN is set to 1. The patchNR has
 730 2908880 parameters in total and the realSN-DnCNN has 556032 parameters.

731 **ULA parameters settings:** While in [3] $\alpha = 700$ is the default setting of the considered
 732 maximum a posteriori estimator, $\alpha = 5000$ (Gaussian noise) works fine for NF-ULA. For PnP-
 733 ULA we set $\alpha = 3$. We use a smaller step size for both algorithms, namely $\delta = 10^{-6}$, to ensure
 734 convergence, since in CT reconstruction the forward operator A has a larger norm (approx-
 735 imately 100) than deblurring and inpainting. The convex set is set to be $C = [-100, 100]^d$.
 736 We initialize X_0 using the filtered back-projection (FBP) reconstruction.

737 **Gaussian noise-corrupted measurement:** We first test the case with additive Gaussian
 738 noise. To be more specific, we add Gaussian noise $n \sim \mathcal{N}(0, \sigma^2 I^m)$ in (4.2) to the clean
 739 projection data. Since $\|A\| \approx 100$, we select $\sigma = 1.0$ to simulate the noisy sinogram y . The

⁵<https://github.com/FabianAltekrueger/patchNR>

740 likelihood can be expressed as

$$741 \quad (4.3) \quad p(y|x) = \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{\|y - Ax\|^2}{2\sigma^2}\right).$$

742 Since the gradient of the log-likelihood is not globally Lipschitz for Poisson likelihood, the
 743 additional experiments with Poisson noise are moved to Appendix B. Note that NF-ULA
 744 with Poisson likelihood still converges although the assumptions needed for the theoretical
 745 guarantees do not hold, which warrants further investigations and we leave it for future work.

746 **Performance of the algorithms:** We test PnP-ULA and NF-ULA on another four images
 747 from LoDoPaB [60] which were not used for training the patchNR network utilized by NF-
 748 ULA and the realSN-DnCNN denoiser used in PnP-ULA. They are different from the six
 749 images trained by patchNR and 128 images trained by realSN-DnCNN. The four ground-
 750 truth images used for evaluating the performance of NF-ULA and PnP-ULA for limited-angle
 751 CT are shown in the first column of Fig. 6.

752 Both PnP-ULA and NF-ULA have more than 20000 burn-in iterations. Since we initialize
 753 by setting X_0 equal to the FBP reconstruction, the PSNR of X_n starts from around 21.90 dB,
 754 then slowly increases, and finally stabilizes. Note that for different test images, the burn-in
 755 time varies. For Image-2 in Table 4, PnP-ULA has 30000 burn-in iterations, and the PSNR of
 756 the samples never exceeds 29 dB. In contrast, the PSNR of the samples increases until 33 dB
 757 for NF-ULA and finally the burn-in time for NF-ULA on Image-2 is around 70000 iterations.

758 After the burn-in time, we calculate the posterior mean and the standard deviation around
 759 it by obtaining 10000 samples and computing the PSNR of the samples' mean. For Gaussian
 760 noise, drawing 10000 samples by NF-ULA takes around 500 seconds, whereas, for PnP-ULA,
 761 it takes about 70 seconds. Thanks to the smaller network size of patchNR compared to Glow,
 762 it saves a large proportion of time in computation.

763 Fig. 6 shows the ground-truth images (1st column), the FBP (2nd column), the posterior
 764 mean and standard deviation of PnP-ULA (in Columns 3 and 4, respectively), and those
 765 corresponding to NF-ULA (in Columns 5 and 6, respectively). The posterior mean images
 766 indicate that NF-ULA has a significantly better sample quality than PnP-ULA, which exhibits
 767 poor reconstruction in the left area, due to the missing angles and the extremely ill-posed
 768 problem. NF-ULA can recover the details well, which is consistent with the results in [3] that
 769 patchNR works well in the limited-angle CT experiments. For standard deviation in the case
 770 of Gaussian noise, NF-ULA shows more realistic uncertainties than PnP-ULA in most areas
 771 but still has relatively large uncertainties in the left area (where no projection is available).
 772 Table 4 shows the PSNR of the posterior mean. NF-ULA achieves a considerably higher
 773 PSNR than PnP-ULA.

774 We also compare the ACF (Image-1) in Fig. 7 to study the convergence speed. The ACF
 775 is calculated by randomly selecting 100 dimensions respectively from YH and YL. The ACF on
 776 the fast direction is different from deblurring and inpainting: On fastest directions NF-ULA
 777 decreases from 1 to 0 within 100 lags and the independence is achieved, while the independence
 778 of PnP-ULA is not achieved (as shown in the first sub-figure). On some fast directions, the
 779 independence of NF-ULA and PnP-ULA is both not well achieved, as demonstrated in the
 780 second sub-figure. For slow directions, both two algorithms decrease slowly and independence

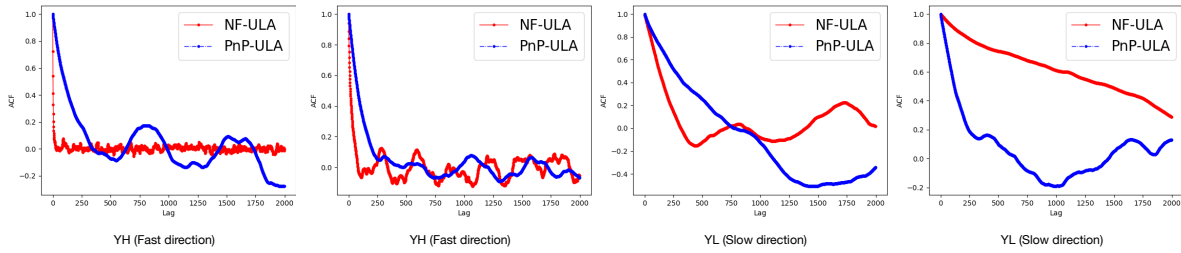


Figure 7. The autocorrelation function (ACF) of the samples (Gaussian noise CT on Image-1). The definition of ACF is given in (4.1). ACF is calculated by wavelet basis using the band-pass coefficients (YH) and the low-pass coefficients (YL). Faster decreasing ACF implies faster convergence of the Markov chain. On slow directions, the independence are not achieved for both algorithms.

781 is not achieved. ACF of Image-2, Image-3 and Image-4 are similar and omitted.

782 **5. Conclusion and Outlook.** We introduced NF-ULA, a Langevin diffusion-based Monte
 783 Carlo algorithm, which takes advantage of a normalizing flow for prior density estimation. The
 784 normalizing flow can be pre-trained agnostic to the forward operator of the inverse problem
 785 that one seeks to solve. Since NF-ULA only requires the log gradient of the prior, our algorithm
 786 still works in cases where the normalizing flow can only evaluate the density but cannot
 787 do unconditional sampling. To guarantee that the posterior distribution is well-defined, we
 788 follow [56] to add a projection operator onto a convex and compact subset of the image space,
 789 although in most cases the projection is not activated, for instance, if the prior is well-trained.
 790 Since the density of normalizing flow itself can be evaluated, NF-ULA can be extended to
 791 a Metropolis-adjusted version, which is left for future studies. For the theoretical analysis
 792 of NF-ULA, we first prove the well-posedness of the posterior distribution that we aim to
 793 draw samples from. To prove the convergence of NF-ULA, the most essential condition is
 794 the Lipschitz drift, and we, therefore, derive a sufficient condition for having a Lipschitz-
 795 continuous gradient of the log density of the normalizing flow. Moreover, we show that
 796 NF-ULA admits a **unique** invariant distribution, and we give a non-asymptotic bound on
 797 the bias. We demonstrate our method through several Bayesian imaging experiments, namely
 798 image deblurring, image inpainting, and limited-angle CT reconstruction. We show that
 799 better training of the normalizing flows leads to better samples and convergence of NF-ULA.
 800 Although currently, NF-ULA has a longer sampling time because of the large network of
 801 normalizing flows, it has the potential to use a better and smaller network to reduce the
 802 computation in the future.

803 There are still some unanswered questions about NF-ULA. Although we give a sufficient
 804 condition for the gradient of the log density of normalizing flow to be Lipschitz, the condition
 805 might be relaxed, or it might even be possible to derive a condition that is both necessary
 806 and sufficient. Moreover, given different curvature conditions [22, 65] on the drift other than
 807 Lipschitz, the studies of ULA on non-convex potentials have shown different convergence
 808 results and they can also be applied to NF-ULA. However, this might require re-training the
 809 normalizing flows to enforce such conditions and necessitates further research. Meanwhile
 810 when the Lipschitz assumption does not hold, the results of our Poisson noise experiments

811 lack an explanation, which also requires a more detailed study.

812 **Acknowledgments.** CBS acknowledges support from the Philip Leverhulme Prize, the
 813 Royal Society Wolfson Fellowship, the EPSRC advanced career fellowship EP/V029428/1,
 814 EPSRC grants EP/S026045/1 and EP/T003553/1, EP/N014588/1, EP/T017961/1, the Well-
 815 come Innovator Awards 215733/Z/19/Z and 221633/Z/20/Z, the European Union Horizon
 816 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No.
 817 777826 NoMADS, the Cantab Capital Institute for the Mathematics of Information and the
 818 Alan Turing Institute. ZC and XZ were partially supported by NSFC (No. 12090024) and
 819 Sino-German center grant (No.M-0187) .

820

REFERENCES

- 821 [1] J. ADLER AND O. ÖKTEM, *Learned primal-dual reconstruction*, IEEE transactions on medical imaging,
 822 37 (2018), pp. 1322–1332.
- 823 [2] C. AGUERREBERE, A. ALMANSA, J. DELON, Y. GOUSSEAU, AND P. MUSÉ, *A bayesian hyperprior*
 824 *approach for joint image denoising and interpolation, with an application to hdr imaging*, IEEE
 825 Transactions on Computational Imaging, 3 (2017), pp. 633–646.
- 826 [3] F. ALTEKRÜGER, A. DENKER, P. HAGEMANN, J. HERTRICH, P. MAASS, AND G. STEIDL,
 827 *Patchnr: Learning from very few images by patch normalizing flow regularization*, arXiv preprint
 828 arXiv:2205.12021, (2022).
- 829 [4] F. ALTEKRÜGER, P. HAGEMANN, AND G. STEIDL, *Conditional generative models are provably robust:*
 830 *Pointwise guarantees for bayesian inverse problems*, arXiv preprint arXiv:2303.15845, (2023).
- 831 [5] B. AMOS, L. XU, AND J. Z. KOLTER, *Input convex neural networks*, in International Conference on
 832 Machine Learning, PMLR, 2017, pp. 146–155.
- 833 [6] L. ARDIZZONE, T. BUNGERT, F. DRAXLER, U. KÖTHE, J. KRUSE, R. SCHMIER, AND P. SORRENSON,
 834 *Framework for Easily Invertible Architectures (FrEIA)*, 2018–2022, [https://github.com/vislearn/](https://github.com/vislearn/FrEIA)
 835 FrEIA.
- 836 [7] S. ARRIDGE, P. MAASS, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Solving inverse problems using data-driven*
 837 *models*, Acta Numerica, 28 (2019), pp. 1–174.
- 838 [8] M. ASIM, M. DANIELS, O. LEONG, A. AHMED, AND P. HAND, *Invertible generative models for inverse*
 839 *problems: mitigating representation error and dataset bias*, in International Conference on Machine
 840 Learning, PMLR, 2020, pp. 399–409.
- 841 [9] H. H. BAUSCHKE, P. L. COMBETTES, ET AL., *Convex analysis and monotone operator theory in Hilbert*
 842 *spaces*, vol. 408, Springer, 2011.
- 843 [10] M. BENNING AND M. BURGER, *Modern regularization methods for inverse problems*, Acta Numerica, 27
 844 (2018), pp. 1–111.
- 845 [11] A. BLAKE, P. KOHLI, AND C. ROTHER, *Markov random fields for vision and image processing*, MIT
 846 press, 2011.
- 847 [12] D. M. BLEI, A. KUCUKELBIR, AND J. D. MCAULIFFE, *Variational inference: A review for statisticians*,
 848 Journal of the American statistical Association, 112 (2017), pp. 859–877.
- 849 [13] A. CHAMBOLLE, V. CASELLES, D. CREMERS, M. NOVAGA, AND T. POCK, *An introduction to total*
 850 *variation for image analysis*, Theoretical foundations and numerical methods for sparse recovery, 9
 851 (2010), p. 227.
- 852 [14] X. CHENG, N. S. CHATTERJI, Y. ABBASI-YADKORI, P. L. BARTLETT, AND M. I. JORDAN, *Sharp*
 853 *convergence rates for langevin dynamics in the nonconvex setting*, arXiv preprint arXiv:1805.01648,
 854 (2018).
- 855 [15] F. COEURDOUX, N. DOBIGEON, AND P. CHAINAIS, *Normalizing flow sampling with langevin dynamics*
 856 *in the latent space*, arXiv preprint arXiv:2305.12149, (2023).
- 857 [16] F. COEURDOUX, N. DOBIGEON, AND P. CHAINAIS, *Plug-and-play split gibbs sampler: embedding deep*
 858 *generative priors in bayesian inference*, arXiv preprint arXiv:2304.11134, (2023).

- 859 [17] P. L. COMBETTES AND J.-C. PESQUET, *Proximal splitting methods in signal processing*, Fixed-point
860 algorithms for inverse problems in science and engineering, (2011), pp. 185–212.
- 861 [18] P. L. COMBETTES AND V. R. WAJS, *Signal recovery by proximal forward-backward splitting*, Multiscale
862 Modeling & Simulation, 4 (2005), pp. 1168–1200.
- 863 [19] A. CRESWELL, T. WHITE, V. DUMOULIN, K. ARULKUMARAN, B. SENGUPTA, AND A. A. BHARATH,
864 *Generative adversarial networks: An overview*, IEEE signal processing magazine, 35 (2018), pp. 53–
865 65.
- 866 [20] A. S. DALALYAN, *Theoretical guarantees for approximate sampling from smooth and log-concave densi-
867 ties*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 79 (2017), pp. 651–
868 676.
- 869 [21] A. S. DALALYAN AND A. KARAGULYAN, *User-friendly guarantees for the langevin monte carlo with
870 inaccurate gradient*, Stochastic Processes and their Applications, 129 (2019), pp. 5278–5311.
- 871 [22] V. DE BORTOLI AND A. DURMUS, *Convergence of diffusions and their discretizations: from continuous
872 to discrete processes and back*, arXiv preprint arXiv:1904.09808, (2019).
- 873 [23] P. DHARIWAL AND A. NICHOL, *Diffusion models beat gans on image synthesis*, Advances in neural
874 information processing systems, 34 (2021), pp. 8780–8794.
- 875 [24] L. DINH, D. KRUEGER, AND Y. BENGIO, *Nice: Non-linear independent components estimation*, arXiv
876 preprint arXiv:1410.8516, (2014).
- 877 [25] L. DINH, J. SOHL-DICKSTEIN, AND S. BENGIO, *Density estimation using real nvp*, arXiv preprint
878 arXiv:1605.08803, (2016).
- 879 [26] R. DOUC, E. MOULINES, P. PRIOURET, AND P. SOULIER, *Markov chains*, Springer, 2018.
- 880 [27] A. DURMUS, S. MAJEWSKI, AND B. MIASOJEDOW, *Analysis of langevin monte carlo via convex opti-
881 mization*, The Journal of Machine Learning Research, 20 (2019), pp. 2666–2711.
- 882 [28] A. DURMUS AND E. MOULINES, *Nonasymptotic convergence analysis for the unadjusted langevin algo-
883 rithm*, The Annals of Applied Probability, 27 (2017), pp. 1551–1587.
- 884 [29] A. DURMUS, E. MOULINES, AND M. PEREYRA, *Efficient bayesian computation by proximal markov
885 chain monte carlo: when langevin meets moreau*, SIAM Journal on Imaging Sciences, 11 (2018),
886 pp. 473–506.
- 887 [30] B. EFRON, *Tweedie’s formula and selection bias*, Journal of the American Statistical Association, 106
888 (2011), pp. 1602–1614.
- 889 [31] M. A. ERDOGDU AND R. HOSSEINZADEH, *On the convergence of langevin monte carlo: The interplay
890 between tail growth and smoothness*, in Conference on Learning Theory, PMLR, 2021, pp. 1776–1822.
- 891 [32] W. R. GILKS, S. RICHARDSON, AND D. SPIEGELHALTER, *Markov chain Monte Carlo in practice*, CRC
892 press, 1995.
- 893 [33] D. GILTON, G. ONGIE, AND R. WILLETT, *Learned patch-based regularization for inverse problems in
894 imaging*, in 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor
895 Adaptive Processing (CAMSAP), IEEE, 2019, pp. 211–215.
- 896 [34] D. GILTON, G. ONGIE, AND R. WILLETT, *Deep equilibrium architectures for inverse problems in imaging*,
897 IEEE Transactions on Computational Imaging, 7 (2021), pp. 1123–1133.
- 898 [35] I. GOODFELLOW, J. POUGET-ABADIE, M. MIRZA, B. XU, D. WARDE-FARLEY, S. OZAI, A. COURVILLE,
899 AND Y. BENGIO, *Generative adversarial nets*, Advances in neural information pro-
900 cessing systems, 27 (2014).
- 901 [36] P. HAGEMANN, J. HERTRICH, AND G. STEIDL, *Stochastic normalizing flows for inverse problems: a
902 markov chains viewpoint*, SIAM/ASA Journal on Uncertainty Quantification, 10 (2022), pp. 1162–
903 1190.
- 904 [37] P. HAGEMANN AND S. NEUMAYER, *Stabilizing invertible neural networks using mixture models*, Inverse
905 Problems, 37 (2021), p. 085002.
- 906 [38] J. HERTRICH, S. NEUMAYER, AND G. STEIDL, *Convolutional proximal neural networks and plug-and-play
907 algorithms*, Linear Algebra and its Applications, 631 (2021), pp. 203–234.
- 908 [39] J. HO, A. JAIN, AND P. ABBEEL, *Denoising diffusion probabilistic models*, Advances in neural informa-
909 tion processing systems, 33 (2020), pp. 6840–6851.
- 910 [40] M. D. HOFFMAN, D. M. BLEI, C. WANG, AND J. PAISLEY, *Stochastic variational inference*, Journal of
911 Machine Learning Research, (2013).
- 912 [41] A. HOUDARD, C. BOUVEYRON, AND J. DELON, *High-dimensional mixture models for unsupervised image*

- 913 *denoising (hdmi)*, SIAM Journal on Imaging Sciences, 11 (2018), pp. 2815–2846.
- 914 [42] S. HURAUULT, A. LECLAIRE, AND N. PAPADAKIS, *Gradient step denoiser for convergent plug-and-play*,
915 in International Conference on Learning Representations, 2022.
- 916 [43] N. IKEDA AND S. WATANABE, *Stochastic differential equations and diffusion processes*, Elsevier, 2014.
- 917 [44] P. JAINI, I. KOBYZEV, Y. YU, AND M. BRUBAKER, *Tails of lipschitz triangular flows*, in International
918 Conference on Machine Learning, PMLR, 2020, pp. 4673–4681.
- 919 [45] K. H. JIN, M. T. McCANN, E. FROUSTEY, AND M. UNSER, *Deep convolutional neural network for*
920 *inverse problems in imaging*, IEEE Transactions on Image Processing, 26 (2017), pp. 4509–4522.
- 921 [46] J. KAPIO AND E. SOMERSALO, *Statistical and computational inverse problems*, vol. 160, Springer Science
922 & Business Media, 2006.
- 923 [47] U. S. KAMILOV, C. A. BOUMAN, G. T. BUZZARD, AND B. WOHLBERG, *Plug-and-play methods for inte-*
924 *grating physical and learned models in computational imaging: Theory, algorithms, and applications*,
925 IEEE Signal Processing Magazine, 40 (2023), pp. 85–97.
- 926 [48] I. KARATZAS, I. KARATZAS, S. SHREVE, AND S. E. SHREVE, *Brownian motion and stochastic calculus*,
927 vol. 113, Springer Science & Business Media, 1991.
- 928 [49] T. KARRAS, S. LAINE, AND T. AILA, *A style-based generator architecture for generative adversarial*
929 *networks*, in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,
930 2019, pp. 4401–4410.
- 931 [50] D. P. KINGMA AND P. DHARIWAL, *Glow: Generative flow with invertible 1x1 convolutions*, Advances
932 in neural information processing systems, 31 (2018).
- 933 [51] D. P. KINGMA, T. SALIMANS, R. JOZEFOWICZ, X. CHEN, I. SUTSKEVER, AND M. WELLING, *Improved*
934 *variational inference with inverse autoregressive flow*, Advances in neural information processing
935 systems, 29 (2016).
- 936 [52] D. P. KINGMA AND M. WELLING, *Auto-encoding variational bayes*, arXiv preprint arXiv:1312.6114,
937 (2013).
- 938 [53] E. KOBLER, A. EFFLAND, K. KUNISCH, AND T. POCK, *Total deep variation: A stable regularization*
939 *method for inverse problems*, IEEE transactions on pattern analysis and machine intelligence, 44
940 (2021), pp. 9163–9180.
- 941 [54] I. KOBYZEV, S. J. PRINCE, AND M. A. BRUBAKER, *Normalizing flows: An introduction and review*
942 *of current methods*, IEEE transactions on pattern analysis and machine intelligence, 43 (2020),
943 pp. 3964–3979.
- 944 [55] J. LATZ, *On the well-posedness of bayesian inverse problems*, SIAM/ASA Journal on Uncertainty Quan-
945 tification, 8 (2020), pp. 451–482.
- 946 [56] R. LAUMONT, V. D. BORTOLI, A. ALMANSA, J. DELON, A. DURMUS, AND M. PEREYRA, *Bayesian*
947 *imaging using plug & play priors: when langevin meets tweedie*, SIAM Journal on Imaging Sciences,
948 15 (2022), pp. 701–737.
- 949 [57] R. LAUMONT, V. D. BORTOLI, A. ALMANSA, J. DELON, A. DURMUS, AND M. PEREYRA, *Supplementary*
950 *materials: Bayesian imaging using plug & play priors: when langevin meets tweedie*. https://epubs.siam.org/doi/suppl/10.1137/21M1406349/suppl_file/M140634_01.pdf, 2022.
- 951 [58] J. LEHEC, *The langevin monte carlo algorithm in the non-smooth log-concave case*, arXiv preprint
952 arXiv:2101.10695, (2021).
- 953 [59] V. LEMPITSKY, A. VEDALDI, AND D. ULYANOV, *Deep image prior*, in 2018 IEEE/CVF Conference on
954 Computer Vision and Pattern Recognition, IEEE, 2018, pp. 9446–9454.
- 955 [60] J. LEUSCHNER, M. SCHMIDT, D. O. BAGUER, AND P. MAASS, *Lodopab-ct, a benchmark dataset for*
956 *low-dose computed tomography reconstruction*, Scientific Data, 8 (2021), p. 109.
- 957 [61] Q. LIU AND D. WANG, *Stein variational gradient descent: A general purpose bayesian inference algo-*
958 *rihm*, Advances in neural information processing systems, 29 (2016).
- 959 [62] C. LOUCHET AND L. MOISAN, *Posterior expectation of the total variation model: properties and exper-*
960 *iments*, SIAM Journal on Imaging Sciences, 6 (2013), pp. 2640–2684.
- 961 [63] S. LUNZ, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Adversarial regularizers in inverse problems*, Advances in
962 neural information processing systems, 31 (2018).
- 963 [64] T. D. LUU, J. FADILI, AND C. CHESNEAU, *Sampling from non-smooth distributions through langevin*
964 *diffusion*, Methodology and Computing in Applied Probability, 23 (2021), pp. 1173–1201.
- 965 [65] M. B. MAJKA, A. MIJATOVIĆ, AND ŁUKASZ SZPRUCH, *Nonasymptotic bounds for sampling algorithms*
966

- 967 *without log-concavity*, The Annals of Applied Probability, 30 (2020), pp. 1534 – 1581, <https://doi.org/10.1214/19-AAP1535>.
- 968
- 969 [66] S. P. MEYN AND R. L. TWEEDIE, *Stability of markovian processes iii: Foster-lyapunov criteria for*
- 970 *continuous-time processes*, Advances in Applied Probability, 25 (1993), pp. 518–548.
- 971 [67] V. MONGA, Y. LI, AND Y. C. ELДАР, *Algorithm unrolling: Interpretable, efficient deep learning for*
- 972 *signal and image processing*, IEEE Signal Processing Magazine, 38 (2021), pp. 18–44.
- 973 [68] W. MOU, N. FLAMMARION, M. J. WAINWRIGHT, AND P. L. BARTLETT, *An efficient sampling algorithm*
- 974 *for non-smooth composite potentials*, Journal of Machine Learning Research, 23 (2022), pp. 1–50.
- 975 [69] S. MUKHERJEE, M. CARIONI, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *End-to-end reconstruction meets data-*
- 976 *driven regularization for inverse problems*, Advances in Neural Information Processing Systems, 34
- 977 (2021), pp. 21413–21425.
- 978 [70] S. MUKHERJEE, S. DITTMER, Z. SHUMAYLOV, S. LUNZ, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Learned*
- 979 *convex regularizers for inverse problems*, arXiv preprint arXiv:2008.02839, (2020).
- 980 [71] R. NEAL, *Bayesian learning via stochastic dynamics*, Advances in neural information processing systems,
- 981 5 (1992).
- 982 [72] G. ONGIE, A. JALAL, C. A. METZLER, R. G. BARANIUK, A. G. DIMAKIS, AND R. WILLETT, *Deep*
- 983 *learning techniques for inverse problems in imaging*, IEEE Journal on Selected Areas in Information
- 984 Theory, 1 (2020), pp. 39–56.
- 985 [73] X. PAN, X. ZHAN, B. DAI, D. LIN, C. C. LOY, AND P. LUO, *Exploiting deep generative prior for*
- 986 *versatile image restoration and manipulation*, IEEE Transactions on Pattern Analysis and Machine
- 987 Intelligence, 44 (2021), pp. 7474–7489.
- 988 [74] G. PAPAMAKARIOS, E. T. NALISNICK, D. J. REZENDE, S. MOHAMED, AND B. LAKSHMINARAYANAN,
- 989 *Normalizing flows for probabilistic modeling and inference.*, J. Mach. Learn. Res., 22 (2021), pp. 1–
- 990 64.
- 991 [75] G. PAPAMAKARIOS, T. PAVLAKOU, AND I. MURRAY, *Masked autoregressive flow for density estimation*,
- 992 Advances in neural information processing systems, 30 (2017).
- 993 [76] M. PEREYRA, *Proximal markov chain monte carlo algorithms*, Statistics and Computing, 26 (2016),
- 994 pp. 745–760.
- 995 [77] J.-C. PESQUET, A. REPETTI, M. TERRIS, AND Y. WIAUX, *Learning maximally monotone operators for*
- 996 *image recovery*, SIAM Journal on Imaging Sciences, 14 (2021), pp. 1206–1237.
- 997 [78] J. PROST, A. HOUDARD, A. ALMANSA, AND N. PAPADAKIS, *Learning local regularization for variational*
- 998 *image restoration*, in Scale Space and Variational Methods in Computer Vision: 8th International
- 999 Conference, SSVN 2021, Virtual Event, May 16–20, 2021, Proceedings, Springer, 2021, pp. 358–370.
- 1000 [79] D. REZENDE AND S. MOHAMED, *Variational inference with normalizing flows*, in International conference
- 1001 on machine learning, PMLR, 2015, pp. 1530–1538.
- 1002 [80] G. O. ROBERTS AND R. L. TWEEDIE, *Exponential convergence of langevin distributions and their discrete*
- 1003 *approximations*, Bernoulli, (1996), pp. 341–363.
- 1004 [81] Y. ROMANO, M. ELAD, AND P. MILANFAR, *The little engine that could: Regularization by denoising*
- 1005 *(red)*, SIAM Journal on Imaging Sciences, 10 (2017), pp. 1804–1844.
- 1006 [82] M. RONCHETTI, *Torchradon: Fast differentiable routines for computed tomography*, arXiv preprint
- 1007 arXiv:2009.14788, (2020), <https://arxiv.org/abs/arXiv:2009.14788>.
- 1008 [83] O. RONNEBERGER, P. FISCHER, AND T. BROX, *U-net: Convolutional networks for biomedical image*
- 1009 *segmentation*, in Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015:
- 1010 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18,
- 1011 Springer, 2015, pp. 234–241.
- 1012 [84] E. RYU, J. LIU, S. WANG, X. CHEN, Z. WANG, AND W. YIN, *Plug-and-play methods provably converge*
- 1013 *with properly trained denoisers*, in International Conference on Machine Learning, PMLR, 2019,
- 1014 pp. 5546–5557.
- 1015 [85] A. SALIM, D. KOVALEV, AND P. RICHTÁRIK, *Stochastic proximal langevin algorithm: Potential splitting*
- 1016 *and nonasymptotic rates*, Advances in Neural Information Processing Systems, 32 (2019).
- 1017 [86] A. SALMONA, V. DE BORTOLI, J. DELON, AND A. DESOLNEUX, *Can push-forward generative mod-*
- 1018 *els fit multimodal distributions?*, Advances in Neural Information Processing Systems, 35 (2022),
- 1019 pp. 10766–10779.
- 1020 [87] Y. SONG, L. SHEN, L. XING, AND S. ERMON, *Solving inverse problems in medical imaging with score-*

- 1021 *based generative models*, arXiv preprint arXiv:2111.08005, (2021).
- 1022 [88] Y. SONG, J. SOHL-DICKSTEIN, D. P. KINGMA, A. KUMAR, S. ERMON, AND B. POOLE, *Score-based generative modeling through stochastic differential equations*, arXiv preprint arXiv:2011.13456, (2020).
- 1023 [89] B. SPRUNGK, *On the local lipschitz stability of bayesian inverse problems*, Inverse Problems, 36 (2020), p. 055015.
- 1024 [90] S. SREEHARI, S. V. VENKATAKRISHNAN, B. WOHLBERG, G. T. BUZZARD, L. F. DRUMMY, J. P. SIMMONS, AND C. A. BOUMAN, *Plug-and-play priors for bright field electron tomography and sparse interpolation*, IEEE Transactions on Computational Imaging, 2 (2016), pp. 408–423.
- 1025 [91] G. STEIDL, P. L. HAGEMANN, AND J. HERTRICH, *Generalized normalizing flows via markov chains*, Elements in Non-local Data Interactions: Foundations and Applications, (2022).
- 1026 [92] O. STRAMER AND R. TWEEDIE, *Langevin-type models i: Diffusions with given stationary distributions and their discretizations*, Methodology and Computing in Applied Probability, 1 (1999), pp. 283–306.
- 1027 [93] A. M. STUART, *Inverse problems: a bayesian perspective*, Acta numerica, 19 (2010), pp. 451–559.
- 1028 [94] H. Y. TAN, S. MUKHERJEE, J. TANG, AND C.-B. SCHÖNLIEB, *Provably convergent plug-and-play quasi-newton methods*, arXiv preprint arXiv:2303.07271, (2023).
- 1029 [95] A. TARANTOLA, *Inverse problem theory and methods for model parameter estimation*, SIAM, 2005.
- 1030 [96] S. V. VENKATAKRISHNAN, C. A. BOUMAN, AND B. WOHLBERG, *Plug-and-play priors for model based reconstruction*, in 2013 IEEE Global Conference on Signal and Information Processing, IEEE, 2013, pp. 945–948.
- 1031 [97] A. VERINE, B. NEGREVERGNE, F. ROSSI, AND Y. CHEVALEYRE, *On the expressivity of bi-lipschitz normalizing flows*, arXiv preprint arXiv:2107.07232, (2021).
- 1032 [98] C. VILLANI ET AL., *Optimal transport: old and new*, vol. 338, Springer, 2009.
- 1033 [99] J. WHANG, Q. LEI, AND A. DIMAKIS, *Solving inverse problems with a flow-based noise model*, in International Conference on Machine Learning, PMLR, 2021, pp. 11146–11157.
- 1034 [100] H. WU, J. KÖHLER, AND F. NOÉ, *Stochastic normalizing flows*, Advances in Neural Information Processing Systems, 33 (2020), pp. 5933–5944.
- 1035 [101] L. YANG, Z. ZHANG, Y. SONG, S. HONG, R. XU, Y. ZHAO, Y. SHAO, W. ZHANG, B. CUI, AND M.-H. YANG, *Diffusion models: A comprehensive survey of methods and applications*, arXiv preprint arXiv:2209.00796, (2022).
- 1036 [102] G. YU, G. SAPIRO, AND S. MALLAT, *Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity*, IEEE Transactions on Image Processing, 21 (2011), pp. 2481–2499.
- 1037 [103] K. ZHANG, Y. LI, W. ZUO, L. ZHANG, L. VAN GOOL, AND R. TIMOFTE, *Plug-and-play image restoration with deep denoiser prior*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (2021), pp. 6360–6376.
- 1038 [104] K. ZHANG, W. ZUO, Y. CHEN, D. MENG, AND L. ZHANG, *Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising*, IEEE transactions on image processing, 26 (2017), pp. 3142–3155.
- 1039 [105] D. ZORAN AND Y. WEISS, *From learning models of natural image patches to whole image restoration*, in 2011 international conference on computer vision, IEEE, 2011, pp. 479–486.

1061 Appendix A. Proofs.

1062 A.1. Proof of Lemma 3.1.

1063 *Proof.* For a constant $R_0 > 0$, let

$$1064 \quad B(0, R_0) := \left\{ z \in \mathbb{R}^d : \|z\|_2 \leq R_0 \right\}$$

1065 be the closed ball of radius R_0 centered at the origin. Since $C \subset \mathbb{R}^d$ is compact, there exists
 1066 $R_0 > 0$ such that $C \subset B(0, R_0)$. Therefore, for all $x \notin B(0, R_0)$, it follows that

$$1067 \quad \|x - \Pi_C(x)\|_2 \stackrel{(a)}{\geq} \|x - \Pi_{B(0, R_0)}(x)\|_2 \stackrel{(b)}{\geq} \|x\|_2 - R_0 \geq 0,$$

1069 where (a) is true since $C \subset B(0, R_0)$ and (b) follows from the triangle inequality. Then, for
 1070 all $k \in \mathbb{N}$, the following holds:

$$\begin{aligned}
 & \int_{\mathbb{R}^d \setminus B(0, R_0)} \|x\|^k \exp\left(-\frac{\|x - \Pi_C(x)\|_2^2}{2\lambda}\right) dx \\
 1071 & \leq \int_{\mathbb{R}^d \setminus B(0, R_0)} \|x\|^k \exp\left(-\frac{(\|x\|_2 - R_0)^2}{2\lambda}\right) dx \\
 & \leq \int_{\mathbb{R}^d \setminus B(0, R_0)} \|x\|^k \exp\left(-\frac{\|x\|_2^2 - 2R_0^2}{4\lambda}\right) dx \\
 & < +\infty,
 \end{aligned}$$

1072 where the last inequality follows from the fact that k -order moments of Gaussian distribution
 1073 are finite for any k . ■

1074 **A.2. Proof of Proposition 3.3.**

1075 *Proof.* Without loss of generality, we only need to consider the cases when the total number
 1076 of layers is $k = 1, 2$.

1077 (1) We firstly consider the case that $k = 1$ and $T^{-1} = G$ is a composition of only a
 1078 one-layer coupling network. Then (3.4) can be simplified as:

$$1079 \quad (\text{A.1}) \quad G_j(x_j, x_{<j}) = \varphi_j(x_{<j})x_j + \eta_j(x_{<j}), \quad j = 1, \dots, d.$$

1080 Since $\forall r < j$, G_r is independent of x_j and the diagonal of the Jacobian is $(J_G(x))_{j,j} = \varphi_j(x_{<j})$,
 1081 from the change of variables

$$\begin{aligned}
 1082 \quad (\text{A.2}) \quad q(x) &= q_z(z) |\det J_T(z)|^{-1} \\
 &= q_z(T^{-1}(x)) |\det J_{T^{-1}}(x)|,
 \end{aligned}$$

1083 we have that

$$\begin{aligned}
 & \log q_\theta(x) = \log q_z(G(x)) + \log |\det J_G(x)| \\
 & = -\frac{1}{2} \|G(x)\|_2^2 + \log |\det J_G(x)| + \text{const.} \\
 1084 & = -\frac{1}{2} \|G(x)\|_2^2 + \sum_{j=1}^d \log |\varphi_j(x_{<j})| + \text{const.} \\
 & \leq \sum_{j=1}^d \log |\varphi_j(x_{<j})| + \text{const.}
 \end{aligned}$$

1085 Since φ_j is a bounded function $\forall j$, it follows that $\log |\varphi_j(x_{<j})|$ is upper bounded for all j and
 1086 $\log q_\theta(x)$ is upper bounded on \mathbb{R}^d .

1087 (2) Secondly, assume that $k = 2$ and $T^{-1} = G \circ H(x)$, where $H : x \mapsto \omega$ and $G : \omega \mapsto z$.
 1088 Similarly, we have that

$$\begin{aligned}
 \log q_\theta(x) &= \log q_z(G \circ H(x)) + \log |\det J_{G \circ H}(x)| \\
 &= -\frac{1}{2} \|G \circ H(x)\|_2^2 + \log |\det J_G(\omega)| + \log |\det J_H(x)| + \text{const.} \\
 1089 \quad &= -\frac{1}{2} \|G \circ H(x)\|_2^2 + \sum_{j=1}^d \left(\log |\varphi_j^{(2)}(\omega_{<j})| + \log |\varphi_j^{(1)}(x_{<j})| \right) + \text{const.} \\
 &\leq \sum_{j=1}^d \left(\log |\varphi_j^{(2)}(\omega_{<j})| + \log |\varphi_j^{(1)}(x_{<j})| \right) + \text{const.}
 \end{aligned}$$

1090 Since $\varphi_j^{(1)}$ and $\varphi_j^{(2)}$ are bounded functions $\forall j$, it follows that $\log |\varphi_j^{(2)}(\omega_{<j})| + \log |\varphi_j^{(1)}(x_{<j})|$
 1091 is upper bounded for all j and $\log q_\theta(x)$ is upper bounded on \mathbb{R}^d . ■

1092 A.3. Proof of Proposition 3.5.

1093 *Proof.* By Assumption 3.2, we have that

$$1094 \quad \int_{\mathbb{R}^d} (1 + \Phi_1(\tilde{x})) \exp \left[c_0 \Phi_1(\tilde{x}) - \iota_C^{(\lambda)}(\tilde{x}) \right] q_\theta^\alpha(\tilde{x}) d\tilde{x} < +\infty,$$

1095 and we conclude the proof from Proposition 2.3 of [56].

1096 A.4. Proof of Lemma 3.7.

1097 **Lemma A.1.** *Let Assumption 3.6 be true. Then, $\nabla \log p_\lambda(x|y)$ is Lipschitz continuous if*
 1098 *and only if $\nabla \log q_\theta(x)$ is Lipschitz continuous.*

1099 *Proof.* Since Assumption 3.6 is satisfied, from Algorithm 2.1 and (2.13) we have that
 1100 $\nabla \log p_\lambda(x|y)$ is Lipschitz continuous if and only if $\alpha \nabla \log q_\theta(x) + (\Pi_C(x) - x)/\lambda$ is Lipschitz
 1101 continuous.

1102 From Proposition 12.28 in [9], the operator $(\text{Id} - \text{Prox}_{\iota_C})$ is firmly non-expansive, i.e., for
 1103 all $x, y \in \mathbb{R}^d$,

$$\begin{aligned}
 1104 \quad \|\Pi_C(x) - x - (\Pi_C(y) - y)\|_2^2 &\leq \langle \Pi_C(x) - x - (\Pi_C(y) - y), x - y \rangle \\
 &\leq \|\Pi_C(x) - x - (\Pi_C(y) - y)\|_2 \|x - y\|_2.
 \end{aligned}$$

1105 Therefore, $(\Pi_C(x) - x)/\lambda$ is $1/\lambda$ -Lipschitz. Hence, for any $\alpha > 0$, $\nabla \log p_\lambda(x|y)$ is Lipschitz-
 1106 continuous if and only if $\nabla \log q_\theta(x)$ is Lipschitz-continuous. ■

1107 A.5. Proof of Proposition 3.9.

1108 *Proof.* Without loss of generality, we only need to consider the cases when the total number
 1109 of layers is $k = 1, 2$.

1110 (1) We firstly consider the case that $k = 1$ and $T^{-1} = G$ is a composition of only a
 1111 one-layer coupling network. Then (3.5) can be simplified as:

$$1112 \quad (\text{A.3}) \quad G_j(x_j, x_{<j}) = \varphi_j(x_{<j})x_j + \eta_j(x_{<j}), \quad j = 1, \dots, d.$$

1113 Since $\forall r < j$, G_r is independent of x_j and the diagonal of the Jacobian is $(J_G(x))_{j,j} = \varphi_j(x_{<j})$,
 1114 from the change of variables

$$\begin{aligned} 1115 \quad (\text{A.4}) \quad q(x) &= q_z(z) |\det J_T(z)|^{-1} \\ &= q_z(T^{-1}(x)) |\det J_{T^{-1}}(x)|, \end{aligned}$$

1116 we have that

$$\begin{aligned} \log q_\theta(x) &= \log q_z(G(x)) + \log |\det J_G(x)| \\ &= -\frac{1}{2} \|G(x)\|_2^2 + \log |\det J_G(x)| + \text{const.} \\ 1117 \quad &= -\frac{1}{2} \|G(x)\|_2^2 + \sum_{j=1}^d \log |\varphi_j(x_{<j})| + \text{const.} \end{aligned}$$

1118 Taking the gradient of both sides w.r.t. x , we get

$$1119 \quad (\text{A.5}) \quad \nabla \log q_\theta(x) = -(J_G(x))^T G(x) + \sum_{j=1}^d \nabla \log \varphi_j(x_{<j}).$$

1120 Since φ_j is a constant function, we have that $\nabla \log \varphi_j = 0$. Furthermore as η_j is Lipschitz and
 1121 $\forall r < j$, $\frac{\partial \eta_j}{\partial x_r}$ is piecewise constant on \mathbb{R} , $\frac{\partial \eta_j}{\partial x_r}$ is hence bounded. Meanwhile, $(J_G(x))_{j,r} = \frac{\partial \eta_j}{\partial x_r}$,
 1122 therefore every element of $J_G(x)$ is a bounded piecewise constant function of x . Then both
 1123 $G(x)$ and $(J_G(x))^T G(x)$ are Lipschitz, therefore $\nabla \log q_\theta(x)$ is Lipschitz.

1124 (2) Secondly, assume that $k = 2$ and $T^{-1} = G \circ H(x)$, where $H : x \mapsto \omega$ and $G : \omega \mapsto z$.
 1125 Similarly, we have that

$$\begin{aligned} \log q_\theta(x) &= \log q_z(G \circ H(x)) + \log |\det J_{G \circ H}(x)| \\ &= -\frac{1}{2} \|G \circ H(x)\|_2^2 + \log |\det J_G(\omega)| + \log |\det J_H(x)| + \text{const.} \\ 1126 \quad &= -\frac{1}{2} \|G \circ H(x)\|_2^2 + \sum_{j=1}^d \left(\log |\varphi_j^{(2)}(\omega_{<j})| + \log |\varphi_j^{(1)}(x_{<j})| \right) + \text{const.} \end{aligned}$$

1127 and

$$\begin{aligned} 1128 \quad (\text{A.6}) \quad \nabla \log q_\theta(x) &= -(J_{G \circ H}(x))^T G \circ H(x) + 0 \\ &= -(J_G(H(x)) J_H(x))^T G \circ H(x). \end{aligned}$$

1129 Since every element of $J_H(x)$ is a bounded piecewise constant function of x , every element of
 1130 $J_G(w)$ is a bounded piecewise constant function of w , and meanwhile $w = H(x)$ is continuous
 1131 w.r.t. x , then every element of $J_{G \circ H}(x)$ is a bounded piecewise constant function of x . Then
 1132 both $G \circ H(x)$ and $(J_{G \circ H}(x))^T G \circ H(x)$ are Lipschitz, therefore $\nabla \log q_\theta(x)$ is Lipschitz. \blacksquare

1133 **A.6. Proof of theorem 3.11.**

 1134 *Proof.* Denote $R_C = \sup \{\|x_1 - x_2\| : x_1, x_2 \in C\}$. Since we have $2\lambda(\alpha L - m_y) \leq 1$, from A
 1135 3.8, A 3.10, $b_\lambda(x)$ in (3.6) and the Cauchy-Schwarz inequality we have that for any $x_1, x_2 \in \mathbb{R}^d$,

1136 (A.7)
$$\begin{aligned} \langle b_\lambda(x_1) - b_\lambda(x_2), x_1 - x_2 \rangle &\leq (-m_y + \alpha L) \|x_1 - x_2\|^2 - \frac{\|x_1 - x_2\|^2}{\lambda} + \frac{R_C \|x_1 - x_2\|}{\lambda} \\ &\leq -\frac{\|x_1 - x_2\|^2}{2\lambda} + \frac{R_C \|x_1 - x_2\|}{\lambda}. \end{aligned}$$

 1137 For any $x_1, x_2 \in \mathbb{R}^d$ satisfying $\|x_1 - x_2\| \geq 4R_C$, we obtain the contractivity at infinity
 1138 condition on the drift b_λ

1139 (A.8)
$$\langle b_\lambda(x_1) - b_\lambda(x_2), x_1 - x_2 \rangle \leq -\frac{\|x_1 - x_2\|^2}{4\lambda},$$

1140 which indicates the strongly convexity at infinity.

 1141 After simple computation by letting $x_2 = 0$ in (A.7), we also have that for any $x \in \mathbb{R}^d$,

1142 (A.9)
$$\langle b_\lambda(x), x \rangle \leq -\|x\|^2/(4\lambda) + \sup_{\tilde{x} \in \mathbb{R}^d} \{(R_C/\lambda + \|b_\lambda(0)\|) \|\tilde{x}\| - \|\tilde{x}\|^2/(4\lambda)\}.$$

 1143 From A 3.6, A 3.8, $b_\lambda(x)$ in (3.6) and that $(\text{Id} - \Pi_C)/\lambda$ is $1/\lambda$ -Lipschitz, we have that for
 1144 any $x_1, x_2 \in \mathbb{R}^d$,

1145 (A.10)
$$\|b_\lambda(x_1) - b_\lambda(x_2)\|_2 \leq (L_y + \alpha L + 1/\lambda) \|x_1 - x_2\|_2.$$

 1146 Let $\bar{\gamma} = (4\lambda)^{-1} (L_y + \alpha L + 1/\lambda)^{-2}$. From (A.9) and (A.10), using Lemma SM5.1 in [57]
 1147 and we get that there exist $\lambda_V \in (0, 1]$, $c \geq 0$ such that for any $\delta \in (0, \bar{\gamma}]$, R_δ satisfies the
 1148 discrete drift condition $\mathbf{D}_d(V, \lambda_V^\delta, c\delta)$.

 1149 For any probability measure ν_1, ν_2 , from the definition (3.1) and Hölder's inequality we
 1150 have that

1151 (A.11)
$$\|\nu_1 - \nu_2\|_V \leq \|\nu_1 - \nu_2\|_{\text{TV}}^{1/2} (\nu_1[V^2] + \nu_2[V^2])^{1/2}.$$

 1152 Since $\bar{\delta} \leq \bar{\gamma}$, the contractivity condition (A.8) holds, (A.11) holds, then from Theorem 8
 1153 and Corollary 2 in [22], we can find $A_2 \geq 0$ and $\rho_2 \in [0, 1)$ such that for any $\delta \in (0, \bar{\delta}]$, $x_1, x_2 \in$
 1154 \mathbb{R}^d , and $k \in \mathbb{N}$,

1155 (A.12)
$$\begin{aligned} \left\| \delta_{x_1} R_\delta^k - \delta_{x_2} R_\delta^k \right\|_{\text{TV}} &\leq A_2 \rho_2^{k\delta} (V(x_1) + V(x_2)) \\ &\leq A_2 \rho_2^{k\delta} (V^2(x_1) + V^2(x_2)), \\ \mathbf{W}_1(\delta_{x_1} R_\delta^k, \delta_{x_2} R_\delta^k) &\leq A_2 \rho_2^{k\delta} \|x_1 - x_2\|_2. \end{aligned}$$

 1156 Then we conclude the proof from (A.11). ■

1157 **A.7. Proof of theorem 3.12.**1158 *Proof.* Most of our proof is based on [57] and [22].

1159 Recall that

1160 (A.13)
$$R_\delta(x, A) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} \mathbf{1}_A \left(x + \delta b_\lambda(x) + \sqrt{2\delta} z \right) \exp \left[-\|z\|^2/2 \right] dz.$$

1161 We introduce the stochastic process $(\bar{\mathbf{X}}_t)_{t \geq 0}$, which is exactly the solution of the following
1162 SDE:

1163 (A.14)
$$\begin{cases} d\bar{\mathbf{X}}_t = b_\lambda(\bar{\mathbf{X}}_t) dt + \sqrt{2} d\mathbf{B}_t \\ b_\lambda(x) = \nabla \log(p(y|x)) + \alpha \nabla \log q_\theta(x) + \frac{\Pi_C(x) - x}{\lambda} \\ \bar{\mathbf{X}}_0 = X_0, \end{cases}$$

1164 where $(\mathbf{B}_t)_{t \geq 0}$ is a d -dimensional Brownian motion.1165 From Lemma 3.7, b_λ is $(L_y + \alpha L + 1/\lambda)$ -Lipschitz continuous. From Chapter 5, Theorem 2.9
1166 of [48] we have that the SDE (A.14) admits a unique strong solution for any initial condition
1167 $\bar{\mathbf{X}}_0$ with $\mathbb{E} \left[\|\bar{\mathbf{X}}_0\|^2 \right] < +\infty$. We denote by $(P_t)_{t \geq 0}$ the semigroup associated with the strong
1168 solutions of SDE (A.14). Similarly to the proof of Theorem 3.11, replacing Corollary 2 in [22]
1169 by Theorem 21 and Corollary 22 in [22], there exist $\tilde{A}_1 \geq 0$ and $\tilde{\rho}_1 \in [0, 1)$ such that that for
1170 any $x_1, x_2 \in \mathbb{R}^d$ and $t \geq 0$,

1171 (A.15)
$$\begin{aligned} \|\delta_{x_1} P_t - \delta_{x_2} P_t\|_V &\leq \tilde{A}_1 \tilde{\rho}_1^t (V^2(x_1) + V^2(x_2)), \\ \mathbf{W}_1(\delta_{x_1} P_t, \delta_{x_2} P_t) &\leq \tilde{A}_1 \tilde{\rho}_1^t \|x_1 - x_2\|_2. \end{aligned}$$

1172 Combining (A.15), Theorem 3.11, the fact that $(\mathcal{P}_1(\mathbb{R}^d), \mathbf{W}_1)$ is a complete metric space
1173 and the Picard fixed point theorem, we can obtain that for any $\delta \in (0, \bar{\delta}]$ there exist **unique**
1174 $\pi_{\delta, \lambda}, \tilde{\pi}_\lambda \in \mathcal{P}_1(\mathbb{R}^d)$ such that $\pi_{\delta, \lambda} R_\delta = \pi_{\delta, \lambda}$ and for any $t \geq 0$, $\tilde{\pi}_\lambda P_t = \tilde{\pi}_\lambda$. By Theorem 2.1
1175 in [80] we have that for any $x \in \mathbb{R}^d$,

1176 (A.16)
$$(d\tilde{\pi}_\lambda/d\text{Leb})(x) \propto \exp \left[-\iota_C^{(\lambda)}(x) \right] p(y|x) p_\lambda^\alpha(x),$$

1177 Therefore from (2.12) π_λ and $\tilde{\pi}_\lambda$ are exactly the same.1178 Similar to (3.7), from (A.15) we have that for any $t \geq 0$ and $x \in \mathbb{R}^d$,

1179 (A.17)
$$\|\delta_x P_t - \pi_\lambda\|_V \leq \tilde{A}_1 \tilde{\rho}_1^t \left(V^2(x) + \int_{\mathbb{R}^d} V^2(\tilde{x}) d\pi_\lambda(\tilde{x}) \right).$$

1180 Since we already proved that $\int_{\mathbb{R}^d} V^2(\tilde{x}) d\pi_\lambda(\tilde{x}) < +\infty$ in Lemma 3.1, we can find $B_1 \geq 0$ such
1181 that for any $x \in \mathbb{R}^d$ we have

1182 (A.18)
$$\|\delta_x P_t - \pi_\lambda\|_V \leq B_1 \tilde{\rho}_1^t V^2(x).$$

1183 Select a large $m_1 \in \mathbb{N}^*$ such that $m_1 \geq \bar{\delta}^{-1}$. Let's now consider the interval $[0, l]$, $l \in \mathbb{N}^*$.
1184 To compare $\pi_{\delta, \lambda}$ with π_δ , we first construct a continuous time Markov process $X_t^{(1)}$ such

1185 that $X_{j/m_1}^{(1)}$ has the same distribution as the j -th sample X_j by NF-ULA (2.13). Define
 1186 $b_1\left(t, (w_t)_{t \in [0, l]}\right) = \sum_{j=0}^{m_1 l - 1} \mathbf{1}_{[j/m_1, (j+1)/m_1)}(t) b_\lambda(w_{j/m_1})$ and $b_2\left(t, (w_t)_{t \in [0, l]}\right) = b_\lambda(w_t)$. Let
 1187 $\mathbf{X}_t^{(1)}$ and $\mathbf{X}_t^{(2)}$ be the unique strong solution of SDE $d\mathbf{X}_t = b\left(t, (\mathbf{X}_t)_{t \in [0, l]}\right) dt + \sqrt{2} d\mathbf{B}_t$
 1188 with $\mathbf{X}_0 = x \in \mathbb{R}^d$ and $b = b_1$, respectively $b = b_2$. Note that $\left(\mathbf{X}_{k/m_1}^{(1)}\right) = (X_k)_{k \in \mathbb{N}}$ and
 1189 $\left(\mathbf{X}_t^{(2)}\right)_{t \geq 0} = (\bar{\mathbf{X}}_t)_{t \geq 0}$. Denote $P_t^{(1)}$ and $P_t^{(2)}$ the Markov semigroup associated with $\mathbf{X}_t^{(1)}$ and
 1190 $\mathbf{X}_t^{(2)}$. Then for any $x \in \mathbb{R}^d$, $k \in \mathbb{N}^*$ we have

$$1191 \quad (\text{A.19}) \quad \delta_x \mathbf{R}_{1/m_1}^{k m_1} = \delta_x \mathbf{P}_k^{(1)}, \quad \delta_x \mathbf{P}_k = \delta_x \mathbf{P}_k^{(2)}.$$

1192 From Lemma 3.7 and A 3.8, for any $t \in [j/m_1, (j+1)/m_1)$, $j \in \{0, \dots, m_1 l - 1\}$ and
 1193 $(w_t)_{t \in [0, l]} \in C([0, l], \mathbb{R}^d)$ we have that

$$1194 \quad (\text{A.20}) \quad \left\| b_1\left(t, (w_t)_{t \in [0, l]}\right) - b_2\left(t, (w_t)_{t \in [0, l]}\right) \right\|^2 = \|b_\lambda(w_{j/m_1}) - b_\lambda(w_t)\|^2 \\ \leq (\mathbf{L}_y + \alpha \mathbf{L} + 1/\lambda)^2 \|w_{j/m_1} - w_t\|^2.$$

1195 Using Cauchy-Schwarz inequality, Hölder's inequality and Itô's isometry we have for any
 1196 $t \in [j/m_1, (j+1)/m_1)$,

$$1197 \quad (\text{A.21}) \quad \mathbb{E} \left[\left\| \mathbf{X}_t^{(2)} - \mathbf{X}_{j/m_1}^{(2)} \right\|^2 \right] = \mathbb{E} \left[\left\| \int_{j/m_1}^t \left(b_\lambda\left(\mathbf{X}_\tau^{(2)}\right) d\tau + \sqrt{2} d\mathbf{B}_\tau \right) \right\|^2 \right] \\ \leq \mathbb{E} \left[2 \left\| \int_{j/m_1}^t b_\lambda\left(\mathbf{X}_\tau^{(2)}\right) d\tau \right\|^2 + 2 \left\| \sqrt{2} (\mathbf{B}_t - \mathbf{B}_{j/m_1}) \right\|^2 \right] \\ \leq 2 \left(t - \frac{j}{m_1} \right) \mathbb{E} \left[\int_{j/m_1}^t \left\| b_\lambda\left(\mathbf{X}_\tau^{(2)}\right) \right\|^2 d\tau \right] + 4d \left(t - \frac{j}{m_1} \right) \\ \leq 2 \left(t - \frac{j}{m_1} \right)^2 \sup_{\tau \leq (j+1)/m_1} \mathbb{E} \left\| b_\lambda\left(\bar{\mathbf{X}}_\tau\right) \right\|^2 + 4d \left(t - \frac{j}{m_1} \right).$$

1198 Since we have proved (A.8), (A.9), (A.10) in Appendix A.6, from Lemma 2.11 and Lemma
 1199 2.12 in [65], for any $\tau > 0$ we have

$$1200 \quad (\text{A.22}) \quad \mathbb{E} \left\| \bar{\mathbf{X}}_\tau \right\|^2 \leq B_{0,0},$$

1201 where $B_{0,0}$ is an upper bound formed by $\lambda, C, b_\lambda(0), d, x$. Then from (A.10) we have that

$$1202 \quad (\text{A.23}) \quad \mathbb{E} \left\| b_\lambda\left(\bar{\mathbf{X}}_\tau\right) \right\|^2 \leq 2 (\mathbf{L}_y + \alpha \mathbf{L} + 1/\lambda)^2 \mathbb{E} \left\| \bar{\mathbf{X}}_\tau \right\|^2 + 2 \|b_\lambda(0)\|^2 \leq B_3, \quad \forall \tau > 0,$$

1203 where $B_3 = 2 (\mathbf{L}_y + \alpha \mathbf{L} + 1/\lambda)^2 B_{0,0} + 2 \|b_\lambda(0)\|^2 \geq 0$.

1204 Then from (A.20), (A.21), (A.23), for $i \in \{0, \dots, l-1\}$ we have that

$$\begin{aligned}
& \int_i^{i+1} \mathbb{E} \left[\left\| b_1 \left(t, \mathbf{X}_t^{(2)} \right) - b_2 \left(t, \mathbf{X}_t^{(2)} \right) \right\|^2 \right] dt \\
& \leq \sum_{j=im_1}^{(i+1)m_1-1} \int_{j/m_1}^{(j+1)/m_1} \mathbb{E} \left[\left\| b_1 \left(t, \mathbf{X}_t^{(2)} \right) - b_2 \left(t, \mathbf{X}_t^{(2)} \right) \right\|^2 \right] dt \\
1205 \quad (A.24) \quad & \leq (L_y + \alpha L + 1/\lambda)^2 \sum_{j=im_1}^{(i+1)m_1-1} \int_{j/m_1}^{(j+1)/m_1} \mathbb{E} \left[\left\| \mathbf{X}_t^{(2)} - \mathbf{X}_{j/m_1}^{(2)} \right\|^2 \right] dt \\
& \leq (L_y + \alpha L + 1/\lambda)^2 \left(\frac{2B_3}{3m_1^2} + \frac{2d}{m_1} \right).
\end{aligned}$$

1206

1207 From (A.19) and Lemma SM6.1 in [57], we obtain that there exists $B_b \geq 0$ such that for
1208 any $x \in \mathbb{R}^d$,

$$\begin{aligned}
1209 \quad (A.25) \quad & \left\| \delta_x R_{1/m_1}^{lm_1} - \delta_x P_l \right\|_V = \left\| \delta_x P_l^{(1)} - \delta_x P_l^{(2)} \right\|_V = \left\| \delta_x P_l^{(2)} - \delta_x P_l^{(1)} \right\|_V \\
& \leq \left(\delta_x P_l^{(1)} [V^2] + \delta_x P_l^{(2)} [V^2] \right)^{1/2} \times \left(\sum_{i=0}^{l-1} \int_i^{i+1} \mathbb{E} \left[\left\| b_1 \left(t, \mathbf{X}_t^{(2)} \right) - b_2 \left(t, \mathbf{X}_t^{(2)} \right) \right\|^2 \right] dt \right)^{1/2} \\
& \leq (L_y + \alpha L + 1/\lambda) \sqrt{l \left(\frac{2B_3}{3m_1^2} + \frac{2d}{m_1} \right)} \left(\delta_x P_l^{(1)} [V^2] + \delta_x P_l^{(2)} [V^2] \right)^{1/2}.
\end{aligned}$$

1210 Assume that there is a function $W \in C^2(\mathbb{R}^d, [1, +\infty))$ such that $\lim_{\|x\| \rightarrow +\infty} W(x) = +\infty$.
1211 Recall that from (A.9), using Lemma SM5.1 in [57] and we get that there exist $\lambda_W \in (0, 1]$,
1212 $c, \beta \geq 0$ and $\zeta > 0$ such that for any $\delta \in (0, (4\lambda)^{-1} (L_y + \alpha L + 1/\lambda)^{-2}]$, R_δ satisfies the
1213 discrete drift condition $\mathbf{D}_d(W, \lambda_W^\delta, c\delta)$ and $(P_t)_{t \geq 0}$ satisfies the continuous drift condition
1214 $\mathbf{D}_c(W, \zeta, \beta)$. From Lemma SM5.2 in [57], there exists $B_c \geq 0$ such that for any $x \in \mathbb{R}^d, t \geq 0$
1215 and $k \in \mathbb{N}^*$ we have

$$1216 \quad (A.26) \quad R_\delta^k W(x) + P_t W(x) \leq B_c^2 W(x).$$

1217 Let $W(x) = V^2(x)$ and $k = m_1 l, \delta = 1/m_1, t = l$, then $\forall x \in \mathbb{R}^d$,

$$1218 \quad (A.27) \quad \delta_x P_l^{(1)} [V^2] + \delta_x P_l^{(2)} [V^2] \leq B_c^2 V^2(x).$$

1219 Combined with (A.25), we have that

$$1220 \quad (A.28) \quad \left\| \delta_x R_{1/m_1}^{m_1 l} - \delta_x P_l \right\|_V \leq B_c V(x) (L_y + \alpha L + 1/\lambda) \sqrt{l \left(\frac{2B_3}{3m_1^2} + \frac{2d}{m_1} \right)}.$$

1221 To give a bound on $\left\| \delta_x R_{1/m_1}^{m_1 l} - \pi_\lambda \right\|_V$, we use triangular inequality to split it into two
1222 terms:

$$1223 \quad (A.29) \quad \left\| \delta_x R_{1/m_1}^{m_1 l} - \pi_\lambda \right\|_V \leq \left\| \delta_x R_{1/m_1}^{m_1 l} - \delta_x P_l \right\|_V + \left\| \delta_x P_l - \pi_\lambda \right\|_V.$$

1224 Using this result and (A.18), we obtain that there exists $B_1, B_2 \geq 0$ such that for any $m_1 \in \mathbb{N}^*$
 1225 with $1/m_1 \leq \bar{\delta}$,

$$1226 \quad (\text{A.30}) \quad \left\| \delta_x R_{1/m_1}^{m_1 l} - \pi_\lambda \right\|_V \leq B_1 \tilde{\rho}_1^l V^2(x) + B_2 V(x) \sqrt{l \left(\frac{B_3}{3m_1^2} + \frac{d}{m_1} \right)}.$$

1227 The proof in the general case where $\delta \in (0, \bar{\delta}]$ is similar when the interval $[0, l]$ is changed to
 1228 $[0, lm_1 \delta]$.

1229 Then we obtain that there exists $B_1, B_2, B_3 \geq 0$, $\tilde{\rho}_1 \in [0, 1)$ such that for any $\delta \in (0, \bar{\delta}]$,
 1230 $k \in \mathbb{N}^*$,

$$1231 \quad (\text{A.31}) \quad \left\| \delta_x R_\delta^k - \pi_\lambda \right\|_V \leq B_1 \tilde{\rho}_1^{k\delta} V^2(x) + B_2 V(x) \sqrt{\delta^2 k \left(d + \frac{B_3 \delta}{3} \right)}. \quad \blacksquare$$

1232 **Appendix B. Additional experiments.**

1233 The second limited-angle computed tomography reconstruction experiment we test is using
 1234 the Poisson noise, where the model can be formulated as $y \sim P(Ax)$ and $P(Ax)$ denotes adding
 1235 a Poisson noise on Ax . We simulate the noisy sinogram as

$$1236 \quad y = -\frac{1}{\mu} \log \left(\frac{N_1}{N_0} \right), \quad N_1 \sim \text{Poisson} (N_0 \exp(-A(x)\mu)).$$

1237 Here $N_0 = 4096$ is the mean photon count per detector bin without attenuation. $\mu = 0.05$ is
 1238 a constant. Since Poisson noise implies a different likelihood

$$1239 \quad p(y|x) = \frac{1}{K_0} \exp(-J(x, y)),$$

$$J(x, y) = \sum_{i=1}^m e^{-A(x)_i \mu} N_0 + e^{-y_i \mu} N_0 (A(x)_i \mu - \log(N_0)),$$

1240 we calculate $\nabla \log p(y|x) = -\nabla J(x, y)$ by using the auto-gradient library.

1241 We select a different $\alpha = 4000$ for NF-ULA while keeping all the other settings the same
 1242 as in the main paper.

1243 Both PnP-ULA and NF-ULA have burn-in iterations of more than 20000. After the
 1244 burn-in time, we calculate the posterior mean and the standard deviation by obtaining 10000
 1245 samples and computing the PSNR of the samples' mean. For Poisson noise, the likelihood is
 1246 more complicated than Gaussian, and NF-ULA spends 510s.

1247 Fig 8 includes the original image, the FBP, the posterior mean and the standard deviation
 1248 of PnP-ULA (realSN-DnCNN) and NF-ULA (patchNR). Table 5 provides the PSNR of the
 1249 posterior mean. All the samples generated in Table 5 never escape $[-0.2, 1.2]^d$, indicating
 1250 that the projection $\Pi_C(x)$ is never activated. Note that the huge uncertainties of standard
 1251 deviation on the left area in the Gaussian-noise case in the main paper are slightly alleviated
 1252 in the Poisson noise experiments. The ACF test results are similar to the CT experiment with
 1253 Gaussian noise, therefore here we do not repeat them again.

Figure 8. Limited-view CT reconstruction with Poisson noise. Column 1: Original image. Column 2: Filtered back projection (FBP). Columns 3, and 4: Posterior mean and the standard deviation of the samples generated by PnP-ULA (realSN-DnCNN). Columns 5, and 6: Posterior mean and the standard deviation of the samples generated by NF-ULA (patchNR). PSNR values of the sample mean images are provided in Table 5.

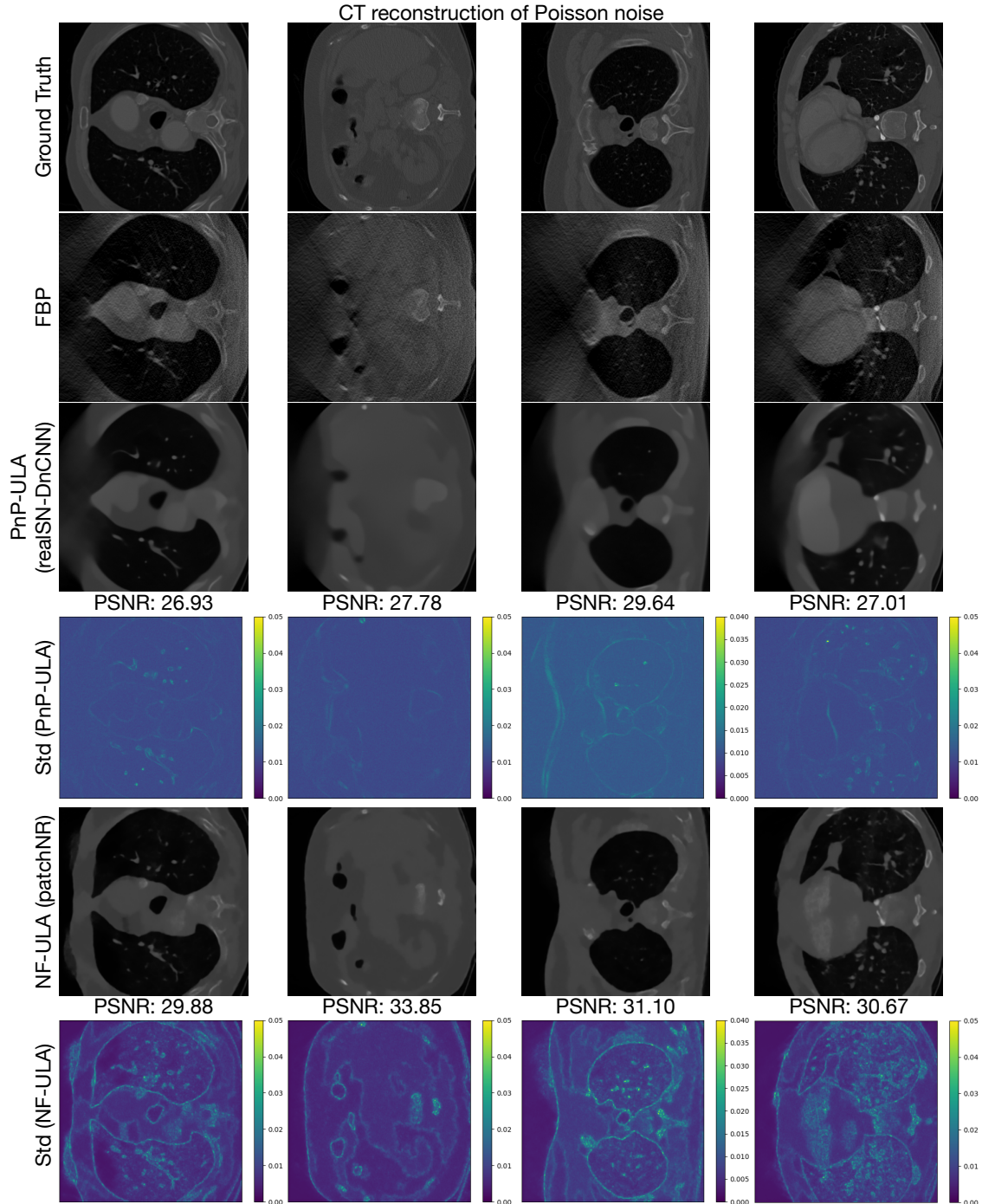


Table 5*CT reconstruction of Poisson noise, limited angles.*

| CT | $C = [-100, 100]^d$ | | |
|---------|---------------------|-----------------|-------|
| | network | parameters | PSNR |
| figure1 | | | |
| NF-ULA | PatchNR | $\alpha = 4000$ | 29.88 |
| PnP-ULA | realSN-DnCNN | $\alpha = 3$ | 26.93 |
| figure2 | | | |
| NF-ULA | PatchNR | $\alpha = 4000$ | 33.85 |
| PnP-ULA | realSN-DnCNN | $\alpha = 3$ | 27.78 |
| figure3 | | | |
| NF-ULA | PatchNR | $\alpha = 4000$ | 31.10 |
| PnP-ULA | realSN-DnCNN | $\alpha = 3$ | 29.64 |
| figure4 | | | |
| NF-ULA | PatchNR | $\alpha = 4000$ | 30.67 |
| PnP-ULA | realSN-DnCNN | $\alpha = 3$ | 27.01 |