Nonlinear MCMC for Bayesian Machine Learning

James Vuckovic
james@jamesvuckovic.com

Abstract

We explore the application of a nonlinear MCMC technique first introduced in [1] to problems in Bayesian machine learning. We provide a convergence guarantee in total variation that uses novel results for long-time convergence and large-particle ("propagation of chaos") convergence. We apply this nonlinear MCMC technique to sampling problems including a Bayesian neural network on CIFAR10.

1 Introduction

Characterizing uncertainty is a fundamental problem in machine learning. It is often desirable for a machine learning model to provide a prediction *and* a measure of how "certain" the model is about that prediction. Having access to a robust measure of uncertainty becomes particularly important in real-world, high risk scenarios such as self-driving cars [2–4], medical diagnosis [5, 6], and classifying harmful text [7].

However, despite the need for uncertainty in machine learning predictions, it is well known that traditional ML training, i.e. based on optimizing an objective function, frequently does not provide robust uncertainty measures [8], yielding overconfident predictions for popular neural networks such as ResNets [9]. [1] An appealing alternative to the traditional optimization paradigm for ML is the Bayesian probabilistic framework, due to its relatively simple formulation and extensive theoretical grounding; see for example [10].

From the probabilistic perspective of machine learning [10], one combines a prior $P(\theta)$ over the parameter space $\theta \in \Theta$ and a likelihood of the data given model parameters $P(\mathcal{D}|\theta)$ using Bayes' rule to obtain a posterior over the parameters $P(\theta|\mathcal{D}) \propto P(\mathcal{D}|\theta)P(\theta)$. The "traditional" approach in machine learning is to *optimize* the posterior (or the likelihood) to obtain $\theta^* \in \arg\max P(\theta|\mathcal{D})$ and generate predictions via $P(y|x, \mathcal{D}) = P(y|x, \theta^*)$. However, if we adopt the Bayesian approach, the posterior characterizes the uncertainty about the parameters of the model (i.e. epistemic uncertainty), which can propagate to uncertainty about a prediction by *integration*: $P(y|x, \mathcal{D}) = \int P(y|x, \theta)P(\theta|\mathcal{D})\mathrm{d}\theta$. This paper studies the problem of how to approximate this integration with samples from $P(\theta|\mathcal{D})$.

1.1 Contributions

- Our main contribution is the novel analysis of a modification of the general nonlinear Markov Chain Monte Carlo (MCMC) sampling method from [1] to obtain quantitative convergence guarantees in both the number of iterations and the number of samples.
- We apply the general results from above to determine the convergence of two specific nonlinear MCMC samplers.
- In experiments, we compare these nonlinear MCMC samplers to their linear counterparts, and find that nonlinear MCMC provides additional flexibility in designing sampling algorithms with as good, or better, performance as the linear variety.

[1] In Appendix C.2.4, we provide an experiment that demonstrates this effect.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

1.2 Background

Bayesian ML & MCMC. In Bayesian machine learning, the "computationally difficult" step is integration since the integral $\int P(y|x,\theta)P(\mathrm{d}\theta|\mathcal{D})$ is not analytically solvable except for rare cases. In practice, one typically uses a Monte Carlo approximation such as

$$\int P(y|x,\theta)P(\theta|\mathcal{D})\mathrm{d}\theta \approx \frac{1}{N}\sum_{i=1}^{N} P(y|x,\theta^i), \quad \text{where} \quad \theta^i \overset{iid}{\sim} P(\theta|\mathcal{D})$$

where the expected error in this approximation is well known to converge to zero like $\mathcal{O}(1/\sqrt{N})$ by the Central Limit Theorem (CLT). There are various approaches to sampling from $P(\theta|\mathcal{D})$, but Markov chain Monte Carlo (MCMC) is perhaps the most widely used. The basic idea of MCMC is to use a Markov transition kernel $\mathcal{T}$ with stationary measure $P(\theta|\mathcal{D})$ to simulate a Markov chain $\theta_{n+1} \sim \mathcal{T}(\theta_n, \bullet)$ that converges rapidly to $P(\theta|\mathcal{D})$. In this case, we can estimate

$$\int P(y|x,\theta)P(\theta|\mathcal{D})\mathrm{d}\theta \approx \frac{1}{N}\sum_{i=1}^{N} P(y|x,\theta^i_\infty) \approx \frac{1}{N}\sum_{i=1}^{N} P(y|x,\theta^i_{n_{\text{sim}}})$$

where n_{sim} is some large number of simulation steps and $\{\theta^i_n\}_{n=0}^{\infty}$ are independent Markov chains governed by $\mathcal{T}$.

The basic problem is then to design an efficient transition kernel $\mathcal{T}$. There is a vast body of literature studying various choices of $\mathcal{T}$; some well known choices are the Metropolis-Hastings algorithm [11, 12], the Gibbs sampler [13], the Langevin algorithm [14–17], Metropolis-Adjusted Langevin [18, 19], and Hamiltonian Monte Carlo [20–24].

However, there are various challenges in Bayesian ML that make applying these samplers difficult in practice. One challenge is that the posterior $P(\theta|\mathcal{D})$ can be highly multimodal [25, 26], which makes it difficult to ensure that the Markov chain θ_n explores all modes of the target distribution. One can combat this issue by employing auxiliary samplers that explore a more "tractable" variation of $P(\theta|\mathcal{D})$ [27, 28]. Other methods that empirically improve posterior sampling quality include tempering [29–32], RMSProp-style preconditioning [33], or adaptive MCMC algorithms [34, 35] such as the popular No U-Turn Sampler [36].

Nonlinear MCMC. Another class of powerful MCMC algorithms, which is less-studied in the context of Bayesian ML, arises from allowing the transition kernel $\mathcal{T}$ to depend on the distribution of the Markov chain as in $\theta_{n+1} \sim \mathcal{T}_{\mathrm{Distribution}(\theta_n)}(\theta_n, \bullet)$. This approach gives rise to so-called *nonlinear* MCMC since $\{\theta_n\}$ is no longer a true Markov chain. Nonlinear Markov theory is a rich area of research [37–42] and has strong connections to nonlinear filtering problems [43, 44], sequential Monte Carlo [45, 46], and nonlinear Feynman-Kac models [47]. One can replace $\mathrm{Distribution}(\theta_n)$, which is often intractable, with an empirical estimate $\mathrm{Distribution}(\theta_n) \approx \frac{1}{N}\sum_{i=1}^{N}\delta_{\theta^i_n}$ to obtain *interacting* particle MCMC (iMCMC, or iPMCMC) methods; see for example [48, 49, 1, 50, 51].

Our view is that nonlinear MCMC offers some appealing features that traditional linear MCMC lacks. One such feature is the ability to leverage *global information* about the state space Θ contained in $\mathrm{Distribution}(\theta^i_n)$ to improve exploration, a central issue in Bayesian ML. Another feature is the increased flexibility of nonlinear MCMC algorithms, which can be leveraged to correct biases that are introduced by other design decisions in MCMC for Bayesian ML such as tempering. These features will be explored empirically in Section 4.

However, the theoretical analysis of nonlinear Markov MCMC presents an added difficulty in that the particles $\{\theta^1_n, \ldots, \theta^N_n\}$ of an interacting particle system are now *statistically dependent*. This means that, in addition to studying the long-time behaviour which is classical in MCMC [52, 53], one must study the large-particle behaviour separately to obtain Monte Carlo estimates since the CLT does not apply. One such large-particle behaviour is the propagation of chaos [41], which is the tendency for groups of interacting particles to become independent as the number of particles, N, increases; see [41]. We will need both of these elements — long-time convergence and propagation of chaos — to properly characterize the convergence of nonlinear MCMC.

Other Sampling Methods. Finally, let us mention that MCMC is certainly not the only way to obtain Monte Carlo sample estimates in Bayesian ML; some popular examples include MC dropout [8], black-box variational inference [54], and normalizing flows [55, 56].

1.3 Common Notation

Let $\mathcal{P}(\mathbb{R}^d)$ be the set of probability measures on the measurable space $(\mathbb{R}^d, \mathscr{B}(\mathbb{R}^d))$. For $\mu \in \mathcal{P}(\mathbb{R}^d)$ and $f \in \mathcal{B}_b(\mathbb{R}^d) := \{f : \mathbb{R}^d \to \mathbb{R} \mid f \text{ is bounded}\}$, we will denote $\mu(f) := \int f \mathrm{d}\mu$. If $K : \mathbb{R}^d \times \mathscr{B}(\mathbb{R}^d) \to [0,1]$ is a Markov kernel[2] then we will denote $Kf(x) := \int f(y)K(x, \mathrm{d}y)$ and $\mu K(\mathrm{d}y) := \int \mu(\mathrm{d}x)K(x, \mathrm{d}y)$. Finally, for $\overline{y} := \{y^1, \ldots, y^N\} \subset \mathbb{R}^d$, we will denote the empirical measure of $\overline{y}$ as $m(\overline{y}) := \frac{1}{N}\sum_{i=1}^N \delta_{y^i} \in \mathcal{P}(\mathbb{R}^d)$.

2 Nonlinear MCMC

In this section, we outline the family of MCMC algorithms that will be studied in rest of the work. We will use general notation for simplicity but this difference is merely cosmetic; the "target distribution" π in this section corresponds directly to $P(\theta|\mathcal{D})$ from the previous section.

2.1 Nonlinear Jump Interaction Markov Kernels

To specify a MCMC algorithm, we must specify the Markov transition kernel. The family of nonlinear Markov kernels that we will be studying was introduced in [1] and is a mixture of a linear kernel, denoted K, and a nonlinear jump-interaction kernel indexed by a probability measure η, denoted J_η, to obtain

$$K_\eta(x, \mathrm{d}y) := (1-\varepsilon)K(x, \mathrm{d}y) + \varepsilon J_\eta(x, \mathrm{d}y) \tag{1}$$

where $\varepsilon \in]0,1[$ is the mixture hyperparameter. The Markov kernel K_η will be the main object of interest throughout this paper. We will give specific examples of J_η in Section 2.2, which were also introduced in [1]. Despite building on the constructions of [1], this work proceeds in some different substantially directions; see Appendix A.1 for more details.

Mean Field System. Now we show how the kernel K_η can be used to construct a Markov chain. Following [1], we use an auxiliary Markov chain $\{Y_n\}$ with transition kernel Q on the same state space as K_η (i.e. $\mathbb{R}^d$) to obtain the nonlinear Markov chain $\{(Y_n, X_n)\}_{n=0}^\infty$ defined by

$$\begin{cases} Y_{n+1} \sim Q(Y_n, \bullet) \\ \eta_{n+1} := \mathrm{Distribution}(Y_{n+1}) \quad Y_0 \sim \eta_0,\ X_0 \sim \mu_0 \\ X_{n+1} \sim K_{\eta_{n+1}}(X_n, \bullet) \end{cases} \tag{2}$$

where $\mu_0, \eta_0 \in \mathcal{P}(\mathbb{R}^d)$ are the initial distributions and $\sim$ denotes "sample from". One should interpret this as a sequence of steps where first we sample the auxiliary state Y_{n+1} from Q, then we obtain the distribution of Y_{n+1} denoted η_{n+1}, and we use this distribution to index the primary kernel $K_{\eta_{n+1}}$ and obtain a sample X_{n+1}. We sample X_{n+1} with probability $(1-\varepsilon)$ from the linear kernel K, and with probability ε it will "jump" according to $J_{\eta_{n+1}}(X_n, \bullet)$. Because the Markov dynamics depend on $\mathrm{Distribution}(Y_n)$, we call this a "mean field system".

Interacting Particle System. One issue with the mean field system (2) is the fact that computing $\mathrm{Distribution}(Y_{n+1})$ is generally impossible except in special cases. Hence, to get a viable simulation algorithm, we must approximate $\mathrm{Distribution}(Y_n)$, and we do this by replacing $\mathrm{Distribution}(Y_n)$ with its empirical measure estimated from a set of N particles $\overline{Y}_n := \{Y_n^1, \ldots, Y_n^N\}$ as follows:

$$\begin{cases} Y_{n+1}^i \sim Q(Y_n^i, \bullet) \\ \eta_{n+1}^N := m(\overline{Y}_{n+1}) \qquad Y_0^i \overset{iid}{\sim} \eta_0,\ X_0^i \overset{iid}{\sim} \mu_0,\ i = 1, \ldots, N. \\ X_{n+1}^i \sim K_{\eta_{n+1}^N}(X_n^i, \bullet) \end{cases} \tag{3}$$

2.2 Application to MCMC

Now we detail how to apply K_η and the Markov chains (2) and (3) to MCMC. In particular, we must understand how to choose Q, K, J_η such that K_η will be invariant w.r.t. a target distribution π.

[2] i.e. $K(x, \bullet)$ is a probability measure $\forall x \in \mathbb{R}^d$ and $K(\bullet, A)$ is measurable $\forall A \in \mathscr{B}(\mathbb{R}^d)$

As is usually the case in probabilistic inference problems, we will assume that the target distribution π is known only up to a normalizing constant and that it has a density, also denoted π. We also make the simplifying assumption that Q has an invariant measure $\eta^\star$ (also with density denoted $\eta^\star$) i.e. $\eta^\star Q = \eta^\star$. This is not burdensome; in practice we can, and will, obtain Q from a *linear* MCMC algorithm for some choice of $\eta^\star$. In fact, being able to choose $\eta^\star$ is a powerful design parameter of our methods as we will see in Section 4. We will also assume that the linear kernel K is π-invariant, i.e. $\pi K = \pi$.

To see how we can ensure that π is K_η-invariant, consider the fact that we will design Q s.t. $\eta_n := \text{Distribution}(Y_n)$ converges to $\eta^\star$. This means we will eventually be sampling from the kernel $K_{\eta^\star}$ and we already have π-invariance of K. Therefore, if we arrange for $J_{\eta^\star}$ to be π-invariant, π will be invariant for $K_{\eta^\star}$ since

$$\pi K_{\eta^\star} = (1-\varepsilon)\pi K + \varepsilon \pi J_{\eta^\star} = (1-\varepsilon)\pi + \varepsilon\pi = \pi.$$

Intuitively, if the auxiliary chain converges to a steady state and the jumps in that steady state preserve π (and K preserves π), then so will $K_{\eta^\star}$. Now the remaining task is to design nonlinear interaction kernels J_η that will yield good performance; we detail two choices below.

Boltzmann-Gibbs Interaction. The first choice of J_η we will investigate, from [1], relies on the Boltzmann-Gibbs transformation [47], which we now explain. Let $G : \mathbb{R}^d \to]0, \infty[$ be a potential function; then the Boltzmann-Gibbs (BG) transformation is a nonlinear mapping $\Psi_G : \mathcal{P}(\mathbb{R}^d) \to \mathcal{P}(\mathbb{R}^d)$ defined by

$$\Psi_G(\mu)(\mathrm{d}x) := \frac{G(x)}{\mu(G)}\mu(\mathrm{d}x) \quad \text{or equivalently} \quad \int f(x)\Psi_G(\mu)(\mathrm{d}x) := \int f(x)\frac{G(x)}{\mu(G)}\mu(\mathrm{d}x)$$

for any $f \in \mathcal{B}_b(\mathbb{R}^d)$ and whenever $\mu(G) \neq 0$. This transformation has many interesting properties and been extensively studied in [47] and related works.

To use the BG transformation in MCMC, we will assume that the densities π and $\eta^\star$ are positive[3] and make the choice that $G(x)$ will be the function

$$G(x) := \frac{\pi(x)}{\eta^\star(x)}. \tag{4}$$

With this choice, we get an interaction kernel $J_\eta^{BG}(x, \mathrm{d}y) := \Psi_G(\eta)(\mathrm{d}y)$. We can easily see that

$$\eta^\star(G) = \int \frac{\pi}{\eta^\star}\mathrm{d}\eta^\star = \int \mathrm{d}\pi = 1 \quad \text{and} \quad \Psi_G(\eta^\star)(\mathrm{d}x) = \frac{G(x)}{\eta^\star(G)}\eta^\star(\mathrm{d}x) = \frac{\pi(x)}{\eta^\star(x)}\eta^\star(\mathrm{d}x) = \pi(\mathrm{d}x),$$

i.e. Ψ_G is the multiplicative "change of measure" from $\eta^\star$ to π. Hence the first nonlinear Markov kernel we will investigate is

$$K_\eta^{BG}(x, \mathrm{d}y) := (1-\varepsilon)K(x, \mathrm{d}y) + \varepsilon\Psi_G(\eta)(\mathrm{d}y). \tag{5}$$

From the remarks above, is clear that π is $K_{\eta^\star}$-invariant.

Accept-Reject Interaction. The second choice of jump interaction we will study, also introduced in [1], is a type of accept-reject interaction related to the Metropolis-Hastings algorithm. For the *same* choice of potential function G in (4), we can define the acceptance ratio

$$\alpha(x, y) := 1 \wedge \frac{G(y)}{G(x)} = 1 \wedge \frac{\pi(y)\eta^\star(x)}{\eta^\star(y)\pi(x)} \quad \text{and the quantity} \quad A_\eta(x) := \int \alpha(x, y)\eta(\mathrm{d}y)$$

for $\eta \in \mathcal{P}(\mathbb{R}^d)$. Hence we can define the accept-reject interaction kernel as[4]

$$J_\eta^{AR}(x, \mathrm{d}y) := \alpha(x, y)\eta(\mathrm{d}y) + (1 - A_\eta(x))\delta_x(\mathrm{d}y).$$

We can interpret this jump interaction as: starting in state x, we jump to a new state distributed according to $\eta(\mathrm{d}y)$ with probability $\alpha(x, y)$ (i.e. accept the proposed jump) and remain the in current state with probability $1 - A_\eta(x)$ (i.e. reject the proposed jump). This is a form of "adaptive Metropolis-Hastings" in which the proposal distribution evolves over time as the distribution of the auxiliary Markov chain. Hence we obtain the accept-reject nonlinear jump interaction kernel

$$K_\eta^{AR}(x, \mathrm{d}y) := (1-\varepsilon)K(x, \mathrm{d}y) + \varepsilon[\alpha(x, y)\eta(\mathrm{d}y) + (1 - A_\eta(x))\delta_x(\mathrm{d}y)]. \tag{6}$$

We note that π is also $J_{\eta^\star}^{AR}$-invariant; see Proposition 3 in Appendix G for a simple calculation.

[3] this can be relaxed to $\pi \ll \mu$ and $\mu \ll \pi$

[4] Given $f \in \mathcal{B}_b(\mathbb{R}^d)$, we can also write this as $J_\eta^{AR} f(x) = \int [f(y) - f(x)]\alpha(x, y)\eta(\mathrm{d}y) + f(x)$

Simulation. Let us note briefly that using both K_η^{BG} and K_η^{AR} in (3) produce interacting particle systems that can, and will, be simulated. The simulation is relatively straightforward, see Appendix A for pseudocode implementing the nonlinear MCMC algorithms we have now constructed.

3 Convergence Analysis

We will now study whether the nonlinear MCMC algorithms based on K_η from Section 2 — i.e., the interacting particle system (3) with the restrictions on K, Q, J_η from Section 2.2 — will actually converge to the target distribution π. In other words, we would like to estimate $\|\mu_n^N - \pi\|$ for some suitable notion of distance on $\mathcal{P}(\mathbb{R}^d)$, where $\mu_n^N := \text{Distribution}(X_n^1)$ is the distribution of a single particle (it doesn't matter which particle as the X_n^i are *exchangeable*).

The nonlinear nature of K_η makes this analysis more difficult than of a linear MCMC method. We break the problem into two parts: one studying the convergence of the mean-field system (2) as the number of steps $n \to \infty$, and one studying the convergence of the interacting particle system (3) to the mean field system as the number of particles $N \to \infty$. This will allow us to apply the triangle inequality as follows:

$$\|\mu_n^N - \pi\| \leq \underbrace{\|\mu_n^N - \mu_n\|}_{\text{large-particle convergence}} + \underbrace{\|\mu_n - \pi\|}_{\text{long-time convergence}}$$

where $\mu_n := \text{Distribution}(X_n)$ is the distribution of the mean-field system. Crucially, our analysis of the large-particle limit is *uniform* in the number of steps n, which will allow us to establish bounds above that hold as $n \to \infty$. The actual result is contained in Theorem 1.

While our analysis does not rely on heavy mathematical machinery, to state the full set of conditions and results for long-time and large-particle convergence — each of which is a substantial result in its own right — would occupy too much space in the main text. Instead, we will state the main result in Theorem 1, which is essentially a corollary of the long-time and large-particle analyses Theorems 2 and 3 in Appendices E and F respectively, and below we will sketch the general arguments used in those appendices. The proofs of the main results are in Appendix F.2. Note that our analysis, and the results we obtain, are novel and not found in [1]; see Appendix D for an elaboration.

The following result is stated in terms of the *total variation* metric, defined here for $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$ as $\|\mu - \nu\|_{tv} := \sup_{\|f\|_\infty \leq 1} |\mu(f) - \nu(f)|$ where $\| \bullet \|_\infty$ is the sup-norm on $\mathcal{B}_b(\mathbb{R}^d)$.

Theorem 1. *[Convergence of Nonlinear MCMC] Under suitable conditions on K_η and Q, there exist fixed constants $C_1, C_2, C_3 > 0$, a function $\mathcal{R} : [0, \infty[\to [1, \infty[$, and $\rho > 0$ s.t.*

$$\|\mu_n^N - \pi\|_{tv} \leq C_1 \frac{1}{N} \mathcal{R}(1/N) + C_2 \rho^n + C_3 n \rho^n.$$

♦

Let us make a couple of remarks:

- This result shows that, to control the approximation error $\|\mu_n^N - \pi\|_{tv}$, it does not necessarily suffice to run the MCMC algorithm for a large number of steps n, since if $n \to \infty$ but $N < \infty$ then our bound on $\|\mu_n^N - \pi\| \not\to 0$. However, this approximation cannot lead to arbitrarily bad results: Theorem 1 provides a quantitative upper bound on how much the MCMC algorithm can be biased. This behaviour is supported empirically; in Figure 5 of Appendix C.1.4 we provide a clear illustration of how changing N significantly affects the bias of our nonlinear MCMC methods while having no effect on the bias of linear MCMC, as expected.
- This result uses total variation, which is a strong metric that represents a worst-case over *all* bounded functions f (up to rescaling by $\|f\|_\infty$). It is entirely possible that, for many choices of practical f, the approximation will be better as we will see empirically.
- The constant ρ is, roughly speaking, the slower of the rate of convergence for Q and for K. Hence if K, Q are chosen to be efficient samplers with fast convergence, this will result in $\rho \ll 1$ and hence μ_n^N will also converge quickly.

- In our specific samplers K_η^{BG} and K^{AR}, we will see in Appendix G that $\mathcal{R}$ is a monotonically increasing function that is lower-bounded by 1. Hence, as $N \to \infty$, $\frac{1}{N}\mathcal{R}(\frac{1}{N}) \to 0$ as expected.

A corollary of Theorem 1 is that that we regain a Monte Carlo estimate for the interacting particle system. This result is essentially due to [41] Theorem 2.2.

Corollary 1. *[Adapted from [41], Theorem 2.2] Suppose that Theorem 1 applies to K_η. Let $\overline{X}_n := \{X_n^1, \dots, X_n^N\}$ be the interacting particle system from (3). Then for every $n \in \mathbb{N}$ and $f \in \mathcal{B}_b(\mathbb{R}^d)$ we have*

$$\lim_{N\to\infty} \mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^{N} f(X_n^i) - \mu_n(f)\right|\right] = 0.$$

♦

This corollary directly relates to the application of Bayesian ML we are interested in, where we would have $f(\theta) = P(y|x,\theta)$.

3.1 Long-Time Bounds

There are two main ingredients in the general result on long-time convergence: ergodicity of K and Q, and Lipschitz regularity of the interaction kernel $\eta \mapsto J_\eta$. These, along with other technical conditions, produce an estimate of the form $\|\mu_n - \pi\| \leq C_2\rho^n + C_3 n\rho^n$ where $\| \bullet \|$ is a weighted total variation norm. The full statement is in Theorem 2 of Appendix E.

Ergodicity of K and Q. A fundamental requirement of our results is that the *linear* building blocks of K_η must converge to their respective stationary measures in an appropriate metric. This type of result is now standard in the Markov chain literature, and we use a result from [57] for K and a result from [1] for Q. The former is actually able to ensure that K is a contraction on $\mathcal{P}(\mathbb{R}^d)$ w.r.t. a suitable weighted total variation; we use this feature repeatedly in our analysis.

Lipschitz Regularity of J_η. We also need that $\eta \mapsto J_\eta$ is Lipschitz-continuous w.r.t. a weighted total variation norm on Markov kernels (the Lipschitz constant does not have to be < 1). This regularity is used to translate the convergence of $\eta_n \to \eta^\star$ (as guaranteed by the ergodicity of Q) into convergence of $J_\eta \to J_{\eta^\star}$ with the Lipschitz estimate $\|J_{\eta_n} - J_{\eta^\star}\| \lesssim \|\eta_n - \eta^\star\|$. In Appendix G, we verify this analytically for J_η^{BG} and J_η^{AR}, see Lemma 5 and [1] Proposition 5.3.

3.2 Large-Particle Bounds

To study the large-particle behaviour, we would like to measure how close subset of $q \in \{1, \dots, N\}$ interacting particles $\{X_n^1, \dots, X_n^q\} \subset \{X_n^1, \dots, X_n^N\} =: \overline{X}_n$ from (3) is to being i.i.d. according to the mean-field measure μ_n. This analysis was pioneered in [41] under the name "propagation of chaos" and formalizes the intuition that, as $N \to \infty$, the influence of any individual particle $\to 0$.

To state this more precisely, first note that if we had random variables $Z^i \overset{iid}{\sim} \eta \in \mathcal{P}(\mathbb{R}^d)$ then the joint distribution of $\overline{Z} := \{Z^1, \dots, Z^N\}$ would be $\eta^{\otimes N}$. Hence, as $N \to \infty$ for the interacting particles $\overline{X}_n$ at time n, we expect the distribution of $\{X_n^1, \dots, X_n^q\}$, denoted $\mu_n^{q,N}$, to get closer to the distribution of i.i.d. mean field particles from (2), denoted $\mu_n^{\otimes q}$. In other words, we expect $\|\mu_n^{q,N} - \mu_n^{\otimes q}\|_{tv} \to 0$ as $N \to \infty$. The full statement of this result is Theorem 3 of Appendix F.

The main condition in our propagation of chaos result is another type of regularity for $\eta \mapsto J_\eta$ which basically requires that, if one approximates a distribution $\eta \in \mathcal{P}(\mathbb{R}^d)$ by its empirical measure $m(\overline{Y})$ where $\overline{Y} := \{Y^1, \dots, Y^N\}$ and $Y^i \overset{iid}{\sim} \eta$, then $J_{m(\overline{Y})} \to J_\eta$ as $N \to \infty$. More precisely, there should be a function $\mathcal{R} : [0, \infty[\to [1, \infty[$, which is ideally nondecreasing, s.t.

$$|\mathbb{E}[J_{m(\overline{Y})}^{\otimes q} f(x)] - J_\eta^{\otimes q} f(x)| \lesssim \frac{q^2}{N}\mathcal{R}(q^2/N) \quad \forall x \in \mathbb{R}^d \text{ and } f \in \mathcal{B}_b(\mathbb{R}^d) \text{ with "oscillations" } \mathrm{osc}(f) \leq 1.$$

The expectation is taken over $\eta^{\otimes N}$, and "oscillations" are defined precisely in Appendix D. This inequality is essentially a total variation regularity since we can alternately write $\|\mu - \nu\|_{tv} = \sup\{|\mu(f) - \nu(f)| \mid f \in \mathcal{B}_b(\mathbb{R}^d),\ \mathrm{osc}(f) \leq 1\}$ [47].

3.3 Analysis of Specific Interaction Kernels

The main result Theorem 1 is in terms of conditions on a general K_η (i.e. general choices of K, Q, J_η). To apply this result to the samplers in Section 2, we must establish if these conditions hold for K_η^{BG} and K_η^{AR}. Fortunately this can be done analytically; in Appendix G, we present conditions under which the Lipschitz regularity (see Lemma 5 and [1] Proposition 5.3) and large-particle regularity (see Corollaries 3 and 5) hold. These results, particularly for K_η^{BG}, are interesting and rely on novel techniques for controlling the various quantities, sometimes improving over previous methods. Due to space constraints, the results for K_η^{BG} and K_η^{AR} are in Corollaries 2 and 4 from Appendix G.

4 Experiments

In this section, we detail two experiments designed to explore how one might apply the nonlinear MCMC methods developed in the previous sections to Bayesian machine learning.[5] Let us state explicitly that our aim is *not* to achieve state-of-the-art with these experiments, nor do we claim that this method will necessarily lead to state-of-the-art results on a particular task. Rather, the aims of these experiments are: to demonstrate that nonlinear MCMC can be applied successfully to large-scale problems; to compare linear vs nonlinear methods to understand what benefits and drawbacks nonlinear MCMC offers compared to linear MCMC in practice; and to develop some recipes for choosing the various hyperparameters and samplers that determine a nonlinear MCMC method.

4.1 Two-Dimensional Toy Experiments

First, we use a toy setting of two-dimensional distributions to compare the relative benefits of linear and nonlinear MCMC. A benefit of this simple setting is that the multimodal toy distributions can be exactly sampled. This gives us the opportunity to quantify the quality of our samples via an unbiased estimator of the Maximum Mean Discrepancy (MMD) metric on $\mathcal{P}(\mathbb{R}^d)$ [58]. This approach stands in contrast to many previous works, which use simplistic distributions (e.g. Gaussians) with analytically tractable statistics to measure quality. See Appendix C.1 for an overview of our methodology.

Setup. Our setup will compare the Metropolis Adjusted Langevin Algorithm (MALA) [18] (see Appendix B for an overview of MALA) with the nonlinear BG and AR samplers using MALA for the kernels K, Q in our nonlinear setup from Section 2. This will allow us to examine the effects of the nonlinearity in K_η^{AR} and K_η^{BG} while controlling for the type of sampler used and its hyperparameters.

The main difference between the linear and nonlinear algorithms, aside from the interaction itself, is the extra "design knob" to control in the form of the choice of the auxiliary density $\eta^\star$. Below, we show how one can use $\eta^\star$ to incorporate additional insight to guide the sampling, such as regions of the state space to explore. In our experiments, we choose $\eta^\star$ to be a centered, 2-dimensional Gaussian with a large variance ($\Sigma = 4I_2$ for the circular MoG and two-rings densities, and $\Sigma = 20I_2$ for the grid MoG). This conveys "coarse-grained" information of roughly where the support of the target density is located – in this case, a neighbourhood of $(0, 0)$. See Table 2 in Appendix C.1 for a full account of our experimental settings.

Results. From Figure 1, we see that having a simple auxiliary density with good coverage of the support of the target distribution is quite helpful. In all three examples, one or both of the nonlinear samplers outperformed the equivalent linear sampler in the empirical MMD metric. For the two most challenging densities, the "two rings" and "grid MoG" distributions, the improved exploration is particularly evident. We also include an analysis of the runtime of the algorithms in Appendix C.1.3.

Comparison With [1]. We also compared the performance of our methods with those of [1] in this two-dimensional toy setting. See Appendix C.1.4 for an overview of the results; they support all of the theoretical considerations and design principles we have introduced in this paper.

[5] The code used in our experiments can be found at `https://github.com/jamesvuc/nonlinear-mcmc-paper`. See also Appendix A for a discussion of the implementation details.

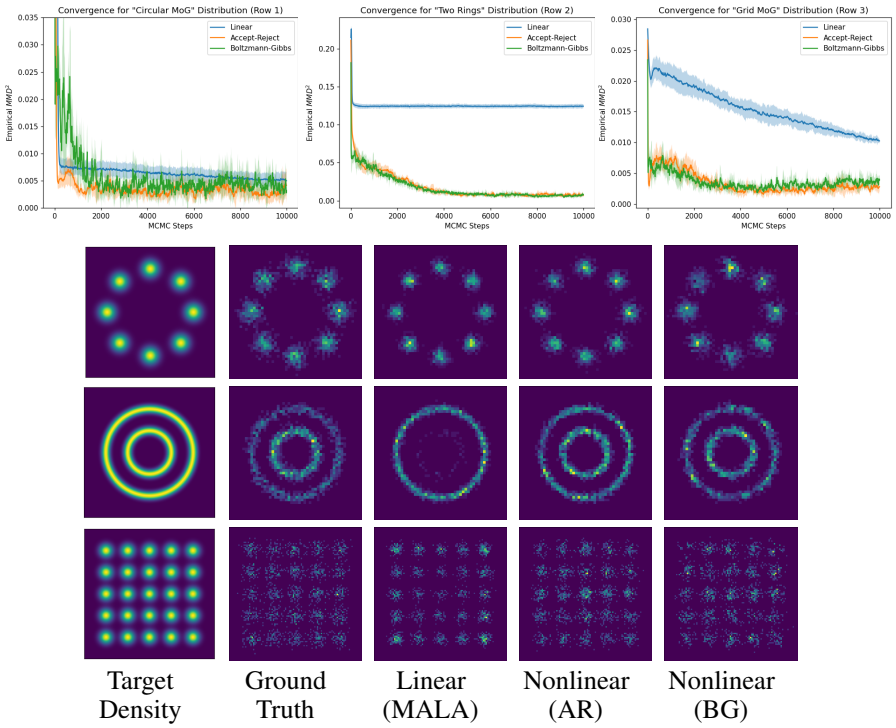


Figure 1: Visualizations of the 2d experiment. The top row shows the empirical MMD-squared plotted over number of sampled steps, where the shaded region is ± 1 standard deviation with 5 independent runs. The bottom three rows show histograms for the $N = 2000$ samples of the Circular Mixture of Gaussians (MoG) density [59], the Two Rings density [59], and the Grid Mixture of Gaussians density [60] respectively.

4.2 CIFAR10

4.2.1 Setup

To examine the properties of the nonlinear sampler outside of a toy setting, we have also implemented a Bayesian neural network on the CIFAR10 dataset. We use a likelihood $P(y|x, \theta)$ parameterized by a ResNet-18 convolutional neural network [61] and a Gaussian prior $P(\theta)$ on the parameters of this neural network which are combined to form a posterior $P(\theta|\mathcal{D}_{\text{train}}) \propto P(\theta) \prod_{(x_i,y_i)\in\mathcal{D}_{\text{train}}} P(y_i|x_i, \theta)$. The goal is to sample $\theta^i,\ i = 1, \ldots, N$ from this posterior. See Table 4 in Appendix C.2 for a full account of the experimental settings.

To deal with the fact that this sampling problem is very high-dimensional ($d \approx 11\text{M}$) and $|\mathcal{D}_{\text{train}}|$ is large ($|\mathcal{D}_{\text{train}}| = 60,000$) we use a variety of techniques:

1. We sample minibatches $\widehat{\mathcal{D}}$ of size 256 to obtain the surrogate target density $P(\theta|\widehat{\mathcal{D}})$ [17].
2. We use an RMSProp-like "preconditioned" Langevin algorithm, called RMS-Langevin or RMS-ULA, as in [33] for the auxiliary sampler Q; see Appendix B for details on this sampler. As shown in [33], this sampler is biased.
3. We use tempering, wherein we aim to sample from π or $\eta \propto P(\theta|\mathcal{D}_{\text{train}})^{1/\tau}$ where τ is a small number. This substantially improves mixing for hard-to-sample distributions such as $P(\theta|\mathcal{D}_{\text{train}})$ at the cost of bias since we are no longer sampling from the true posterior [32].

Using our nonlinear algorithm presents a novel opportunity to correct the bias introduced by tempering. For our experiments, we pick $\eta^\star \propto P(\theta|\mathcal{D}_{\text{train}})^{1/\tau}$ and $\pi = P(\theta|\mathcal{D}_{\text{train}})$. This means that the auxiliary chain Y_n explores a tempered version of the target, whereas the target chain X_n (in theory) explores the true target distribution. This is a novel strategy that is made possible by being able to select $\eta^\star$ almost independently of the target π. We study the case when π is tempered as well.

For our experiments, we use the RMS-Langevin sampler as the baseline, and we also use it for the auxiliary sampler Q. For the target sampler, it is not possible to use the RMS-Langevin algorithm because the smoothed square-gradient estimate is incompatible with the discontinuities (i.e. jumps) introduced by the nonlinear interaction. Instead, for the linear sampler K we use the unadjusted Langevin algorithm, ULA, [16] (see Appendix B). We investigate both test accuracy and calibration error [9] to assess performance.

Table 1: Results for CIFAR10 experiments. $\pm$ represents 1 standard deviation on 5 random seeds. The tempered results are using $\tau = 10^{-4}$. See Appendix C.2 for an overview of expected calibration error. We also compute the maximum calibration error in Appendix C.2. All Expected Calibration Error numbers are multiplied by 10^2 in this table.

	Test Accuracy (↑)		**Expected Calibration Error** (↓)	
Algorithm	Non-Tempered	Tempered	Non-Tempered	Tempered
Linear	$85.01_{\pm 0.10}$	$85.01_{\pm 0.19}$	$0.24_{\pm 0.02}$	$0.26_{\pm 0.014}$
Nonlinear (BG)	$84.28_{\pm 0.28}$	$84.74_{\pm 0.08}$	$0.14_{\pm 0.03}$	$0.16_{\pm 0.03}$
Nonlinear (AR)	*Diverged*	$84.67_{\pm 0.23}$	*Diverged*	$0.15_{\pm 0.05}$

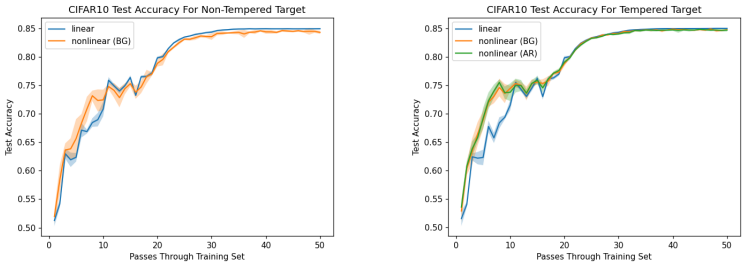


Figure 2: Evaluation of test accuracy during sampling for CIFAR10. The shaded areas represent ± 1 standard deviation for 5 random seeds. For readability, we omit the AR interaction curve on the non-tempered result since it diverged and it distorts the scale of the plot. For completeness, all the curves for the non-tempered case are plotted in Appendix C.2.2, Figure 6.

4.2.2 Results

Linear vs Nonlinear. From Table 1, we see that the linear (RMS-Langevin) sampler has slightly higher, but comparable test accuracy to the nonlinear samplers. This is likely because RMS-Langevin algorithm has better stability around the regions of high probability due to its adaptive stepsize scaling. However, from Figure 2, we see that for both the tempered and non-tempered cases, the nonlinear interaction appears to benefit during early exploration. This is an expected and desired property of these nonlinear samplers, which incorporate global information about the sampler state (in this case, the relative potential G of each auxiliary chain's state) and are able to emphasize those states with higher probability. However, the linear method is able to eventually explore the relevant regions of the state space, and the difference disappears. See Appendix C.2.5 for a comparison linear vs nonlinear performance scaled by the number of gradient evaluations.

Tempering Vs Non-Tempering. The linear MCMC sampler is always tempered in our experiments so there should not be any statistically significant difference in the linear case. For the nonlinear sampler, the tempered version has slightly higher accuracy; this trend is also observed in [32]. On the other hand, the calibration errors are the same for both tempered and non-tempered variants. This is somewhat surprising, given the aggressive tempering used, and one would expect that this reduces the variance of the posterior estimate.[6] This observation can perhaps be explained by the fact that we are using $N = 10$ samples which may not be enough to accurately change the tempered auxiliary distribution $\eta^\star$ into the non-tempered primary distribution π in the jump interaction.

[6] As $\tau \to 0$, this $\theta \sim P(\theta|\mathcal{D}_{\text{train}})^{1/\tau}$ converges to the maximum *a posteriori* estimate with zero variance.

Calibration. Considering the expected calibration error (ECE) [9], in Table 1 we see that the nonlinear method has statistically significantly lower ECE ($p \ll 0.05$) compared to the linear method. We hypothesize that this is due to a tension between the RMS scaling of the gradient which improves the efficiency of each MCMC step but at the cost of bias, which may be measurable in the form of calibration error. The Langevin algorithm is also biased, but is generally known to have good convergence properties [16] and much less is known about the RMS-Langevin variant. By using our nonlinear setup, we are able to aggressively explore the auxiliary distribution without sacrificing calibration on the target distribution.

5 Conclusion

In this paper, we have studied the theoretical and empirical properties of nonlinear MCMC methods. We have obtained powerful theoretical results to characterize the convergence of our MCMC methods, and we have applied these methods to Bayesian neural networks. The results on BNNs are comparable to, but not better than, the linear methods we studied. We hypothesize that this is because more investigation into choosing the best auxiliary density $\eta^\star$ is required; our choice is simplistic and may not be optimal. This hypothesis is supported by our toy experiments, which show significant improvement when $\eta^\star$ is able to incorporate some additional insight into the problem. How to do this in high dimensions is an exciting direction for future research.

Broader Impact. There are benefits and drawbacks to the nonlinear MCMC methods we describe. The benefits are mainly that properly accounting for uncertainty in machine learning will lead to better real-world outcomes for high-value scenarios such as self-driving cars or medical imaging. The drawbacks are that MCMC methods require $\mathcal{O}(N)$ storage and computations relative to the $\mathcal{O}(1)$ for deterministic methods; in fact, our nonlinear method would require $2N$ resources compared with N for a linear method (see also Appendices C.1.3 and C.2.5). If our algorithms were applied to a large swath of ML "as is", this would mean a substantial increase in the energy consumption required for experimentation and deployment, worsening an already substantial issue in the field.

References

[1] Christophe Andrieu, Ajay Jasra, Arnaud Doucet, and Pierre Del Moral. On nonlinear Markov chain Monte Carlo. *Bernoulli*, 17(3):987 – 1014, 2011. doi: 10.3150/10-BEJ307. URL https://doi.org/10.3150/10-BEJ307.

[2] Rhiannon Michelmore, Marta Kwiatkowska, and Yarin Gal. Evaluating uncertainty quantification in end-to-end autonomous driving control. *arXiv preprint arXiv:1811.06817*, 2018.

[3] Sina Shafaei, Stefan Kugele, Mohd Hafeez Osman, and Alois Knoll. Uncertainty in machine learning: A safety perspective on autonomous driving. In *International Conference on Computer Safety, Reliability, and Security*, pages 458–464. Springer, 2018.

[4] Liuhui Ding, Dachuan Li, Bowen Liu, Wenxing Lan, Bing Bai, Qi Hao, Weipeng Cao, and Ke Pei. Capture uncertainties in deep neural networks for safe operation of autonomous driving vehicles. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 826–835. IEEE, 2021.

[5] Benjamin Kompa, Jasper Snoek, and Andrew L Beam. Second opinion needed: communicating uncertainty in medical machine learning. *NPJ Digital Medicine*, 4(1):1–6, 2021.

[6] Roohallah Alizadehsani, Mohamad Roshanzamir, Sadiq Hussain, Abbas Khosravi, Afsaneh Koohestani, Mohammad Hossein Zangooei, Moloud Abdar, Adham Beykikhoshk, Afshin Shoeibi, Assef Zare, et al. Handling of uncertainty in medical data using machine learning and probability theory techniques: A review of 30 years (1991–2020). *Annals of Operations Research*, pages 1–42, 2021.

[7] Jianfeng He, Xuchao Zhang, Shuo Lei, Zhiqian Chen, Fanglan Chen, Abdulaziz Alhamadani, Bei Xiao, and ChangTien Lu. Towards more accurate uncertainty estimation in text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8362–8372, 2020.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[9] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[10] K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning series. MIT Press, 2012. ISBN 9780262018029. URL https://books.google.ca/books?id=NZP6AQAAQBAJ.

[11] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[12] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

[13] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6): 721–741, 1984.

[14] Donald L. Ermak. A computer simulation of charged particles in solution. i. technique and equilibrium properties. *The Journal of Chemical Physics*, 62(10):4189–4196, 1975. doi: 10.1063/1.430300. URL https://doi.org/10.1063/1.430300.

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

[16] Alain Durmus and Eric Moulines. High-dimensional Bayesian inference via the unadjusted Langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019.

[17] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688. Citeseer, 2011.

[18] Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

[19] Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341 – 363, 1996. doi: bj/1178291835. URL https://doi.org/.

[20] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

[21] Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

[22] Radford M. Neal. An improved acceptance procedure for the hybrid Monte Carlo algorithm. *Journal of Computational Physics*, 111:194–203, 1992.

[23] R.M. Neal. *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics. Springer New York, 2012. ISBN 9781461207450. URL https://books.google.ca/books?id=LHHrBwAAQBAJ.

[24] Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73 (2):123–214, 2011.

[25] Peter Müller and David Rios Insua. Issues in Bayesian analysis of neural network models. *Neural Computation*, 10(3):749–770, 1998.

[26] Arya A Pourzanjani, Richard M Jiang, and Linda R Petzold. Improving the identifiability of neural networks for Bayesian inference. In *NIPS Workshop on Bayesian Deep Learning*, volume 4, page 31, 2017.

[27] David M. Higdon. Auxiliary variable methods for Markov chain Monte Carlo with applications. *Journal of the American Statistical Association*, 93(442):585–595, 1998. ISSN 01621459. URL http://www.jstor.org/stable/2670110.

[28] Raza Habib and David Barber. Auxiliary variational MCMC. In *International Conference on Learning Representations*, 2018.

[29] Malcolm Sambridge. A Parallel Tempering algorithm for probabilistic sampling and multimodal optimization. *Geophysical Journal International*, 196(1):357–374, 10 2013. ISSN 0956-540X. doi: 10.1093/gji/ggt342. URL https://doi.org/10.1093/gji/ggt342.

[30] Robert Swendsen and Jian-Sheng Wang. Replica Monte Carlo simulation of spin-glasses. *Physical review letters*, 57:2607–2609, 12 1986. doi: 10.1103/PhysRevLett.57.2607.

[31] Rohitash Chandra, Konark Jain, Ratneel V Deo, and Sally Cripps. Langevin-gradient parallel tempering for Bayesian neural learning. *Neurocomputing*, 359:315–326, 2019.

[32] Florian Wenzel, Kevin Roth, Bastiaan Veeling, Jakub Swiatkowski, Linh Tran, Stephan Mandt, Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How good is the Bayes posterior in deep neural networks really? In *International Conference on Machine Learning*, pages 10248–10259. PMLR, 2020.

[33] Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[34] Christophe Andrieu and Éric Moulines. On the ergodicity properties of some adaptive MCMC algorithms. *The Annals of Applied Probability*, 16(3):1462–1505, 2006.

[35] Christophe Andrieu and Johannes Thoms. A tutorial on adaptive MCMC. *Statistics and computing*, 18(4):343–373, 2008.

[36] Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

[37] Henry P McKean Jr. A class of Markov processes associated with nonlinear parabolic equations. *Proceedings of the National Academy of Sciences of the United States of America*, 56(6):1907, 1966.

[38] Sylvie Méléard. Asymptotic behaviour of some interacting particle systems; mckean-vlasov and boltzmann models. In *Probabilistic models for nonlinear partial differential equations*, pages 42–95. Springer, 1996.

[39] José A Carrillo, Robert J McCann, and Cédric Villani. Kinetic equilibration rates for granular media and related equations: entropy dissipation and mass transportation estimates. *Revista Matematica Iberoamericana*, 19(3):971–1018, 2003.

[40] Arnaud Guillin and Pierre Monmarché. Uniform long-time and propagation of chaos estimates for mean field kinetic particles in non-convex landscapes. *Journal of Statistical Physics*, 185(2): 1–20, 2021.

[41] Alain-Sol Sznitman. Topics in propagation of chaos. In *Ecole d'été de probabilités de Saint-Flour XIX—1989*, pages 165–251. Springer, 1991.

[42] OA Butkovsky. On ergodic properties of nonlinear Markov chains and stochastic McKean–Vlasov equations. *Theory of Probability & Its Applications*, 58(4):661–674, 2014.

[43] Pierre Del Moral. Nonlinear filtering: Interacting particle resolution. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, 325(6):653–658, 1997.

[44] Pierre Del Moral and Laurent Miclo. Asymptotic results for genetic algorithms with applications to nonlinear estimation. In *Theoretical aspects of evolutionary computing*, pages 439–493. Springer, 2001.

[45] Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

[46] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[47] P.D. Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Probability and Its Applications. Springer New York, 2004. ISBN 9780387202686. URL https://books.google.ca/books?id=8LypfuG8ZLYC.

[48] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72 (3):269–342, 2010.

[49] Tom Rainforth, Christian Naesseth, Fredrik Lindsten, Brooks Paige, Jan-Willem Vandemeent, Arnaud Doucet, and Frank Wood. Interacting particle Markov chain Monte Carlo. In *International Conference on Machine Learning*, pages 2616–2625. PMLR, 2016.

[50] Grégoire Clarté, Antoine Diez, and Jean Feydy. Collective proposal distributions for nonlinear MCMC samplers: Mean-field theory and fast implementation. *arXiv preprint arXiv:1909.08988*, 2019.

[51] Michael Arbel, Anna Korba, Adil Salim, and Arthur Gretton. Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems*, 32, 2019.

[52] Gareth O Roberts and Jeffrey S Rosenthal. General state space Markov chains and MCMC algorithms. *Probability surveys*, 1:20–71, 2004.

[53] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[54] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial intelligence and statistics*, pages 814–822. PMLR, 2014.

[55] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[56] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

[57] Martin Hairer and Jonathan C Mattingly. Yet another look at Harris' ergodic theorem for Markov chains. In *Seminar on Stochastic Analysis, Random Fields and Applications VI*, pages 109–117. Springer, 2011.

[58] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[59] Vincent Stimper, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Resampling base distributions of normalizing flows. In *International Conference on Artificial Intelligence and Statistics*, pages 4915–4936. PMLR, 2022.

[60] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2019.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[62] Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999.

[63] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

[64] Grigorios A Pavliotis. *Stochastic processes and applications: diffusion processes, the Fokker-Planck and Langevin equations*, volume 60. Springer, 2014.

[65] Denis Talay and Luciano Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic analysis and applications*, 8(4):483–509, 1990.

[66] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[67] Chii-Ruey Hwang, Shu-Yin Hwang-Ma, and Shuenn-Jyi Sheu. Accelerating diffusions. *The Annals of Applied Probability*, 15(2):1433–1444, 2005.

[68] Andrew B Duncan, Tony Lelievre, and Grigorios A Pavliotis. Variance reduction using nonreversible Langevin samplers. *Journal of statistical physics*, 163(3):457–491, 2016.

[69] Alessandro Barp, Lancelot Da Costa, Guilherme França, Karl Friston, Mark Girolami, Michael I Jordan, and Grigorios A Pavliotis. Geometric methods for sampling, optimisation, inference and adaptive agents. *arXiv preprint arXiv:2203.10592*, 2022.

[70] Joris Bierkens, Paul Fearnhead, and Gareth Roberts. The zig-zag process and super-efficient sampling for Bayesian analysis of big data. *The Annals of Statistics*, 47(3):1288–1320, 2019.

[71] George Deligiannidis, Alexandre Bouchard-Côté, and Arnaud Doucet. Exponential ergodicity of the bouncy particle sampler. *The Annals of Statistics*, 47(3):1268–1287, 2019.

[72] Augustin Chevallier, Sam Power, Andi Q Wang, and Paul Fearnhead. PDMP Monte Carlo methods for piecewise-smooth densities. *arXiv preprint arXiv:2111.05859*, 2021.

[73] Alain Durmus, Arnaud Guillin, and Pierre Monmarché. Piecewise deterministic Markov processes and their invariant measures. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 57, pages 1442–1475. Institut Henri Poincaré, 2021.

[74] Pierre Monmarché. High-dimensional MCMC with a standard splitting scheme for the underdamped Langevin diffusion. *Electronic Journal of Statistics*, 15(2):4117–4166, 2021.

[75] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[76] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[77] Tom Hennigan, Trevor Cai, Tamara Norman, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020. URL http://github.com/deepmind/dm-haiku.

[78] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Appendix D
   (b) Did you include complete proofs of all theoretical results? [Yes] Appendix E, F, G
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code is in the supplementary materials. Pseudocode is in Appendix A.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Appendix C
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In Appendix C
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In Appendix A
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Pseudocode & Numerical Implementation Details

A.1 Simulation Algorithm

The method of simulating K_η is where this work and [1] diverge: we will provide an algorithm that uses a fixed number of samples $N \in \mathbb{N}$ to simulate the IPS (3), whereas [1] investigates an algorithm in which the empirical measure is formed from all the *past samples*.

While the algorithm in [1] yields asymptotically unbiased estimate of the target measure as $n \to \infty$ (for $K_\bullet^{AR}$ at least, see [1]), their MCMC algorithm's memory complexity increases linearly with time. This behaviour is not well-suited to large-scale software implementations, which typically favour fixed-size "batches" of computations. Additionally, a practitioner cannot add more samples to increase the sampling accuracy in a fixed time frame.

By contrast, Algorithm 1 below uses a fixed number of particles N throughout the lifetime of the algorithm. Theorem 1 indicates that this produces a biased estimate of the target measure π, but that this bias can be controlled both in the number of particles, N, and the number of steps, n. In exchange, our algorithm can be efficiently implemented in vectorized computing frameworks, as we demonstrate in Section 4. In Appendix C.1.4, we investigate the differences in empirical performance between the algorithms in [1] and Algorithm 1; those findings unambiguously support this discussion.

Algorithm 1: Sampling from a nonlinear Markov chain with transition K_η.

Input: Initial samples $X_0^i \overset{iid}{\sim} \mu_0,\ Y_0^i \overset{iid}{\sim} \eta_0,\ i = 1, \dots, N$
Input: Primary and auxiliary Markov kernels K, Q resp. and jump kernel J_η
Input: Number of iterations, n_{sim}, jump probability ε
Output: Collection of samples $\{X_{n_{\text{sim}}}^1, \dots, X_{n_{\text{sim}}}^N\}$.

for $n = 0, \dots, n_{sim} - 1$ **do**
  `// Sample auxiliary Markov chain`
  for $i = 1, \dots, N$ **do**
    $Y_{n+1}^i \sim Q(Y_n^i, \bullet)$
  `// Sample nonlinear Markov chain`
  for $i = 1, \dots, N$ **do**
    $B^i \sim \text{Bernoulli}(\varepsilon)$ `// Sample binary jump/no jump random variable`
    if $B^i == 0$ **then**
      Set $X_{n+1}^i \sim K(X_n^i, \bullet)$ `// No jump; evolve according to` K
    else
      Set $\overline{Y}_{n+1} = \{Y_{n+1}^1, \dots, Y_{n+1}^N\}$
      Sample $X_{n+1}^i \sim J_{m(\overline{Y}_{n+1})}(X_n^i, \bullet)$ `// Jump; sample a new position.`

Additionally, let us note that the decision to use an auxiliary Markov chain is highly pragmatic. It is possible to develop "autonomous" nonlinear MCMC algorithms of the form $\widetilde{\mu}_{n+1} = K_{\widetilde{\mu}_n}$ (see e.g. [47] Ch. 5) with strong theoretical guarantees but bad empirical performance. A primary reason for this is *sample degeneracy*; due to the properties of nonlinear interaction, in an autonomous nonlinear Markov kernel, quite often a single particle X_n^i will be given a large potential $G(X_n^i)$ which results in the jump interaction being concentrated on a single point. From this point onwards, the algorithm will be unable to generate enough diversity within its particles to correctly estimate the variance of the target measure. Due to the close relationship of nonlinear MCMC and nonlinear filtering, these issues have been studied in many settings such as [62, 48] and using auxiliary dynamics is indeed a common solution.

A.2 Efficient Software Implementation

In Algorithm 1, we gave a high-level pseudocode implementation of the nonlinear sampler. In this implementation, for the sake of notational clarity, we used `for` loops to sample the individual particles. However, with modern single instruction multiple data (SIMD) computing frameworks

such as GPU accelerators, this is highly inefficient; one can, and should, parallelize all `for` loops except the outer "time" loop in Algorithm 1.

We have done this using the variety of powerful tools provided by the `JAX` library [63] within Python such as `vmap` which allows for automatic vectorization, `jit` compilation, and seamless targeting of GPU accelerators. `vmap` is particularly useful for automatic batch-wise and sample-wise vectorization in the case of stochastic gradient MCMC while writing functions in-terms of single inputs and outputs. We have also leveraged the library of linear MCMC algorithms provided by the `jax-bayes` library https://github.com/jamesvuc/jax-bayes. All experiments were run on a single Titan RTX 3090 GPU; the code can be found at https://github.com/jamesvuc/nonlinear-mcmc-paper.

B Linear MCMC Sampling Algorithms

Markov chain Monte Carlo algorithms form an integral part of Bayesian inference. In this section, we will review the basic MCMC algorithms used in this paper. In each case, we assume a C^1, strictly positive target density π known only up to a normalizing constant.

B.1 Unadjusted Langevin Algorithm

Consider the (overdamped) continuous Langevin diffusion [64]

$$\mathrm{d}X_t = \nabla \log \pi(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t$$

where B_t is a standard Brownian motion. This is a fundamental stochastic process with far reaching consequences in many areas of math; in particular, the Langevin diffusion is a Markov process with π as a stationary measure [64]. We obtain the unadjusted Langevin algorithm, or ULA, by applying an Euler-Maruyama discretization with stepsize $\delta > 0$ to the SDE above yielding [64]

$$X_{n+1} = X_n + \delta \nabla \log \pi(X_n) + \sqrt{2\delta} Z_n, \quad Z_n \sim \mathcal{N}(0, I_d).$$

This is a popular and well-studied algorithm for MCMC, although it is known to be *biased* in the sense that the sationary measure of this Markov chain is not π [65]. However, this bias converges to 0 as $\delta \to 0$ [16].

B.2 Metropolis-Adjusted Langevin Algorithm

The Metropolis-Adjusted Langevin algorithm (MALA) [18, 19], like all basic Metropolis-Hastings MCMC methods, consists of two steps: a proposal step and an accept step. In the proposal step, a candidate next state starting from the current state X_n is sampled according to the Langevin dynamics above, i.e. $\widetilde{X}_{n+1} = X_n + \delta \nabla \log \pi(X_n) \sqrt{2\delta} Z_n$. We will write $q(y|x)$ for the proposal distribution; in this case $q(y|x) = \mathcal{N}(x + \delta \nabla \log \pi(x); 2\delta)(y)$. However, unlike ULA, MALA consists of a second step that will accept the proposal $\widetilde{X}_{n+1}$ with probability

$$\alpha(X_n, \widetilde{X}_{n+1}) := 1 \wedge \frac{\pi(\widetilde{X}_{n+1}) q(X_n|\widetilde{X}_{n+1})}{\pi(X_n) q(\widetilde{X}_{n+1}|X_n)}.$$

In other words

$$X_{n+1} = \begin{cases} \widetilde{X}_{n+1} & \text{if } U_n \leq \alpha(X_n, \widetilde{X}_{n+1}) \\ X_n & \text{otherwise} \end{cases}; \quad U_n \sim [0,1].$$

B.3 RMS-Unadjusted Langevin Algorithm

The ULA is a powerful technique for "black-box" MCMC in which one only has access to gradient information about the target density. However, for very high dimensional problems in which π is highly anisotropic, it can be inefficient to simulate an isotropic diffusion such as the Langevin algorithm. A simple and effective technique, borrowed from the optimization literature in which the same phenomenon can cause problems, is to *precondition* the dynamics as follows. First we maintain an exponentially smoothed squared-gradient estimate r_n defined as

$$r_{n+1} = \beta r_n + (1-\beta) \nabla \log \pi(X_n)^2; \quad \beta \in [0, 1[$$

where the gradient squared is meant element-wise. Then, as in RMSProp [66], we can construct an adaptive stepsize by dividing by the square root of r_n as follows

$$\widehat{\delta}_{n+1} := \frac{\delta}{\sqrt{r_{n+1} + \epsilon}}$$

and then using this stepsize in the Langevin update

$$\begin{aligned} X_{n+1} &= X_n + \widehat{\delta}_{n+1} \nabla \log \pi(X_n) + \sqrt{2\widehat{\delta}_{n+1}} Z_n \\ &= X_n + \frac{\delta}{\sqrt{r_{n+1} + \epsilon}} \nabla \log \pi(X_n) + \sqrt{2 \frac{\delta}{\sqrt{r_{n+1} + \epsilon}}} Z_n. \end{aligned}$$

This has the effect of using gradient information to scale the stepsize of original Langevin diffusion independently along each dimension, which likely reduces the negative impacts of anisotropy and substantially increases mixing rate. As studied in in [33], this algorithm, which we call RMS-ULA, is biased but this bias can be controlled with δ.

B.4 RMS-MALA.

Similar to the progression from ULA to MALA, we can "metropolize" the RMS-ULA algorithm to correct the bias. This follows the same structure as the MALA, except that the proposal depends on r[7]

$$\begin{aligned} r' &= \beta r + (1 - \beta) \nabla \log \pi(x)^2 \\ q(y|x) &= \mathcal{N}\left(x + \frac{\delta}{\sqrt{r' + \epsilon}} \nabla \log \pi(X_n), 2 \frac{\delta}{\sqrt{r' + \epsilon}} \right). \end{aligned}$$

B.5 Practical Considerations for Bayesian Neural Networks: Reversibility & Tempering.

In practical settings, the high dimensional, anisotropic characteristics and limited computation budget of Bayesian neural networks (BNNs) necessitate some modifications to the above algorithms.

The first is that the Metropolis-Hastings (MH) step is rarely used except in simple settings. This is because, while ensuring that that π is the invariant measure, the reversibility of the MH step is too inefficient when we cannot affort to reject many samples. Because they cannot "backtrack", nonreversible dynamics are generally much more efficient than reversible dynamics [67, 68, 64, 69]. This has led to the recent interest in using piecewise deterministic Markov processes [70–73] and other nonreversible MCMC dynamics such as [74] for high dimensional Bayesian inference. See [69], Section 3.1 for an interesting dicussion on nonreversibility and efficiency.

The second modification is tempering. A tempered version of a distribution π is $\pi_\beta \propto \pi^\beta$. This is a pragmatic solution to speed up exploration of a target distribution when the noise component of Langevin-like algorithms causes slow or unstable dynamics. In the limit as $\beta \to \infty$, sampling from π_∞ is equivalent to the maximum *a posteriori* estimate (this is simulated annealing [75]). In practice, tempering can be achieved by simply scaling the noise by $\sqrt{\tau}$

$$X_{n+1} = X_n + \delta \nabla \log \pi(X_n) + \sqrt{2\delta}\sqrt{\tau} Z_n, \quad Z_n \sim \mathcal{N}(0, I_d).$$

which targets π^β where $\beta = 1/\tau$.[8] See also [32] for a discussion of tempering in BNNs.

C Experimental Details

No hyperparameter sweeps were used in the any of these experiments. Hyperparameters were chosen based on reasonable guesses and minimal manual tuning.

[7] A "proper" setup would expand the state space to include be (x, r), which would restore the Markov property

[8] To see this, for $\pi_\beta \propto \pi^\beta$ we have $\nabla \log \pi_\beta = \beta \nabla \log \pi$ so by using a stepsize $\delta' = \delta/\beta$ we get $X_{n+1} = X_n + \beta\delta' \nabla \log \pi(X_n) + \sqrt{2\delta'} Z_n = X_n + \delta \nabla \log \pi(X_n) + \sqrt{2\delta/\beta} Z_n$ and $\tau = 1/\beta$.

C.1 2d Toy Experiments

C.1.1 Maximum Mean Discrepancy

We use the maximum mean discrepancy (MMD) metric [58] as a way to quantitatively evaluate our MCMC algorithms. The MMD metric is an integral probability metric of the form

$$\|\mu - \nu\|_{MMD} = \sup_{\|f\|_{\mathcal{H}} \leq 1} |\mu(f) - \nu(f)| = \sup_{\|f\|_{\mathcal{H}} \leq 1} |\mathbb{E}_{X\sim\mu}[f(X)] - \mathbb{E}_{Y\sim\nu}[f(Y)]|$$

where $\mathcal{H}$ is a reproducing kernel Hilbert space (RHKS) associated to a positive-definite kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$; see [58] for details. For our purposes, the important features of the MMD metric are that 1) it is similar to the total variation metric, which optimizes over the unit ball in $\mathcal{B}_b(\mathbb{R}^d)$ instead of $\mathcal{H}$; and 2) the MMD metric $\|\mu - \nu\|_{MMD}$ can be efficiently empirically estimated from samples of μ, ν. This is because, according to Lemma 6 in [58], we have

$$\|\mu - \nu\|^2_{MMD} = \mathbb{E}_{X,X'\sim\mu\otimes\mu}[k(X, X')] - 2\mathbb{E}_{X,Y\sim\mu\otimes\nu}[k(X, Y)] + \mathbb{E}_{Y,Y'\sim\nu\otimes\nu}[k(Y, Y')]$$

which depends only on the kernel function k, and an *unbiased* estimator of $\|\mu - \nu\|^2_{MMD}$ is

$$\|\mu-\nu\|^2_{MMD} \approx \frac{1}{N_\mu(N_\mu - 1)} \sum_{i,j=1,\ j\neq i}^{N_\mu} k(X^i, X^j) - 2\frac{1}{N_\mu N_\nu} \sum_{i=1}^{N_\mu}\sum_{j=1}^{N_\nu} k(X^i, Y^j) + \frac{1}{N_\nu(N_\nu - 1)} \sum_{i,j=1,\ j\neq i}^{N_\nu} k(Y^i, Y^j)$$

where $X^i \overset{iid}{\sim} \mu$ for $i = 1, \ldots, N_\mu$ and $Y^i \overset{iid}{\sim} \nu$ for $i = 1, \ldots, N_\nu$.

If we treat μ as the distribution μ_n of a (linear or nonlinear) MCMC algorithm at step n and ν as π, then we can apply this estimator provided we can sample from π exactly. This is not feasible for complex high-dimensional distributions, but it is possible for the toy distributions we have chosen, which are all mixtures or transformations of Gaussians.

C.1.2 Experimental Setup

Refer to Table 2 for the experimental settings. Additionally, we used $10,000$ samples from π and the kernel

$$k(x, y) = \exp(-\|x - y\|^2) + \exp(-2\|x - y\|^2)$$

to estimate the MMD as in the previous section.

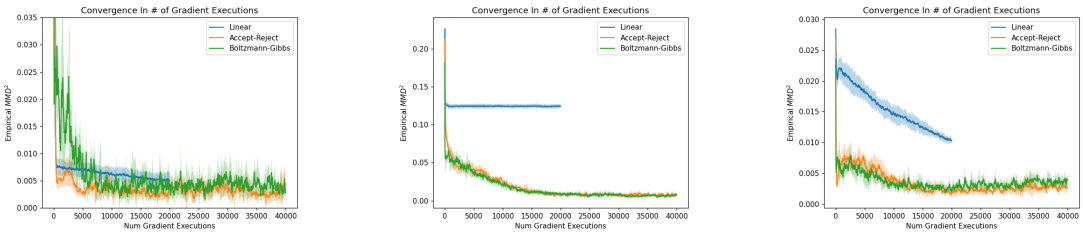
Table 2: Experimental settings for the 2d toy experiments.

Setting	Symbol	Value
Auxiliary/Linear Markov kernel	Q	MALA
Auxiliary/Linear kernel stepsize	δ_{aux}	0.001
Auxiliary/Linear target density	$\eta^\star$	$\mathcal{N}(0, \sigma^2 I_2)$ with $\sigma = 4$ for circular MoG and two rings and $\sigma = 20$ for grid MoG
Primary Markov kernel	K	MALA
Primary kernel stepsize	δ	0.001
Number of samples	N	2000
Initial Auxiliary/Linear distribution	η_0	$\mathcal{U}([-7.5, 7.5] \times [-7.5, 7.5])$
Initial Primary distribution	η_0	$\mathcal{U}([-7.5, 7.5] \times [-7.5, 7.5])$
Jump probability	ε	0.1
Number of simulation steps	n_{sim}	$10,000$

C.1.3 Runtime Analysis

We also report on the performance vs runtime of the linear and nonlinear algorithms measured in terms of number of gradient executions and the wallclock time.

In general, the nonlinear methods will require $2\times$ the number of gradient executions as the linear methods due to the requirement that we sample from the auxiliary chain in addition to the primary chain. In Figure 3, we show the MMD-squared metric plotted against the number of gradient



Circular MoG Two Rings Grid MoG

Figure 3: MMD^2 performance vs number of gradient executions (per sample) for the two-dimensional examples.

evaluations. Since we ran all algorithms for a fixed number of steps, this amounts to using $2\times$ the gradient evaluations for the nonlinear methods, although not $2\times$ the steps for a given particle. Despite this, the nonlinear methods offer better performance per gradient evaluation.

In Table 3, we also report the wallclock time for each of the methods and each of the distributions. We omit the first twenty iterations in our calculation since these can be slow while JAX compiles the program and we only want to measure the steady-state execution performance.

Target Density	**Sampler**	**Wall time (s)**	**Slowdown vs Linear**
Circular MoG	Linear	$0.97_{\pm 0.02}$	-
	AR	$2.57_{\pm 0.06}$	2.64 $\times$
	BG	$2.99_{\pm 0.01}$	3.08 $\times$
Two Rings	Linear	$0.97_{\pm 0.02}$	-
	AR	$2.57_{\pm 0.06}$	2.65 $\times$
	BG	$3.18_{\pm 0.01}$	3.27 $\times$
Grid MoG	Linear	$1.33_{\pm 0.01}$	-
	AR	$2.66_{\pm 0.01}$	2.00 $\times$
	BG	$3.70_{\pm 0.01}$	2.79 $\times$

Table 3: Wall Time analysis of the two-dimensional sampling problems (mean ± 1 standard deviation on 5 trials). We measure the steady-state wall time by omitting the first 20 iterations to account for the JAX jit-compilation time at the start of the program.

C.1.4 Comparison With Algorithms from [1]

Thanks to a very helpful suggestion from an Anonymous Reviewer, we conducted a comparison of the nonlinear MCMC algorithms proposed in this paper with those proposed in the "parent" work of this paper, i.e. [1]. Below, we detail the implementation details as they are nontrivial, and the results we obtained. See also Figure 4 for the visual results.

Implementation Details. As detailed in Appendix A.1, the key difference between algorithms in this work and those in [1] is how one constructs the empirical measure η_n^N for indexing $K_{\eta_n^N}$ in (3). Recall, ours uses a fixed batch of samples, and the algorithm from [1] uses all past samples from a single trajectory. In other words, our approach uses $\eta_n^N = m(\{Y_n^1, \ldots, Y_n^N\})$ whereas [1] uses $\eta_n^n = m(\{Y_0, \ldots, Y_n\})$.

This decision to use a dynamic number of samples is well-motivated from a theoretical point of view in [1], but is highly suboptimal from an implementation standpoint. The reason is that using a dynamic number of samples, i.e. n in the case of [1], results in frequent memory reallocations which are costly operations in common software frameworks and especially on GPUs. Indeed, modern software frameworks and accelerators have extensive optimizations for fixed-size (i.e. batched) workloads and this approach is contrary to this paradigm. In fact, this work was motivated by the need to understand how fixed sample sizes would perform in the setting introduced by [1].

Nevertheless, to implement the algorithm in [1] as a benchmark, we were faced with the classical memory-speed trade-off in software engineering. On the one hand, one can disable jit-compilation

in JAX and accept the memory-allocation costs to use a variable-size array to store the samples. Alternately, one can aggressively use more memory to speed up the computation by pre-allocating enough device memory for all future samples and apply masking to do computations on a subset of those samples. The former approach yielded a slowdown of $\sim 140\times$ on the 2-dimensional examples, which was deemed unacceptable. Hence we used the pre-allocation method at the cost of allocating a $N \times n \times d$-`float32` tensor. For small d, i.e. 2 in our case, this was acceptable at $2,000 \times 10,000 \times 2 \times 4$ =152MB. However, for larger-scale problems such as the CIFAR10 experiments below, this would be completely intractable with either approach.

Bias-Implementation Trade-off. The results of our comparison experiment unequivocally support the bias-speed trade-off we expect in our work. From Figure 4, we see that the algorithms from [1] are considerably more stable and have lower variance *and* bias. This is explainable by comparing the effective number of samples each method uses – ours uses $N = 2,000$ throughout, but theirs effectively uses $N = 10,000$ at the end of simulation. In our main result Theorem 1, we show that the bias of our algorithm decays like $1/N$ for fixed timestep n, therefore using more samples in our method would result in a less-biased approximation of π. On the other hand, as shown in [1], their algorithm is asymptotically *unbiased*. Despite this desirable property, the discussion above clearly shows why accepting *some* bias is required for practical applications of this type of algorithm.

In Figure 5, we show the results of a study matching the $N = 10,000$ samples used by the algorithms from [1] in our own algorithms. The results align completely with the predictions made by Theorem 1; i.e. the bias of our methods decreases significantly. We see it is also comparable to that of the methods in [1] for many of the samplers and distributions used in these experiments. We note that this study is a qualitative example of the fundamental difference between nonlinear and linear MCMC algorithms: increasing N directly impacts the convergence rate of our algorithms but has *no effect* on the convergence of the linear ones. This is due to the interacting (i.e. nonlinear) nature of our MCMC methods, and is supported by all the theoretical results we have developed so far.

Convergence Rate. From Figure 4, we see empirically that our algorithms have better convergence rates than those in [1]. While the results in [1] are asymptotic and hence make no claim about rate of convergence, this aligns with our intuition of how these samplers work. In particular, using past samples — including from the very beginning of simulation — can slow down the convergence since one expects that *future* samples will be of higher quality than *past* samples for a *converging* algorithm. In [1], they use a burn-in period but of course this does nothing to help with convergence rate. On the other hand, Theorem 1 upper-bounds convergence by the "pseudo-geometric" rate $\mathcal{O}(n\rho^n)$ for $\rho < 1$.

C.2 CIFAR10 Experiments

C.2.1 Experimental Setup

In this experiment, we implement a Bayesian neural network for image classification on the CIFAR10 dataset [76]. The neural network architecture we use is a standard ResNet-18 architecture [61] using the default settings implemented by the DeepMind `haiku` library [77]. The likelihood function $P(y|x, \theta)$ is the standard crossentropy on the logits y given an input image x and parameters θ. We also use standard data augmentation techniques including standardizing the images by the mean and variance, random crops and random flips. See the Table 4 for the experimental parameters.

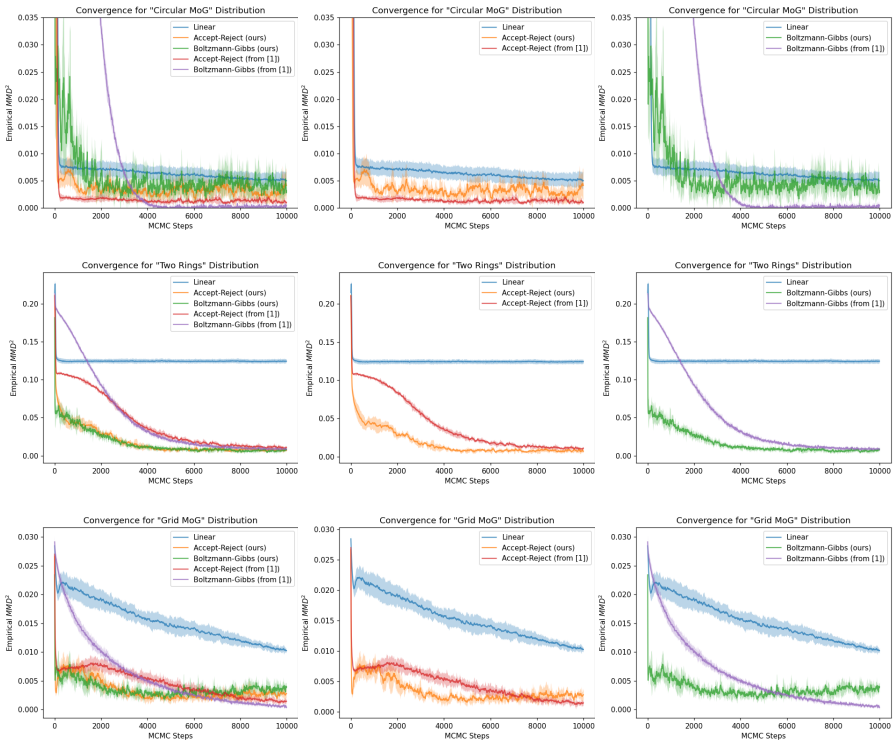
C.2.2 Additional Plots

See Figure 6 for the plot of CIFAR10 test accuracy during sampling for the non-tempered case.

C.2.3 Calibration Analysis

Calibration of a classifier is a measure of how well that classifier's logits represent probabilities [9]. We use this as a necessary but not sufficient test for correct sampling, since we know *a priori* that the Bayesian posterior gives true probabilities. We will use the methodology in [9], particularly "expected calibration error" and "maximum calibration error" to study the calibration of our Bayesian neural network, which we describe below.

To measure calibration, assume first that we have a test set of data $\mathcal{D}_{\text{test}} = \{(x_1, y_1), \ldots, (x_{N_{\text{test}}}, y_{N_{\text{test}}})\} \subset \mathcal{X} \times \{0, \ldots, C\}$ where $\mathcal{X}$ is the input space and there are



All Algorithms | Accept-Reject | Boltzmann-Gibbs

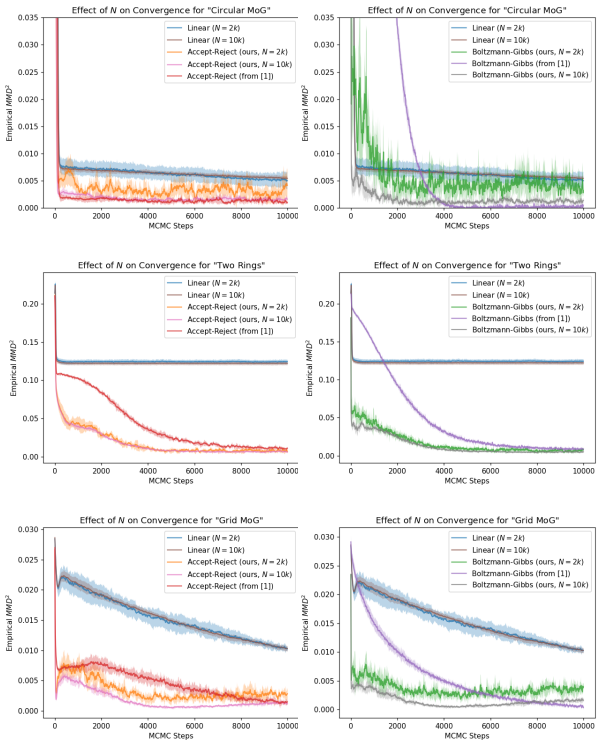
Figure 4: A comparison of the empirical MMD-squared performance for our nonlinear MCMC algorithms with $N = 2,000$ particles against those of [1] in the same setup as Section 4.1. The rows show the convergence for the Circular Mixture of Gaussians (MoG) density [59], the Two Rings density [59], and the Grid Mixture of Gaussians density [60] respectively.

Table 4: Experimental settings for the CIFAR10 experiments.

Setting	**Symbol**	**Value**
Auxiliary Markov kernel	Q	RMS-Langevin
Auxiliary kernel stepsize	$\delta_{aux}(n)$	$0.001 \times 0.1^{\lfloor n/2000 \rfloor}$
Auxiliary RMS β, ϵ parameters	N/A	$0.9, 1 \times 10^{-9}$
Auxiliary target density	$\eta^\star$	$\eta^\star \propto \pi^{0.9/\tau}$
Noise scaling (tempering)	$\sqrt{\tau}$	1×10^{-4}
Primary Markov kernel	K	Unadjusted Langevin
Primary kernel stepsize	δ	5×10^{-5}
Number of samples	N	10
Minibatch size	$\lvert\widehat{\mathcal{D}}\rvert$	256
Initial Auxiliary distribution	η_0	$\mathcal{N}(\theta_0, 0.001)$ where θ_0 is the initialized set of parameters from the `Haiku` implementation
Initial Primary distribution	μ_0	same as η_0
Jump probability	ε	0.05
Number of simulation steps	n_{sim}	50 passes through the dataset
Bayesian Prior	$P(\theta)$	$\mathcal{N}(0, 1 \times 10^{-4} I_d)$

$C \in \mathbb{Z}_{>1}$ labels. Suppose $\widehat{p} : \mathcal{X} \to \Delta^{C-1}$ is a supposed probability distribution representing $p(y|x)$. If $\widehat{p}$ were a good representation of $p(y|x)$, we would expect that the values of $\widehat{p}$ should correlate with the accuracy of $\widehat{p}$, i.e. when $\widehat{p}$ is $\alpha\%$ certain, then $\alpha\%$ of the time $\widehat{p}$ is correct. More formally, perfect calibration is when

$$P(y = y_i | \widehat{p}[y_i] = \alpha) = \alpha.$$



Accept-Reject Boltzmann-Gibbs

Figure 5: An illustration of the effect of increasing the number of particles N on our nonlinear algorithms. We show how increasing N from $N = 2,000$ in previous experiments (e.g. Figure 4) to $N = 10,000$ affects the rate of convergence and bias for our nonlinear algorithms. $N = 10,000$ was chosen purposefully to match the effective number of particles used in η_n^N for the algorithms in [1] to provide a fair comparison of bias.

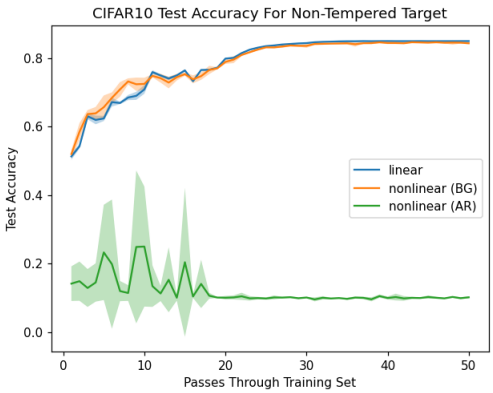


Figure 6: The full set of curves for the non-tempered CIFAR10 experiment. The AR sampler diverged early during sampling and was not able to achieve better than random performance. This shows that tempering is a useful technique for improving stability of these high-dimensional Bayesian sampling problems.

We can empirically estimate this relationship between predicted probability and true probability using histograms. Specifically, given predicted probabilities $\widehat{p}_i = \widehat{p}(y_i|x)$, we can form bins $B_k,\ k = 1, \ldots, M$ on $[0, 1]$ and estimate the accuracy in the k-th bin as A_k and the average probability in that bin as α_k. If $\widehat{p}$ represents a true probability distribution, then $A_k \approx \alpha_k$ for each $k = 1, \ldots, M$. The

expected calibration error (ECE) is then

$$ECE(\widehat{p}) := \sum_{k=1}^{M} \frac{|B_k|}{N_{\text{test}}} |A_k - \alpha_k|.$$

We can also study the maximum calibration error

$$MCE(\widehat{p}) := \max_{k=1,\ldots,M} |A_k - \alpha_k|.$$

In Table 5, we report the ECE and MCE for the CIFAR10 experiment, where now

$$\widehat{p}(y|x, \mathcal{D}) = \sum_{i=1}^{N} p(y|x, \theta^i) \approx \int p(y|x, \theta) p(\mathrm{d}\theta|\mathcal{D}).$$

Table 5: Calibration errors for the CIFAR10 experiment. 5 runs were used, and $\pm$ represents one standard deviation. Both ECEs and tempered MCE appear to be statistically significantly lower for the nonlinear than the linear algorithm, indicating better calibration in those cases. The ECE numbers have been multiplied by 10^2 in this table.

	Expected Calibration Error ($\downarrow$)		**Max Calibration Error** ($\downarrow$)	
Algorithm	Non-Tempered	Tempered	Non-Tempered	Tempered
Linear	$0.24_{\pm 0.02}$	$0.26_{\pm 0.014}$	$3.78_{\pm 0.43}$	$4.88_{\pm 0.85}$
Nonlinear (BG)	$0.14_{\pm 0.03}$	$0.16_{\pm 0.03}$	$3.21_{\pm 0.75}$	$3.72_{\pm 0.40}$
p-values BG vs Linear	**0.00063**	**0.00028**	0.22143	**0.03953**
Nonlinear (AR)	*Diverged*	$0.15_{\pm 0.05}$	*Diverged*	$3.45_{\pm 1.09}$
p-values AR vs Linear	-	**0.000748**	-	0.28993

C.2.4 Distribution Shift

One useful outcome of using the probabilistic paradigm for ML is robustness to distribution shift. This is well-documented [8, 9]; traditional ML models tend to be overconfident even in settings where they have no hope of making good predictions. In short, traditional ML models lack the ability to say "I don't know" in a way that probabilistic ML models do not. This is particularly salient in real-world systems such as self-driving cars, where the distribution of images is highly nonstationary and the costs for overconfident predictions are high.

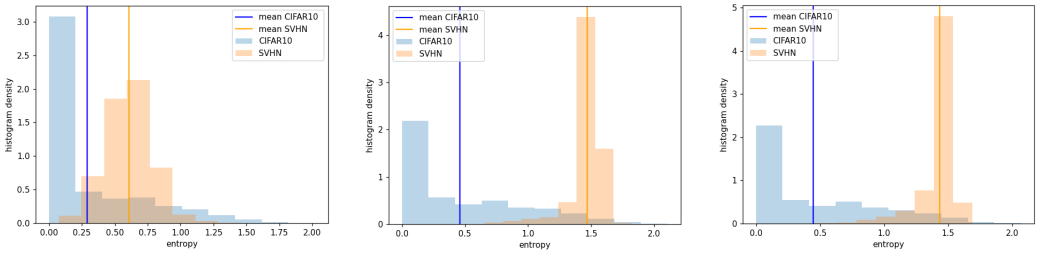
In Figure 7 below, we show how the predictive entropy differs between in-domain examples, i.e. the CIFAR10 test set, and out-of-domain examples. For the out-of-domain examples, we use the SVHN dataset [78] which also contains 32×32 RGB images, but this time of housing numbers (i.e. the digits 0-9) rather than objects such as airplanes or dogs. The entropy (a measure of uncertainty) is calculated as

$$H(\mathbf{p}) = -\sum_{i=1}^{10} p_i \log p_i$$

where $\mathbf{p} \in \Delta^9$ is the vector of probabilities for a prediction $p(y|x, \mathcal{D}_{\text{train}})$. In the traditional ML case, the probabilities are made with $p(y|x, \theta^*)$. We average $H(\mathbf{p})$ over the test data for the in-domain and out-of-domain examples. The optimized method was trained to the same test accuracy (85%) using essentially the same hyperparameters (learning rate, etc).

C.2.5 Runtime Analysis

In Figure C.2.5, we plot the performance of the tempered target vs the number of gradient evaluations. As described in Appendix C.1.3, the nonlinear algorithm uses $2\times$ the gradient computations. When plotting the eval performance vs the number of computations, we see that the linear algorithm is significantly more efficient. Further work into designing nonlinear samplers that achieve better performance (such as in the two-dimensional examples above) is required to make this trade-off worthwhile in practice.



Optimized Linear Nonlinear (BG)

Figure 7: Demonstration of the in/out of domain performance of optimized, linear MCMC, and nonlinear MCMC. The linear and nonlinear methods perform essentially the same, which is *expected* since they both approximate the same distribution $P(y|x, \mathcal{D}_{\text{train}})$

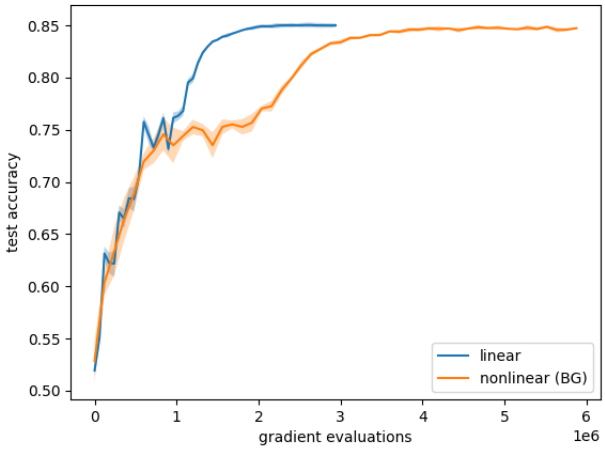


Figure 8: CIFAR10 test performance plotted against number of gradient evaluations (per sample) on the tempered target density, for the linear and nonlinear (BG) MCMC algorithms.

D Notation & Assumptions

The remaining sections will be devoted to proving the theoretical results contained in this paper. They should be read in sequence.

Note that, while some of the assumptions found below appear in [1], our analysis and the results we obtain are new. In particular, [1] studies a different MCMC algorithm based on K_η, see Appendix A for a discussion of algorithm differences. Moreover, they obtain an asymptotic "strong law of large numbers" result whereas we obtain nonasymptotic mean field long-time convergence and uniform propagation of chaos results, which are qualitatively different results using different proof techniques.

D.1 Notation and Definitions

D.1.1 Probability Spaces, Measures, and Kernels

We will be working on the measurable space $(\mathbb{R}^d, \mathscr{B}(\mathbb{R}^d))$ with $\mathscr{B}(\mathbb{R}^d)$ the Borel σ-algebra[9]; let $\mathcal{P}(\mathbb{R}^d)$ denote the space of probability measures on $\mathscr{B}(\mathbb{R}^d)$. Let $\mu \in \mathcal{P}(\mathbb{R}^d)$, $f : \mathbb{R}^d \to \mathbb{R}$ be a measurable function, and, $K : \mathbb{R}^d \times \mathscr{B}(\mathbb{R}^d) \to [0, 1]$ be a Markov kernel. Throughout the sequel,

[9] This choice is merely for simplicity of exposition; most if not all our results will hold for any polish space and its Borel σ-algebra.

we will write

$$\mu(f) := \int f(x)\mu(\mathrm{d}x), \quad Kf(x) := \int f(y)K(x,\mathrm{d}y), \quad \mu K(\mathrm{d}y) := \int \mu(\mathrm{d}x)K(x,\mathrm{d}y).$$

An obvious consequence of this is the notation $\mu K(f) = \int f(y)\mu(\mathrm{d}x)K(x,\mathrm{d}y)$. We will also need the following subset of $\mathcal{P}(\mathbb{R}^d)$ for our results: let $G, U : \mathbb{R}^d \to [0,\infty[$ and fix constants $m_G, M_U > 0$; then we define

$$\mathcal{P}_{m_G,M_U}(\mathbb{R}^d) := \{\mu \in \mathcal{P}(\mathbb{R}^d) \mid \mu(G) > m_G,\ \mu(U) \leq M_U\}.$$

We will abbreviate $\mathcal{P}_{m_G,M_U}(\mathbb{R}^d) = \mathcal{P}_{m,M}(\mathbb{R}^d)$ when there is no chance of confusion.

We will also frequently use the following notation for the empirical measure associated to $\overline{y} := \{y^1,\dots,y^N\} \subset \mathbb{R}^d$:

$$m(\overline{y}) := \frac{1}{N}\sum_{i=1}^{N} \delta_{y^i}$$

with $\delta_x \in \mathcal{P}(\mathbb{R}^d)$ the Dirac measure. Additionally, we will need the notion of tensor products for functions and measures: let $q \in \mathbb{N}$, $f_i : \mathbb{R}^d \to \mathbb{R}$ and $\mu_i \in \mathcal{P}(\mathbb{R}^d)$ for $i = 1,\dots,q$. Then for $x = (x^1,\dots,x^q) \in (\mathbb{R}^d)^q$ and measurable $g : (\mathbb{R}^d)^q \to \mathbb{R}$

$$f_1 \otimes \cdots \otimes f_q(x) := f_1(x^1)\cdots f_q(x^q), \ \ \mu_1 \otimes \cdots \otimes \mu_q(g) := \int g(x^1,\dots,x^q)\mu_1(\mathrm{d}x^1)\cdots\mu_q(\mathrm{d}x^q).$$

When $f = f_1 = \cdots = f_q$ and $\mu = \mu_1 = \cdots = \mu_q$, we will write $f^{\otimes q}$ and $\mu^{\otimes q}$ respectively. Finally, we write $\mu \ll \nu$ to mean that ν dominates μ and we write $\frac{\mathrm{d}\mu}{\mathrm{d}\nu}$ for the Radon Nikodym derivative. The notation $\mu \sim \nu$ means that $\mu \ll \nu$ and $\nu \ll \mu$.

D.1.2 Norms

Denote by $\mathcal{B}_b(\mathbb{R}^d)$ the set of bounded measurable functions $f : \mathbb{R}^d \to \mathbb{R}$. We will be working with a family of norms on functions, and its dual norms on probability measures, which is parameterized by functions of the form $U : \mathbb{R}^d \to [1,\infty[$. For such a U, we define

$$\|f\|_U := \sup_{x\in\mathbb{R}^d} \frac{|f(x)|}{U(x)}$$

for $f : \mathbb{R}^d \to \mathbb{R}$. The dual norm to $\|f\|_U$ corresponds to the weighted total variation distance; for $\mu,\nu \in \mathcal{P}(\mathbb{R}^d)$ we have

$$\|\mu - \nu\|_{tv,U} := \sup_{\|f\|_U \leq 1} |\mu(f) - \nu(f)|.$$

Note that we could replace the condition $\|f\|_U \leq 1$ by $|f| \leq U$. For $V : \mathbb{R}^d \to [0,\infty[$, we will often work with $V_\beta(x) := 1 + \beta V(x)$ for $\beta > 0$; in this case, we will write $\|f\|_\beta := \|f\|_{V_\beta}$ and $\|\bullet\|_{tv,\beta} := \|\bullet\|_{tv,V_\beta}$. We will also need the definition of the maximum oscillation of $f : \mathbb{R}^d \to \mathbb{R}$, which we denote and define as $\mathrm{osc}(f) := \sup\{|f(x) - f(y)| \mid x,y \in \mathbb{R}^d\}$.

For Markov kernels K, Q on $\mathbb{R}^d$, we obtain the weighted kernel distance

$$\|K - Q\|_{ker,U} := \sup_x \frac{\|K(x,\bullet) - Q(x,\bullet)\|_{tv,U}}{U(x)}.$$

It is worth noting the special case $U \equiv \mathbf{1}$ which corresponds to the usual total variation distance; in this case we will write $\|f\|_\infty$ and $\|\mu - \nu\|_{tv}$ in place of $\|f\|_U$ and $\|\mu - \nu\|_{tv,U}$ respectively. Lastly, we will use the notation $\epsilon(K)$ to denote the contraction coefficient of a Markov kernel K defined as $\epsilon(K) := \inf\{c \in [0,1] \mid \|\mu K - \nu K\|_{tv} \leq c\|\mu - \nu\|_{tv}\ \forall \mu,\nu \in \mathcal{P}\}$ see, e.g. [47] Ch 4.

D.2 Assumptions

Assumption 1 (Drift and Minorization)**.** *Let $K, Q : \mathbb{R}^d \times \mathscr{B}(\mathbb{R}^d) \to [0,1]$ be Markov kernels.*

K1 *K satisfies the drift criterion*

$$KV(x) \leq aV(x) + b$$

for $a \in]0,1[$, $b > 0$, and $V : \mathbb{R}^d \to [0, \infty[$ with $\lim_{\|x\| \to \infty} V(x) = \infty$.

K2 *K satisfies the following uniform minorization condition on the level sets of V: there exists $\overline{\gamma} \in]0,1[$, $\nu \in \mathcal{P}(\mathbb{R}^d)$, and $R > 2b/(1-a)$ s.t.*

$$\inf_{\{x \mid V(x) \leq R\}} K(x, A) \geq \overline{\gamma}\nu(A) \quad \forall A \in \mathscr{B}(\mathbb{R}^d).$$

Q1 *Q satisfies the drift criterion*

$$QU(x) \leq \xi U(x) + c$$

for $\xi \in]0,1[$, $c > 0$, and $U : \mathbb{R}^d \to [1, \infty[$ with $\lim_{\|x\| \to \infty} U(x) = \infty$.

Q2 *Q satisfies the following uniform minorization condition on the level sets of U: there exists $\zeta \in]0,1[$, $\nu' \in \mathcal{P}(\mathbb{R}^d)$, $R' > 2c/(1-\xi)$ s.t.*

$$\inf_{\{x \mid U(x) \leq R'\}} Q(x, A) \geq \zeta\nu'(A) \quad \forall A \in \mathscr{B}(\mathbb{R}^d).$$

The requirements in Assumption 1 are from [57] and are by now fairly standard in the Markov chain literature. They will in particular imply the following properties for K and Q respectively:

Proposition 1 (Basic Properties of K, Q; [57]&[1])**.**

1. *Suppose Assumptions 1-K1,K2 hold. Then there exists $\gamma \in]0,1[$ and $\beta > 0$ s.t. $\forall \mu, \nu \in \mathcal{P}(\mathbb{R}^d)$, we have*
$$\|\mu K - \nu K\|_{tv,\beta} \leq \gamma \|\mu - \nu\|_{tv,\beta}.$$

2. *Suppose that Assumptions 1-Q1,Q2 hold. Let $\eta_n = \eta_0 Q^n$ for $\eta_0 \in \mathcal{P}(\mathbb{R}^d)$, and $r \in]0,1]$. If $\eta_0(U^r) < \infty$, then for $r \in]0,1[$ there are constants $M(r) > 0$, $\delta \in]0,1[$ s.t.*
$$\|\eta_n - \eta^\star\|_{tv,U^r} \leq M(r)\delta^n. \tag{7}$$

Proof.

1. This is directly from [57].

2. This uses Lemma 1 which shows Assumption 1 Q1,Q2 implies the assumptions in Lemma C.1 from [1]. Then we combine this with Lemma 2. Both lemmas can be found in Appendix E.

□

Throughout the remainder of this paper, β and γ will be fixed and we will adopt the notation

$$V_\beta(x) := 1 + \beta V(x).$$

Additionally, references to $M(r)$ and δ are meant in the sense of the above proposition. Next, we introduce some criteria that ensure that the kernels K, Q are "compatible" in-terms of their drift criteria from Assumption 1.

Assumption 2 (Compatibility)**.**

C1 *There is $r^* \in]0,1]$ s.t. $V_\beta(x) \leq U(x)^{r^*}$ $\forall x \in \mathbb{R}^d$ with V_β as above.*

C2 *G satisfies the lower bound compatibility criterion with U: for every $R > 0$*

$$\theta(R) := \inf\{G(x) \mid x \in \mathbb{R}^d,\ U(x) \leq R\} > 0.$$

Assumption 2-C1 is also present in [1], and ensures that V and U are sufficiently "comparable". Assumption 2-C2 is a novel assumption that we will use to obtain an *a priori* lower bound on $\eta_n(G)$. Finally, some straightforward boundedness assumptions on G.

Assumption 3 (Assumptions on G)**.**

G1 *G is bounded, i.e. $\|G\|_\infty < \infty$.*

G2 *G is bounded in the weighted supremum norm for V_β; i.e. $\|G\|_\beta < \infty$.*

Not all of the results below depend on all of the assumptions in this section. We will make clear which of these assumptions are invoked in each result.

E Long-Time Convergence

We will now state the main long-time convergence result for the mean field system. We will revisit this theorem at the end of this section and provide a full proof. We will often use the equivalent "distribution flow" interpretation of system (2) defined as

$$\begin{cases} \eta_{n+1} = \eta_n Q \\ \mu_{n+1} = \mu_n K_{\eta_{n+1}} \end{cases} \tag{8}$$

for our proofs.

Theorem 2. *Suppose that Assumption 1 and Assumption 2-C1 hold. Suppose also that J_η satisfies*

$$\|J_\eta - J_{\eta'}\|_{ker,\beta} \leq C_J \|\eta - \eta'\|_{tv,\beta} \tag{9}$$

for some constant $C_J > 0$ and that

$$K_\eta V(x) \leq \widetilde{a} V(x) + \widetilde{b} \tag{10}$$

for $\widetilde{a} \in]0,1[$, $\widetilde{b} > 0$ and all $\eta \in \mathcal{P}_{m_G, M_U}$ for suitably chosen constants m_G, M_U. Suppose δ is from (7) and set

Case 1: *$\rho := (1-\varepsilon)\gamma$ if $J_\eta(x, \mathrm{d}y)$ doesn't depend on x; or*

Case 2: *$\rho := (1-\varepsilon)\gamma + \varepsilon \|J_{\eta^\star} V\|_V$ if $J_\eta(x, \mathrm{d}y)$ does depend on x.*

If $\mu_0(V), \eta_0(U) < \infty$ then there exists a constant $C > 0$ s.t.

$$\|\mu_n - \pi\|_{tv,\beta} \leq \rho^n \|\mu_0 - \pi\|_{tv,\beta} + Cn \max(\rho, \delta)^n.$$

In particular, if $\rho < 1$, we have $\lim_{n\to\infty} \mu_n = \pi$ in V_β-total variation.

E.1 Results About Weighted Total Variation

Lemma 1. *Suppose that a Markov kernel P satisfies a drift condition*

$$PV(x) \leq aV(x) + b$$

for $V : \mathbb{R}^d \to [0, \infty[$, $V(x) \to \infty$ as $\|x\| \to \infty$, and the minorization condition with $\epsilon \in]0,1[$, $\nu \in \mathcal{P}(\mathbb{R}^d)$, and $\mathcal{C} := \{x \mid V(x) \leq R\}$ s.t.

$$\inf_{x \in \mathcal{C}} P(x, A) \geq \epsilon \nu(A).$$

holds for some $R > 2K/(1-a)$. Then there exists $\overline{a} \in]a, 1[$, $\overline{b} > 0$, $\overline{\nu} \in \mathcal{P}(\mathbb{R}^d)$, $S \in \mathscr{B}(\mathbb{R}^d)$ s.t.

$$PV(x) \leq \overline{a} V(x) + b \mathbb{1}_{\{x \in S\}}, \quad \inf_{x \in S} P(x, A) \geq \overline{\epsilon}\, \overline{\nu}(A).$$

Proof. This argument merely an elaboration on the remark from the end of [57]. Let $\overline{a} \in]a, 1[$ such that $R > b/(\overline{a} - a)$ and $S = \{x \mid V(x) \leq R\}$. If $x \notin S$ then

$$\begin{aligned} PV(x) \leq aV(x) + b &= aV(x) + \frac{b}{V(x)} V(x) \leq aV(x) + \frac{b}{R} V(x) \\ &\leq aV(x) + b\frac{\overline{a} - a}{b} V(x) = aV(x) + (\overline{a} - a)V(x) \\ &= \overline{a} V(x). \end{aligned}$$

If $x \in S$ then $PV(x) \leq aV(x) + b \leq \overline{a}V(x) + b$. Hence the choices are clear from fixing R as above. □

Lemma 2. *Let P be a Markov kernel with invariant measure π and suppose there are constants $\rho \in]0,1[,\ C > 0$, and $r \in]0,1]$ s.t.*

$$\|P^n f - \pi(f)\|_{V^r} \leq C\rho^n \|f\|_{V^r}$$

for any $f : \mathbb{R}^d \to \mathbb{R}$ with $\|f\|_{V^r} < \infty$. Then if $\mu_0(V^r) < \infty$ and μ_n is the flow of P then there is a constant $C' > 0$ s.t.

$$\|\mu_n - \pi\|_{tv,V^r} \leq C'\rho^n.$$

Proof. Consider

$$\begin{aligned}
\|\mu_n - \pi\|_{tv,V^r} &= \sup_{\|f\|_{V^r}\leq 1} |\mu_0 P^n(f) - \pi(f)| \\
&= \sup_{\|f\|_{V^r}\leq 1} |\mu_0(P^n f - \pi(f))| \\
&\leq \sup_{\|f\|_{V^r}\leq 1} \mu_0(|P^n f(x) - \pi(f)|) \\
&\leq \sup_{\|f\|_{V^r}\leq 1} \mu_0(V^r(x)\|P^n f - \pi(f)\|_{V^r}) \\
&\leq C\rho^n \mu_0(V^r) = C'\rho^n.
\end{aligned}$$

□

Lemma 3. *Let P, Q be Markov kernels and suppose that P satisfies a drift condition*

$$PV(x) \leq aV(x) + b$$

for $V : \mathbb{R}^d \to [1, \infty[$. Then

$$\|\mu P - \nu P\|_{tv,V} \leq (a+b)\|\mu - \nu\|_{tv,V}$$

and

$$\|\mu P - \mu Q\|_{tv,V} \leq \mu(V)\|P - Q\|_{ker,V}.$$

Proof. For the first part:

$$\|\mu P - \nu P\|_{tv,V} = \sup_{\|f\|_V\leq 1} |\mu P(f) - \nu P(f)| \leq \sup_{\|f\|_V, \|h\|_V\leq 1} \|Pf\|_V |\mu(h) - \nu(h)|$$

and

$$\begin{aligned}
\|Pf\|_V &= \sup_x \frac{|Pf(x)|}{V(x)} = \sup_x \frac{|\int P(x, \mathrm{d}y) f(y)|}{V(x)} = \sup_x \frac{|\int P(x, \mathrm{d}y) V(y)[f(y)/V(y)]}{V(x)} \\
&\leq \|f\|_V \sup_x \frac{|PV(x)|}{V(x)} \leq \sup_x \frac{aV(x) + b}{V(x)} \leq a + b.
\end{aligned}$$

For the second part, let $\|f\|_V \leq 1$

$$|\mu P(f) - \mu Q(f)| \leq \mu(|Pf - Qf|) \leq \mu(V\frac{|Pf - Qf|}{V}) \leq \mu(V)\|Pf - Qf\|_V \leq \mu(V)\|P - Q\|_{ker,V}.$$

□

E.2 Proof of Theorem 2

Theorem 2. *Suppose that Assumption 1 and Assumption 2-C1 hold. Suppose also that J_η satisfies*

$$\|J_\eta - J_{\eta'}\|_{ker,\beta} \leq C_J \|\eta - \eta'\|_{tv,\beta} \tag{9}$$

for some constant $C_J > 0$ and that

$$K_\eta V(x) \leq \widetilde{a} V(x) + \widetilde{b} \tag{10}$$

for $\widetilde{a} \in]0,1[,\ \widetilde{b} > 0$ and all $\eta \in \mathcal{P}_{m_G, M_U}$ for suitably chosen constants m_G, M_U. Suppose δ is from (7) and set

Case 1: $\rho := (1-\varepsilon)\gamma$ *if* $J_\eta(x, \mathrm{d}y)$ *doesn't depend on* x*; or*

Case 2: $\rho := (1-\varepsilon)\gamma + \varepsilon\|J_{\eta^\star}V\|_V$ *if* $J_\eta(x, \mathrm{d}y)$ *does depend on* x*.*

If $\mu_0(V), \eta_0(U) < \infty$ *then there exists a constant* $C > 0$ *s.t.*

$$\|\mu_n - \pi\|_{tv,\beta} \le \rho^n \|\mu_0 - \pi\|_{tv,\beta} + Cn\max(\rho,\delta)^n.$$

In particular, if $\rho < 1$*, we have* $\lim_{n\to\infty}\mu_n = \pi$ *in* V_β*-total variation.*

Proof. Case 1: For $n = 1$, consider

$$\begin{aligned}
\|\mu_1 - \pi\|_{tv,\beta} &= \|\mu_0 K_{\eta_1} - \pi K_{\eta^\star}\|_{tv,\beta} \\
&\le (1-\varepsilon)\|\mu_0 K - \pi K\|_{tv,\beta} + \varepsilon\|\mu_0 J_{\eta_1} - \pi J_{\eta^\star}\|_{tv,\beta} \\
&= (1-\varepsilon)\|\mu_0 K - \pi K\|_{tv,\beta} + \varepsilon\|J_{\eta_1} - J_{\eta^\star}\|_{tv,\beta} \\
&\le (1-\varepsilon)\gamma\|\mu_0 - \pi\|_{tv,\beta} + \varepsilon C_J\|\eta_1 - \eta^\star\|_{tv,\beta} \\
&\le \rho\|\mu_0 - \pi\|_{tv,\beta} + \varepsilon C_J M(r^*)\delta \\
&\le \rho\|\mu_0 - \pi\|_{tv,\beta} + C\max(\rho,\delta)
\end{aligned}$$

because from Assumption 2-C1 we have [10]

$$\|\eta - \eta'\|_{tv,\beta} \le \|\eta - \eta'\|_{tv,U^{r^*}}.$$

Hence assume true for $n-1$. Then for n:

$$\begin{aligned}
\|\mu_n - \pi\|_{tv,\beta} &\le (1-\varepsilon)\|\mu_{n-1}K - \pi K\|_{tv,\beta} + \varepsilon\|\mu_{n-1}J_{\eta_n} - \pi J_{\eta^\star}\|_{tv,\beta} \\
&= (1-\varepsilon)\|\mu_{n-1}K - \pi K\|_{tv,\beta} + \varepsilon\|J_{\eta_n} - J_{\eta^\star}\|_{tv,\beta} \\
&\le (1-\varepsilon)\gamma\|\mu_{n-1} - \pi\|_{tv,\beta} + \varepsilon C_J\|\eta_n - \eta^\star\|_{tv,\beta} \\
&\le \rho(\rho^{n-1}\|\mu_0 - \pi\|_{tv,\beta} + C(n-1)\max(\rho,\delta)^{n-1}) + \varepsilon C_J M(r^*)\delta^n \\
&\le \rho^n\|\mu_0 - \pi\|_{tv,\beta} + C[(n-1)\rho\max(\rho,\delta)^{n-1} + \max(\rho,\delta)^n] \\
&\le \rho^n\|\mu_0 - \pi\|_{tv,\beta} + C[(n-1)\max(\rho,\delta)^n + \max(\rho,\delta)^n] \\
&= \rho^n\|\mu_0 - \pi\|_{tv,\beta} + Cn\max(\rho,\delta)^n.
\end{aligned}$$

hence Case 1 is done.

Case 2: We proceed the same way as Case 1 except we need to use

$$\begin{aligned}
\|\mu_{n-1}J_{\eta_n} - \pi J_{\eta^\star}\|_{tv,\beta} &\le \|\mu_{n-1}J_{\eta_n} - \mu_{n-1}J_{\eta^\star}\|_{tv,\beta} + \|\mu_{n-1}J_{\eta^\star} - \pi J_{\eta^\star}\|_{tv,\beta} \\
&\le \mu_{n-1}(V_\beta)\|J_{\eta_n} - J_{\eta^\star}\|_{tv,\beta} + \|J_{\eta^\star}V_\beta\|_\beta\|\mu_{n-1} - \pi\|_{tv,\beta}
\end{aligned}$$

using Lemma 3. Note that since $K_\bullet$ satisfies a uniform drift criterion for V, using linearity we have

$$\mu_n(V_\beta) \le \overline{a}^n\mu_0(V_\beta) + \frac{b}{1-\overline{a}} \le \mu_0(V_\beta) + \frac{b}{1-\overline{a}} =: C_0.$$

Hence for $n = 1$:

$$\begin{aligned}
\|\mu_1 - \pi\|_{tv,\beta} &= \|\mu_0 K_{\eta_1} - \pi K_{\eta^\star}\|_{tv,\beta} \\
&\le (1-\varepsilon)\|\mu_0 K - \pi K\|_{tv,\beta} + \varepsilon\|\mu_0 J_{\eta_1} - \pi J_{\eta^\star}\|_{tv,\beta} \\
&\le (1-\varepsilon)\|\mu_0 K - \pi K\|_{tv,\beta} + \varepsilon\mu_0(V_\beta)\|J_{\eta_1} - J_{\eta^\star}\|_{tv,\beta} + \varepsilon\|J_{\eta^\star}V_\beta\|_\beta\|\mu_0 - \pi\|_{tv,\beta} \\
&\le [(1-\varepsilon)\gamma + \varepsilon\|J_{\eta^\star}V_\beta\|_\beta]\|\mu_0 - \pi\|_{tv,\beta} + \varepsilon C_0 C_J\|\eta_1 - \eta^\star\|_{tv,\beta} \\
&\le \rho\|\mu_0 - \pi\|_{tv,V} + C\max(\rho,\delta).
\end{aligned}$$

Hence assume true for $n-1$. Then for n:

$$\begin{aligned}
\|\mu_n - \pi\|_{tv,\beta} &\le (1-\varepsilon)\|\mu_{n-1}K - \pi K\|_{tv,\beta} + \varepsilon\|\mu_{n-1}J_{\eta_n} - \pi J_{\eta^\star}\|_{tv,\beta} \\
&\le (1-\varepsilon)\|\mu_{n-1}K - \pi K\|_{tv,\beta} + \varepsilon\mu_{n-1}(V_\beta)\|J_{\eta_n} - J_{\eta^\star}\|_{tv,\beta} + \varepsilon\|J_{\eta^\star}V_\beta\|_\beta\|\mu_{n-1} - \pi\|_{tv,\beta} \\
&\le (1-\varepsilon)\gamma\|\mu_{n-1} - \pi\|_{tv,\beta} + \varepsilon\mu_{n-1}(V_\beta)\|J_{\eta_n} - J_{\eta^\star}\|_{tv,\beta} + \varepsilon\|J_{\eta^\star}V_\beta\|_\beta\|\mu_{n-1} - \pi\|_{tv,\beta} \\
&\le \rho(\rho^{n-1}\|\mu_0 - \pi\|_{tv,\beta} + C(n-1)\max(\rho,\delta)^{n-1}) + \varepsilon C_0 C_J M(r^*)\delta^n \\
&\le \rho^n\|\mu_0 - \pi\|_{tv,\beta} + C[(n-1)\rho\max(\rho,\delta)^{n-1} + \max(\rho,\delta)^n] \\
&\le \rho^n\|\mu_0 - \pi\|_{tv,\beta} + C[(n-1)\max(\rho,\delta)^n + \max(\rho,\delta)^n] \\
&= \rho^n\|\mu_0 - \pi\|_{tv,\beta} + Cn\max(\rho,\delta)^n.
\end{aligned}$$

hence Case 2 is done. □

[10] this is because $\|f\|_{U^{r^*}} \le \|f\|_\beta \implies \|\mu - \nu\|_{tv,\beta} \le \|\mu - \nu\|_{tv,U^{r^*}}$

F Large-Particle Convergence

We study the behaviour of the interacting particle system (3) as $N \to \infty$. The behaviour of this system is probabilistically different from that of the mean field system (2) because the particles in (3) are *coupled* whereas the particles in (2) are *independent*. The fact that we recover independence in the limit of a collection of interchangeable particles is a remarkable feature of interacting particle systems [41].

F.1 Result

For a collection of N particles, under suitable assumptions on $K_\bullet$, we will show that any fixed-size q-block of particles of $\{X_n^1, \dots, X_n^q\}$ becomes independent as $N \to \infty$, and moreover this trend towards independence happens *uniformly* in time. This phenomenon is called the uniform *propagation of chaos* property[11].

Let us describe the dynamics of the distribution of the IPS (3). From (3), each X_n^i evolves according to

$$X_n^i \sim K_{m(Y_n)}(X_{n-1}^i, \bullet)$$

which indicates that, given $Y_n := (Y_n^1, \dots, Y_n^N)$, X_n^i is sampled independently. Letting $X_n^{q,N} := (X_n^1, \dots, X_n^q)$ and $\mu_n^{q,N} := \text{Distribution}(X_n^{q,N}) \in \mathcal{P}((\mathbb{R}^d)^q)$, for measurable $f : (\mathbb{R}^d)^q \to \mathbb{R}$ one has

$$\mu_n^{q,N}(f) = \mathbb{E}[f(X_n^{q,N})] = \mathbb{E}[\mathbb{E}[f(X_n^{q,N})|Y_n]] = \mathbb{E}[\mu_{n-1}^{q,N} K_{m(Y_n)}^{\otimes q}(f)]$$

where the expectation is taken over the distribution of $Y_n = (Y_n^1, \dots, Y_n^N)$, which is $\eta_n^{\otimes N}$. We will use this decomposition to derive a uniform propagation of chaos result in the following theorem.

Theorem 3. *Let $N \in \mathbb{N}$, $q \in \{1, \dots, N\}$, and consider the interacting particle system $X_n := (X_n^1, \dots, X_n^N)$, $Y_n := (Y_n^1, \dots, Y_n^N)$ from (3). Let $\mu_n^{q,N} := \text{Distribution}(X_n^1, \dots, X_n^q)$, and let μ_n be the distribution of the (independent) mean field system (2), with $X_0^i \sim \mu_0$. Suppose that $\forall x \in (\mathbb{R}^d)^q$, $\eta \in \mathcal{P}(\mathbb{R}^d)$, and $f \in \mathcal{B}_b((\mathbb{R}^d)^q)$ with $osc(f) \leq 1$*[12]

$$\left| \mathbb{E}[J_{m(Y)}^{\otimes q} f(x)] - J_\eta^{\otimes q} f(x) \right| \leq c \frac{q^2}{N} \mathcal{R}(q^2/N), \text{ where } Y = \{Y^1, \dots, Y^N\}, \ Y^i \sim \eta.$$

Suppose finally that $\mu_0^{q,N} = \mu_0^{\otimes q}$ for any $1 \leq q \leq N$.

If **(Case 1)***: $J_\eta(x, \bullet)$ doesn't depend on x, or if* **(Case 2)***: $J_\eta(x, \bullet)$ does depend on x but additionally that $\epsilon(K) < 1$, then there exists a fixed constant $C > 0$ s.t.*

$$\sup_{n \geq 0} \|\mu_n^{q,N} - \mu_n^{\otimes q}\|_{tv} \leq C \frac{q^2}{N} \mathcal{R}(q^2/N).$$

Proof. **Case 1**: We first claim that

$$\|\mu_{n+1}^{q,N} - \mu_{n+1}^{\otimes q}\|_{tv} \leq c\varepsilon \sum_{j=0}^{n-1} (1-\varepsilon)^j \epsilon(K^{\otimes q})^j \cdot \frac{q^2}{N} \mathcal{R}(q^2/N).$$

Let $f \in \mathcal{B}_b((\mathbb{R}^d)^q)$ s.t. $osc(f) \leq 1$.

[11] here, "chaos" is synonymous with "statistical independence", coming from the statistical physics intuition that a collection of independent particles are maximally disordered, or chaotic. This means that particles which start chaotic will approximately "propagate their chaos" through time despite interactions between the particles

[12] due to the characterization $\|\mu - \nu\|_{tv} = \sup\{|\mu(f) - \nu(f)| \mid f \in \mathcal{B}_b(\mathbb{R}^d), \ \text{osc}(f) \leq 1\}$ from [47] this regularity condition should be interpreted as a total variation Lipschitzness analogous to (9).

Consider the following expression for general n:

$$\begin{aligned}
\left|\mu_{n+1}^{q,N}(f) - \mu_{n+1}^{\otimes q}(f)\right| &= \left|\mathbb{E}[\mu_n^{q,N} K_{m(Y_n)}^{\otimes q}(f)] - \mu_n^{\otimes q} K_{\eta_n}^{\otimes q}(f)\right| \\
&= \left|\mathbb{E}[\mu_n^{q,N} K_{m(Y_n)}^{\otimes q}(f)] - \mu_n^{q,N} K_{\eta_n}^{\otimes q}(f) + \mu_n^{q,N} K_{\eta_n}^{\otimes q}(f) - \mu_n^{\otimes q} K_{\eta_n}^{\otimes q}(f)\right| \\
&\leq \left|\mathbb{E}[\mu_n^{q,N} K_{m(Y_n)}^{\otimes q}(f)] - \mu_n^{q,N} K_{\eta_n}^{\otimes q}(f)\right| + \left|\mu_n^{q,N} K_{\eta_n}^{\otimes q}(f) - \mu_n^{\otimes q} K_{\eta_n}^{\otimes q}(f)\right| \\
&\leq \sup_x \varepsilon \left|\mathbb{E}[J_{m(Y_n)}^{\otimes q} f(x)] - J_{\eta_n}^{\otimes q} f(x)\right| + (1-\varepsilon)\left|\mu_n^{q,N} K^{\otimes q}(f) - \mu_n^{\otimes q} K^{\otimes q}(f)\right| \\
&\leq c\varepsilon \frac{q^2}{N} \mathcal{R}(q^2/N) + (1-\varepsilon)\epsilon(K^{\otimes q})\left|\mu_n^{q,N}(f) - \mu_n^{\otimes q}(f)\right|.
\end{aligned}$$

where we have used the assumption on J.

Thus if $n = 1$ in the above expression, the base case holds since $\mu_0^{q,N} = \mu_0^{\otimes q}$. Now, if the claim holds for n, then for $n+1$ we have

$$\begin{aligned}
|\mu_{n+1}^{q,N}(f) - \mu_{n+1}^{\otimes q}(f)| &\leq c\varepsilon \frac{q^2}{N} \mathcal{R}(q^2/N) + (1-\varepsilon)\epsilon(K^{\otimes q}) c\varepsilon \sum_{j=0}^{n-1} (1-\varepsilon)^j \epsilon(K^{\otimes q})^j \cdot \frac{q^2}{N} \mathcal{R}(q^2/N) \\
&= c\varepsilon \frac{q^2}{N} \mathcal{R}(q^2/N) \left[1 + (1-\varepsilon)\epsilon(K^{\otimes q}) \sum_{j=0}^{n-1} (1-\varepsilon)^j \epsilon(K^{\otimes q})^j\right] \\
&= c\varepsilon \frac{q^2}{N} \mathcal{R}(q^2/N) \sum_{j=0}^{n} (1-\varepsilon)^j \epsilon(K^{\otimes q})^j.
\end{aligned}$$

Thus the claim holds for $n+1$. Since $(1-\varepsilon)\epsilon(K^{\otimes q}) < 1$, we have the result.

Case 2: We first claim that

$$\|\mu_{n+1}^{q,N} - \mu_{n+1}^{\otimes q}\|_{tv} \leq c\varepsilon \sum_{j=0}^{n-1} [\varepsilon + (1-\varepsilon)\epsilon(K^{\otimes q})]^j \cdot \frac{q^2}{N} \mathcal{R}(q^2/N).$$

Let $f \in \mathcal{B}_b((\mathbb{R}^d)^q)$ with $osc(f) \leq 1$. We proceed by induction. Note that:

$$\begin{aligned}
\left|\mu_{n+1}^{q,N}(f) - \mu_{n+1}^{\otimes q}(f)\right| &= \left|\mathbb{E}[\mu_n^{q,N} K_{m(Y_n)}^{\otimes q}(f)] - \mu_n^{\otimes q} K_{\eta_n}^{\otimes q}(f)\right| \\
&= \varepsilon \left|\mathbb{E}[\mu_n^{q,N} J_{m(Y_n)}^{\otimes q}(f)] - \mu_n^{\otimes q} J_{\eta_n}^{\otimes q}(f)\right| + (1-\varepsilon)\left|\mu_n^{q,N} K(f) - \mu_n^{\otimes q} K(f)\right|
\end{aligned}$$

for the first term, we will use the decomposition

$$\left|\mathbb{E}[\mu_n^{q,N} J_{m(Y_n)}(f)] - \mu_n^{\otimes q} J_{\eta_n}(f)\right| \leq \left|\mathbb{E}[\mu_n^{q,N} J_{m(Y_n)}(f)] - \mu_n^{q,N} J_{\eta_n}(f)\right| + \left|\mu_n^{q,N} J_{\eta_n}(f) - \mu_n^{\otimes q} J_{\eta_n}(f)\right|$$

Then, using the assumption on J, we have

$$\left|\mathbb{E}[\mu_n^{q,N} J_{m(Y_n)}(f)] - \mu_n^{q,N} J_{\eta_n}(f)\right| \leq c\frac{q^2}{N} \mathcal{R}(q^2/N))$$

and also we know we can use

$$\begin{aligned}
\left|\mu_n^{q,N} J_{\eta_n}(f) - \mu_n^{\otimes q} J_{\eta_n}(f)\right| &\leq \mathrm{osc}(J_{\eta_n} f)\left|\mu_n^{q,N}(h) - \mu_n^{\otimes q}(h)\right| \\
&\leq \epsilon(J_{\eta_n})\mathrm{osc}(f)\left|\mu_n^{q,N}(h) - \mu_n^{\otimes q}(h)\right| \\
&\leq \left|\mu_n^{q,N}(h) - \mu_n^{\otimes q}(h)\right|
\end{aligned}$$

since $\epsilon(J_{\eta_n}) \leq 1$, with $\mathrm{osc}(h) \leq 1$. Hence putting these together, we see

$$\begin{aligned}
\left|\mu_{n+1}^{q,N}(f) - \mu_{n+1}^{\otimes q}(f)\right| &\leq \varepsilon c \frac{q^2}{N} \mathcal{R}(q^2/2N) + \varepsilon\left|\mu_n^{q,N}(h) - \mu_n^{\otimes q}(h)\right| + (1-\varepsilon)\epsilon(K^{\otimes q})\left|\mu_n^{q,N}(f) - \mu_n^{\otimes q}(f)\right| \\
&\leq \varepsilon c \frac{q^2}{N} \mathcal{R}(q^2/2N) + (\varepsilon + (1-\varepsilon)\epsilon(K^{\otimes q}))\|\mu_n^{q,N} - \mu_n^{\otimes q}\|_{tv}.
\end{aligned}$$

Now if $n = 0$ in the above expression, the base case holds since $\mu_0^{q,N} = \mu_0^{\otimes q}$. If it is true for n, then for $n+1$ we have

$$\begin{aligned}
&\left|\mu_{n+1}^{q,N}(f) - \mu_{n+1}^{\otimes q}(f)\right| = \left|\mathbb{E}[\mu_n^{q,N} K_{m(Y_n)}(f)] - \mu_n^{\otimes q} K_{\eta_n}(f)\right| \\
&\leq c\varepsilon \frac{q^2}{N}\mathcal{R}(q^2/N) + (\varepsilon + (1-\varepsilon)\epsilon(K^{\otimes q}))\|\mu_n^{q,N}(f) - \mu_n^{\otimes q}(f)\|_{tv} \\
&\leq c\varepsilon \frac{q^2}{N}\mathcal{R}(q^2/N) + (\varepsilon + (1-\varepsilon)\epsilon(K^{\otimes q}))\left[c\varepsilon \sum_{j=0}^{n-1}[\varepsilon + (1-\varepsilon)\epsilon(K^{\otimes q})]^j \cdot \frac{q^2}{N}\mathcal{R}(2q^2/N)\right] \\
&= c\varepsilon \frac{q^2}{N}\mathcal{R}(q^2/N)\left[1 + \sum_{j=1}^{n}[\varepsilon + (1-\varepsilon)\epsilon(K^{\otimes q})]^j\right] \\
&= c\varepsilon \frac{q^2}{N}\mathcal{R}(q^2/N)\sum_{j=0}^{n}[\varepsilon + (1-\varepsilon)\epsilon(K^{\otimes q})]^j.
\end{aligned}$$

Hence if $\epsilon(K^{\otimes q}) < 1$, i.e. $\epsilon(K) < 1$, the result holds. □

F.2 Proof of Main Results (Theorem 1 & Corollary 1)

Theorem 1. *[Convergence of Nonlinear MCMC] Under suitable conditions on K_η and Q, there exist fixed constants $C_1, C_2, C_3 > 0$, a function $\mathcal{R} : [0, \infty[\to [1, \infty[$, and $\rho > 0$ s.t.*

$$\|\mu_n^N - \pi\|_{tv} \leq C_1 \frac{1}{N}\mathcal{R}(1/N) + C_2\rho^n + C_3 n\rho^n.$$

♦

Proof. This follows straightforwardly from the above discussion, the only technicality is converting the results from Theorem 2 to the un-weighted total variation. But note that since $\{\|f\|_\infty \leq 1\} \subset \{\|f\|_\beta \leq 1\}$ we have

$$\|\mu_n - \mu_\infty\|_{tv} = \sup_{\|f\|_\infty \leq 1} |\mu_n(f) - \mu_\infty(f)| \leq \sup_{\|f\|_\beta \leq 1} |\mu_n(f) - \mu_\infty(f)| = \|\mu_n - \mu_\infty\|_{tv,\beta}$$

so we're done. □

Corollary 1. *[Adapted from [41], Theorem 2.2] Suppose that Theorem 1 applies to K_η. Let $\overline{X}_n := \{X_n^1, \ldots, X_n^N\}$ be the interacting particle system from (3). Then for every $n \in \mathbb{N}$ and $f \in \mathcal{B}_b(\mathbb{R}^d)$ we have*

$$\lim_{N\to\infty} \mathbb{E}\left[\left|\frac{1}{N}\sum_{i=1}^{N} f(X_n^i) - \mu_n(f)\right|\right] = 0.$$

♦

Proof. Let $f \in \mathcal{B}_b(\mathbb{R}^d)$ and consider

$$\begin{aligned}
\mathbb{E}[(m(X_n)(f) - \mu_n(f))^2] &= \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} f(X_n^i) - \mu_n(f)\right)^2\right] \\
&= \frac{1}{N^2}\sum_{i=1}^{n}\mathbb{E}[f(X_n^i)f(X_n^j)] - \frac{2}{N}\sum_{i=1}^{N}\mathbb{E}[f(X_n^i)]\mu_n(f) + \mu_n(f)^2 \\
&= \frac{1}{N}\mathbb{E}[f(X_n^1)^2] + \frac{N-1}{N}\mathbb{E}[f(X_n^1)f(X_n^2)] - 2\mathbb{E}[f(X_n^1)]\mu_n(f) + \mu_n(f)^2
\end{aligned}$$

using interchangeability of the X_n^is. Since the strong propagation of chaos in Theorem 3 clearly implies weak propagation of chaos, i.e. for any bounded f $|\mu_n^{q,N}(f) - \mu_n^{\otimes q}(f)| \to 0$, we see that the expression above $\to 0$ and we have L^2-convergence which implies weak convergence. □

G Analysis of the Kernels K_η^{BG} and K_η^{AR}

G.1 Analysis of K_η^{BG}

G.1.1 Statement of Results

A key difficulty that arises when working with $\Psi_G(\eta) = \frac{G\mathrm{d}\eta}{\eta(G)}$ is obtaining a uniform lower bound the denominator $\eta(G)$. This is important for deriving uniform regularity results and for generally establishing conditions under which the measures $\{\Psi_G(\eta_n)\}_{n=0}^\infty$ are well-defined. An effective approach used in [47] and related works is to assume the uniform lower bound $G(x) > \epsilon > 0\ \forall x \in \mathbb{R}^d$ for some $\epsilon > 0$. While this assumption is self-contained in that it works for any choice of η, it eliminates ubiquitous families of measures such as Gaussians, e.g. $G(x) = \exp(-\|x\|^2/2)$ and $\mathrm{d}\eta = \mathrm{d}x$.

We will see below how to relax $G(x) > \varepsilon > 0$ using the structure of our problem. In particular, we can use the Lyapunov function U for Q to control the probability that the state $Y_n \sim \eta_n$ will venture "far away" from the center of the state space. This leads to the intuition that, if G is sufficiently "compatible" with U, i.e. the function G stays away from zero in the "center" of the state space as determined by U, then the expectation $\eta_n(G) = \mathbb{E}_{Y_n\sim\eta_n}[G(Y_n)]$ can be lower-bounded using the same Lyapunov function. This insight is encoded in the compatibility criterion Assumption 2-C2, and we can use it to prove the following *a priori* lower bound on $\eta_n(G)$.

Lemma 4. *Suppose that Assumption 1-Q1 and Assumption 2-C2 hold, and let $\eta_n = \eta_{n-1}Q$ with $\eta_0(U) < \infty$. Then we have the lower bound*

$$\eta_n(G) \geq \theta(R^*)\left(1 - \frac{\gamma^n\eta_0(U) + \frac{c}{1-\gamma}}{R^*}\right)$$

where R^* is fixed and doesn't depend on n.

This lemma essentially says that G should be bounded away from zero on the level sets of U. This is not a strong condition — if G is bounded away from zero on compact sets and U is continuous, the lemma applies. As an example, if $G = \exp(-\|x\|^2/2)$ and $U(x) = c\|x\|^2 + 1$ then this result applies. The constant R^* arises since $R \mapsto \theta(R)$ is nonincreasing and $R \mapsto 1 - \frac{1}{R}$ is increasing so we can optimize this bound as a function of R to get R^* (which may not be unique, but the value the bound attains will be). Together with the next lemma, we can obtain the desired convergence from Theorem 2 for the BG interaction.

Lemma 5. *Let $\eta, \eta' \in \mathcal{P}(\mathbb{R}^d)$, and suppose that Assumption 3 holds, i.e. $\|G\|_\infty, \|G\|_\beta < \infty$. Then*

$$\|\Psi_G(\eta) - \Psi_G(\eta')\|_{tv,\beta} \leq \left(\frac{\|G\|_\beta + \|G\|_\infty}{\eta(G)\vee\eta'(G)} + \eta(V)\wedge\eta'(V)\beta\frac{\|G\|_\beta\|G\|_\infty}{\eta(G)\eta'(G)}\right)\|\eta - \eta'\|_{tv,\beta}.$$

Clearly, we will use our knowledge that $\eta_n \in \mathcal{P}_{m_G,M_U}(\mathbb{R}^d)\ \forall n$ to make the Lipschitz constant in Lemma 5 uniform over $\mathcal{P}_{m_G,M_U}$.

Proposition 2. *Suppose that Assumption 1, 2, and Assumption 3-G1 hold. Then the drift criterion (10) holds uniformly over all $\eta \in \mathcal{P}_{m,M}(\mathbb{R}^d)$.*

Now we can apply Theorem 2 to obtain convergence.

Corollary 2. *Suppose that Assumptions 1, 2, 3 hold. If $\eta_0(G) > 0$, $\eta_0(U), \mu_0(V) < \infty$, then Theorem 2 holds for K^{BG}, i.e. the flow μ_n converges to π in V_β-total variation as long as $\rho = (1-\varepsilon)\gamma < 1$.*

Proof. Lemma 4 and Lemma 5 imply the regularity condition (9) for constants determined by those lemmas and by noting that, due to Asssumption 2-C1, $\eta(V_\beta) \leq \eta(U^{r^*}) \leq \eta(U)^{r^*} < \infty$ by Jensen's inequality. Additionally, Proposition 2 implies the condition (10). Hence, since we are in "Case 1" of Theorem 2, the result follows. □

G.1.2 Proofs

Lemma 4. *Suppose that Assumption 1-Q1 and Assumption 2-C2 hold, and let $\eta_n = \eta_{n-1}Q$ with $\eta_0(U) < \infty$. Then we have the lower bound*

$$\eta_n(G) \geq \theta(R^*)\left(1 - \frac{\gamma^n \eta_0(U) + \frac{c}{1-\gamma}}{R^*}\right)$$

where R^* is fixed and doesn't depend on n.

Proof.

$$\begin{aligned}
QG(x) &= \mathbb{E}_{X\sim Q(x,\bullet)}[G(X)] \\
&= \mathbb{E}_{X\sim Q(x,\bullet)}[G(X)\mathbb{1}_{\{U(X)\leq R\}}] + \mathbb{E}_{X\sim Q(x,\bullet)}[G(X)\mathbb{1}_{\{U(X)>R\}}] \\
&\geq \theta(R)\cdot \mathbb{P}_x(U(X)\leq R) + \mathbb{E}_{X\sim Q(x,\bullet)}[G(X)\mathbb{1}_{\{U(X)>R\}}] \\
&\geq \theta(R)\cdot \mathbb{P}_x(U(X)\leq R).
\end{aligned}$$

But by Markov's inequality

$$\mathbb{P}_x(U(X) > R) \leq \frac{\mathbb{E}_x[U(X)]}{R} \leq \frac{\xi U(x) + c}{R}$$

so

$$\mathbb{P}_x(U(X)\leq R) = 1 - \mathbb{P}_x(U(X) > R) \geq 1 - \frac{\xi U(x) + c}{R}$$

and hence

$$QG(x) \geq \theta(R)\mathbb{P}_x(U(X)\leq R) \geq \theta(R)\left(1 - \frac{\xi U(x) + c}{R}\right).$$

Now

$$\begin{aligned}
Q^2G(x) = Q(QG(x)) &\geq Q\left(\theta(R)\left(1 - \frac{\xi U(x) + c}{R}\right)\right) \\
&= \theta(R)\left(1 - \frac{\xi QU(x) + c}{R}\right) \geq \theta(R)\left(1 - \frac{\xi^2 U(x) + c(1+\xi)}{R}\right)
\end{aligned}$$

and iterating this procedure gives

$$Q^nG(x) \geq \theta(R)\left(1 - \frac{\xi^n U(x) + \frac{c}{1-\xi}}{R}\right)$$

where we have used the sum of the geometric series to obtain $c/(1-\xi)$. Now, $\theta(R)$ is nonincreasing w.r.t. R and $1 - 1/R$ is increasing w.r.t. R, so optimizing to get R^* and integrating we obtain

$$\eta_n(G) = \eta_0(Q^nG) \geq \theta(R^*)\left(1 - \frac{\xi^n \eta_0(U) + \frac{c}{1-\xi}}{R^*}\right).$$

□

Lemma 5. *Let $\eta, \eta' \in \mathcal{P}(\mathbb{R}^d)$, and suppose that Assumption 3 holds, i.e. $\|G\|_\infty, \|G\|_\beta < \infty$. Then*

$$\|\Psi_G(\eta) - \Psi_G(\eta')\|_{tv,\beta} \leq \left(\frac{\|G\|_\beta + \|G\|_\infty}{\eta(G)\vee\eta'(G)} + \eta(V)\wedge\eta'(V)\beta\frac{\|G\|_\beta\|G\|_\infty}{\eta(G)\eta'(G)}\right)\|\eta - \eta'\|_{tv,\beta}.$$

Proof. Let $\|f\|_\beta \leq 1$. then

$$\begin{aligned}
&|\Psi_G(\eta)(f) - \Psi_G(\eta')(f)| = \left|\int \frac{G(x)f(x)}{\eta(G)}\eta(\mathrm{d}x) - \int \frac{G(x)f(x)}{\eta'(G)}\eta'(\mathrm{d}x)\right| \\
&\leq \left|\int \frac{G(x)f(x)}{\eta(G)}\eta(\mathrm{d}x) - \int \frac{G(x)f(x)}{\eta'(G)}\eta(\mathrm{d}x)\right| + \left|\int \frac{G(x)f(x)}{\eta'(G)}\eta(\mathrm{d}x) - \int \frac{G(x)f(x)}{\eta'(G)}\eta'(\mathrm{d}x)\right|
\end{aligned}$$

for the first term

$$\begin{aligned}
&\left|\int\left(\frac{G(x)f(x)}{\eta(G)}-\frac{G(x)f(x)}{\eta'(G)}\right)\eta(\mathrm{d}x)\right| \\
&\leq \frac{|\eta(G)-\eta'(G)|}{\eta(G)\eta'(G)}\int G(x)|f(x)|\eta(\mathrm{d}x) \\
&\leq \frac{\|G\|_\beta\|\eta-\eta'\|_{tv,\beta}}{\eta(G)\eta'(G)}\int G(x)|f(x)|\eta(\mathrm{d}x) \\
&\leq \frac{\|G\|_\beta\|\eta-\eta'\|_{tv,\beta}}{\eta(G)\eta'(G)}\eta(G(1+\beta V))\|f\|_\beta \\
&= \left(\frac{\|G\|_\beta}{\eta'(G)}+\beta\eta(V)\frac{\|G\|_\beta\|G\|_\infty}{\eta(G)\eta'(G)}\right)\|\eta-\eta'\|_{tv,\beta}
\end{aligned}$$

and for the second

$$\begin{aligned}
&\left|\int\frac{G(x)f(x)}{\eta'(G)}\eta(\mathrm{d}x)-\int\frac{G(x)f(x)}{\eta'(G)}\eta'(\mathrm{d}x)\right| \\
&= \frac{1}{\eta'(G)}\left|\int G(x)f(x)\eta(\mathrm{d}x)-\int G(x)f(x)\eta'(\mathrm{d}x)\right| \\
&\leq \frac{\|fG\|_\beta}{\eta'(G)}\|\eta-\eta'\|_{tv,\beta} \\
&\leq \frac{\|G\|_\infty\|f\|_\beta}{\eta'(G)}\|\eta-\eta'\|_{tv,\beta} = \frac{\|G\|_\infty}{\eta'(G)}\|\eta-\eta'\|_{tv,\beta}
\end{aligned}$$

since

$$\|fG\|_\beta = \sup_x\frac{|f(x)G(x)|}{1+\beta V(x)} \leq \|G\|_\infty\|f\|_\beta$$

so putting these together

$$\|\Psi_G(\eta)-\Psi_G(\eta')\|_{tv,\beta} \leq \left(\frac{\|G\|_\beta+\|G\|_\infty}{\eta'(G)}+\eta(V)\beta\frac{\|G\|_\beta\|G\|_\infty}{\eta(G)\eta'(G)}\right)\|\eta-\eta'\|_{tv,\beta}$$

Using symmetry completes the proof. □

Proposition 2. *Suppose that Assumption 1, 2, and Assumption 3-G1 hold. Then the drift criterion* (10) *holds uniformly over all* $\eta \in \mathcal{P}_{m,M}(\mathbb{R}^d)$.

Proof. Let $\eta \in \mathcal{P}_{m,M}(\mathbb{R}^d)$ and consider

$$\begin{aligned}
K_\eta V(x) &= (1-\varepsilon)KV(x)+\varepsilon\Psi_G(\eta)(V) \\
&\leq (1-\varepsilon)aV(x)+(1-\varepsilon)b+\varepsilon\frac{\eta(V)}{\eta(G)} \\
&\leq (1-\varepsilon)aV(x)+(1-\varepsilon)b+\varepsilon\frac{M}{m}.
\end{aligned}$$

□

Theorem 4 ([47] Thm 8.7.1 pp.283)**.** *Suppose that* G *has bounded oscillations, let* $N \geq q \geq 1$*, and* $Y=(Y^1,\dots,Y^N)$, $Y^i \sim \eta$*. Then for any* $f \in \mathcal{B}_b((\mathbb{R}^d)^q)$ *with* $\mathrm{osc}(f) \leq 1$*, we have*

$$|\mathbb{E}[\Psi_{G^{\otimes q}}(m(Y)^{\otimes q})(f)-\Psi_{G^{\otimes q}}(\eta^{\otimes q})(f)]| \leq c\frac{q^2}{N}\mathcal{R}_{G,\eta}(2q^2/N)$$

where

$$\mathcal{R}_{G,\eta}(u) := 1+\mathrm{osc}_\eta(G)^2(1+\mathrm{osc}_\eta(G)\sqrt{u})\exp(\mathrm{osc}_\eta(G)^2u), \quad \mathrm{osc}_\eta(G) := \mathrm{osc}(G/\eta(G)).$$

In particular, by picking $G \equiv 1$*, we obtain*

$$|\mathbb{E}[m(Y)^{\otimes q}(f)-\eta^{\otimes q}(f)]| \leq c\frac{q^2}{N}.$$

Corollary 3. *Suppose that Assumption 3 G1 holds (i.e. G is bounded) the compatibility criterion Assumption 2-C2 with $\eta_n(G) \geq m$. Then $K_\bullet^{BG}$ satisfies the uniform propagation of chaos in Case 1 with*

$$\mathcal{R}(u) = \mathcal{R}_G(u) := 1 + \frac{\mathrm{osc}(G)^2}{m^2}\left(1 + \frac{\mathrm{osc}(G)}{m}\sqrt{u}\right)\exp\left(\frac{\mathrm{osc}(G)^2}{m^2}u\right)$$

Proof. This follows from Theorem 3 and Theorem 4 above, noting 1) that $\|G\|_\infty$ implies $\mathrm{osc}(G) < \infty$, and 2) that we can obtain a bound on $\mathcal{R}_{G,\eta_n}$ independent of η_n by applying Lemma 4 to obtain $\eta_n(G) \geq m$ so

$$osc_{\eta_n}(G) = \mathrm{osc}(G/\eta_n(G)) \leq \mathrm{osc}(G)/m.$$

□

G.2 Analysis of K_η^{AR}

G.2.1 Statement of Results

First, the result showing that π is $K_{\eta^\star}^{AR}$-invariant.

Proposition 3. *$K_{\eta^\star}$ is π-invariant.*

Proof. We have

$$\begin{aligned}
\pi(J_{\eta^\star} f) &= \iint [f(y) - f(x)]\alpha(x,y)\eta^\star(\mathrm{d}y)\pi(\mathrm{d}x) \\
&= \iint [f(y) - f(x)]1 \wedge \frac{\pi(y)\eta^\star(x)}{\eta^\star(y)\pi(x)}\eta^\star(\mathrm{d}y)\pi(\mathrm{d}x) + \pi(f) \\
&= \iint [f(y) - f(x)]\pi(x)\eta^\star(y) \wedge \pi(y)\eta^\star(x)\mathrm{d}y\mathrm{d}x + \pi(f) \\
&= \pi(f)
\end{aligned}$$

since the integrals with $f(y)$ and $f(x)$ in the first term are equal. □

We will establish equvalent results from Section G.1but for $K_\bullet^{AR}$. This will not require assumption 2-C2 since the interaction is well-defined for $\eta \in \mathcal{P}(\mathbb{R}^d)$ as $\alpha(x,y)$ is in fact *bounded*. K^{AR} is the main subject of study for [1], and in fact the regularity and uniform drift conditions were established there. Note that the statement $\eta \in \mathcal{P}_{0,\infty}$ is vaccuous since $G(x) > 0 \implies \eta(G) > 0$, and the second says that $\eta(V) < \infty$.

Lemma 6 ([1])**.** *Let $\eta, \eta' \in \mathcal{P}_{0,\infty}(\mathbb{R}^d)$. Then*

$$\|J_\eta^{AR} - J_{\eta'}^{AR}\|_{ker,\beta} \leq 2\|\eta - \eta'\|_{tv,\beta}.$$

Lemma 7. *Suppose that Assumption 1, 2-C2, hold. Then the uniform drift criterion (10) holds for $\eta \in \mathcal{P}_{0,M}(\mathbb{R}^d)$.*

Proposition 4. *Let $J = J^{AR}$, and $Y = \{Y^1, \ldots, Y^N\}$ where $Y^i \overset{iid}{\sim} \eta$. Then for any $f \in \mathcal{B}_b((\mathbb{R}^d)^q$ with $\mathrm{osc}(f) \leq 1$ and $x \in (\mathbb{R}^d)^q$, we have*

$$|\mathbb{E}[J_{m(Y)}^{\otimes q} f(x)] - J_\eta^{\otimes q} f(x)| \leq 2c\frac{q^2}{N}.$$

Corollary 4. *Suppose that Assumption 1, 2-C2 hold. If $\eta_0(U), \mu_0(V) < \infty$, then Theorem 2 holds for K^{BG}, i.e. the flow μ_n converges to π in V_β-total variation as long as $\rho = (1-\varepsilon)\gamma + \varepsilon\|J_{\eta_0}V_\beta\| < 1$.*

G.2.2 Proofs

Lemma 7. *Suppose that Assumption 1, 2-C2, hold. Then the uniform drift criterion (10) holds for $\eta \in \mathcal{P}_{0,M}(\mathbb{R}^d)$.*

Proof. Let $\eta \in \mathcal{P}_{0,M}(\mathbb{R}^d)$ and consider

$$\begin{aligned} K_\eta^{AR} V(x) &= (1-\varepsilon)KV(x) + \varepsilon[\eta(\alpha(x,\bullet)V) + (1 - A_\eta(x))V(x)] \\ &\leq (1-\varepsilon)aV(x) + (1-\varepsilon)b + \varepsilon\eta(V) + \varepsilon V(x) \\ &\leq [(1-\varepsilon)a + \varepsilon]V(x) + (1-\varepsilon)b + \varepsilon M. \end{aligned}$$

□

Proposition 4. *Let $J = J^{AR}$, and $Y = \{Y^1, \ldots, Y^N\}$ where $Y^i \overset{iid}{\sim} \eta$. Then for any $f \in \mathcal{B}_b((\mathbb{R}^d)^q$ with $\operatorname{osc}(f) \leq 1$ and $x \in (\mathbb{R}^d)^q$, we have*

$$|\mathbb{E}[J_{m(Y)}^{\otimes q} f(x)] - J_\eta^{\otimes q} f(x)| \leq 2c\frac{q^2}{N}.$$

Proof. Starting with the first term, for each fixed x

$$\begin{aligned} |\mathbb{E}[J_{m(Y_n)}^{\otimes q} f(x)] - J_{\eta_n}^{\otimes q} f(x)| &= \left| \mathbb{E}\left[\int [f(y) - f(x)]\alpha^{\otimes q}(x,y) m(Y_n)^{\otimes q}(\mathrm{d}y)\right] - \int [f(y) - f(x)]\alpha^{\otimes q}(x,y)\eta_n^{\otimes q}(\mathrm{d}y)\right| \\ &= |\mathbb{E}[m(Y_n)^{\otimes q}(\varphi_f(x,\bullet))] - \eta_n^{\otimes q}(\varphi_f(x,\bullet))| \end{aligned}$$

where

$$\varphi_f(x,y) := \alpha^{\otimes q}(x,y)[f(y) - f(x)].$$

Now

$$\sup_y |\varphi_f(x,y)| = \sup_y |[f(y) - f(x)]\alpha^{\otimes q}(x,y)| \leq \sup_y |f(y) - f(x)| \|\alpha^{\otimes q}(x,\bullet)\|_\infty \leq \operatorname{osc}(f) \leq 1$$

so automatically $\operatorname{osc}(\varphi_f(x,\bullet)) \leq 2$. Hence using Lemma 4

$$|\mathbb{E}[m(Y_n)^{\otimes q}(\varphi_f(x,\bullet))] - \eta_n^{\otimes q}(\varphi_f(x,\bullet))| \leq 2c\frac{q^2}{N}.$$

□

Corollary 5. *If $\epsilon(K) < 1$, then $K_\bullet^{AR}$ satisfies the uniform propagation of chaos in Case 2 of Theorem 3 with $\mathcal{R} \equiv 1$.*