Metropolis-Adjusted Diffusion Models

Kevin H. Lam *
Department of Statistics
University of Oxford

Tyler Farghly *
Inria, CNRS, ENS
PSL Research University

Christopher Williams *
Department of Statistics
University of Oxford

Jun Yang
Dept. of Math. Sciences
University of Copenhagen

Yee Whye Teh
Department of Statistics
University of Oxford

Arnaud Doucet
Department of Statistics
University of Oxford

{lam,williams,y.w.teh,doucet}@stats.ox.ac.uk,
tyler.farghly@inria.fr, jy@math.ku.dk

Abstract

Sampling from score-based diffusion models incurs bias due to both time discretisation and the approximation of the score function. A common strategy for reducing this bias is to apply corrector steps based on the unadjusted Langevin algorithm (ULA) at each noise level within a predictor-corrector framework. However, ULA is itself a *biased* sampler, as it discretises a continuous diffusion process. In this work, we consider *adjusted* Langevin correctors that employ Metropolis–Hastings (MH) or Barker's accept-reject steps to correct for this bias. Since the target density ratio typically required by MH-based algorithms is unavailable, we propose methods that instead utilise the score function to compute the correct acceptance probability. We introduce the first exact method for adjusting Langevin corrections in diffusion models, based on a two-coin Bernoulli factory algorithm. We also propose an efficient approximation based on Simpson's rule that achieves accuracy of order $5/2$ in the step size at near-zero marginal cost. We demonstrate that these procedures improve sample quality on both synthetic and image datasets, yielding consistent gains in Fréchet Inception Distance (FID) on the latter.

1 Introduction

Diffusion models have become a key class of generative models, with state-of-the-art performance in domains such as image, audio, and protein design. They rely on a simple but powerful idea. Instead of modelling the data density directly, one learns its score along a noising path and then simulates a reverse-time diffusion process to transform noise into data [Sohl-Dickstein et al., 2015, Song and Ermon, 2019, Ho et al., 2020, Song et al., 2021]. In practice, this reverse diffusion is implemented using numerical integrators [Song et al., 2021, Karras et al., 2022]. Consequently, even if the learned score were exact, the generated distribution would still be impacted by time-discretisation error. With a learned score, this error is compounded by approximation error in the score network.

This time-discretisation issue is common in Markov chain Monte Carlo (MCMC). To sample from a target distribution, the unadjusted Langevin algorithm (ULA), a discretisation of the Langevin diffusion, uses gradients of a log-density to explore the state space [Durmus and Moulines, 2017], but this does not leave the target distribution invariant. The standard approach to correct for this discretisation bias is to add an accept–reject step, yielding the Metropolis-adjusted Langevin algorithm

*Authors contributed equally to this work.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

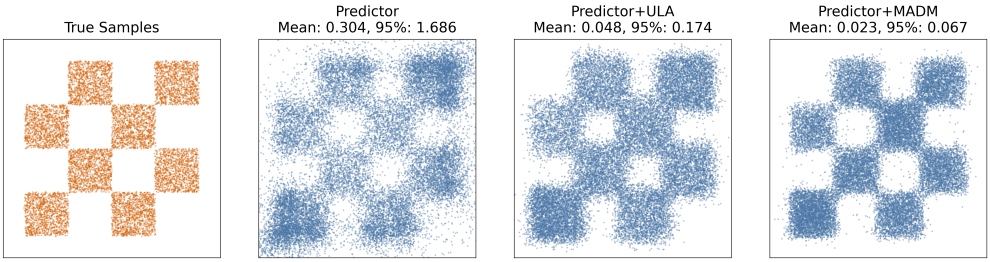

Figure 1: DDM sampling using a probability flow ODE predictor, ULA correction, and MADM correction. Mean distance to the true samples and the 95% sample containment distance are shown. A coarse diffusion schedule and large ULA stepsize are chosen to show the discretisation bias, the same schedule and stepsize are used in MADM to show the bias correction from this method.

(MALA) [Besag, 1994, Roberts and Tweedie, 1996]. The analogy with diffusion sampling is immediate. Predictor–Corrector samplers [Song et al., 2021] already use Langevin-type corrector steps at fixed noise levels: after a predictor step moves the sample between successive marginals, a corrector step refines the sample with respect to the current marginal. The usual corrector is again ULA. Thus, the "corrector" in the Predictor–Corrector scheme is itself biased. It may reduce the error introduced by the predictor, but it introduces its own discretisation error.

This motivates a natural question: can we replace ULA correctors in diffusion models by Metropolis-adjusted correctors? A direct application of MALA is not possible as it requires evaluating the density ratio $p_t(\widetilde{\mathbf{x}})/p_t(\mathbf{x})$ between the proposal $\widetilde{\mathbf{x}}$ and the current state $\mathbf{x}$ for the intractable target marginal p_t at noise level t, whereas diffusion models only provide a score estimate $s_\phi(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

To address this problem, we propose *Metropolis-Adjusted Diffusion Models* (MADM), a simple modification of the Predictor–Corrector framework where the usual ULA corrector is replaced by an accept–reject corrected Langevin step. The key point is that, although the density ratio is unavailable, its logarithm can be represented as a line integral of the score. This result shows that the score can be used not only to construct Langevin proposals, but also to estimate the acceptance probability that corrects their discretisation bias.

We develop two approaches to exploit this representation. First, we give an exact Barker adjusted corrector based on a two-coin Bernoulli factory [Gonçalves et al., 2017]. Under the assumption that $s(\cdot, t) = \nabla \log p_t(\cdot)$ and that a suitable bound on the estimator of the line integral is available, this procedure produces an accept–reject mechanism with the Barker acceptance probability without ever evaluating p_t. This gives, to our knowledge, the first exact Metropolis-style Langevin corrector for any smooth target distribution only known through its score. Second, we introduce a practical approximation of the density ratio based on Simpson's rule. The resulting MH correction is no longer exact, but it is very cheap. The resulting approximation has high-order accuracy in the Langevin step size, is easy to implement, and only requires access to a pre-trained score function.

Empirically, Metropolis adjustments address a failure mode of unadjusted correctors on toy examples; see Figure 1. On image datasets, we combined a predictor with an MADM corrector based on Simpson's rule using pre-trained models from Karras et al. [2022]. While gains are modest in some unconditional settings, we found that MADM yields consistent and systematic improvements in FID.

Our contributions are as follows:

- We introduce MADM, a Metropolis-adjusted corrector framework for diffusion models that replaces biased ULA corrector steps.
- We propose an exact method based on the Barker acceptance rule and a two-coin Bernoulli factory, giving an adjusted corrector under assumptions and a control variate formulation that enables image-scale evaluations.
- We present a scalable quadrature approximation that requires only one additional midpoint score evaluation per proposal based on Simpson's rule.
- We demonstrate that MADM correctors reduce artifacts on synthetic datasets and consistently improve FID on standard image benchmarks.

2 Background and Setup

2.1 Langevin Diffusion and Unadjusted Langevin Algorithm

Consider a smooth distribution p, then a standard technique for sampling from p is to simulate the Langevin diffusion

$$\mathrm{d}\mathbf{X}_t = \tfrac{g^2}{2}\nabla_{\mathbf{x}} \log p(\mathbf{X}_t)\,\mathrm{d}t + g\,\mathrm{d}\mathbf{B}_t, \tag{1}$$

for some parameters $g > 0$. Whatever being the initial state $\mathbf{X}_0$, this diffusion is such that the law of $\mathbf{X}_t$ converges to p as t grows without bound. One cannot simulate exactly this nonlinear continuous-time diffusion process so a standard approach is to use an Euler–Maruyama integrator

$$\mathbf{X}_{kh} = \mathbf{X}_{(k-1)h} + h\tfrac{g^2}{2}\nabla_{\mathbf{x}} \log p(\mathbf{X}_{(k-1)h}) + \sqrt{h}g\mathbf{Z}_k, \qquad \mathbf{Z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

where $h > 0$ is a step-size. This is known as the Unadjusted Langevin Algorithm (ULA). This time-discretisation introduces a bias so that we do not sample exactly from p.

2.2 Diffusion Models

A Denoising Diffusion Model (DDM) [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2021] uses a *forward SDE* to transport the data distribution p_{data} to a multivariate normal $\mathcal{N}(\mathbf{0}, \mathbf{I})$ through

$$\mathrm{d}\mathbf{X}_t = f_t\mathbf{X}_t\,\mathrm{d}t + g_t\,\mathrm{d}\mathbf{B}_t, \qquad \mathbf{X}_0 \sim p_{\text{data}}, \tag{3}$$

where $t \in [0, 1]$ and $\mathbf{B}_t$ is a multivariate standard Brownian motion. The functions f_t and g_t must ensure that $\mathbf{X}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, at least approximately, but are otherwise design choices which can be selected to optimise sample quality; see, e.g. [Williams et al., 2024]. The reverse-time process $(\mathbf{Y}_t)_{t\in[0,1]} = (\mathbf{X}_{1-t})_{t\in[0,1]}$ thus transports $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to p_{data} and satisfies

$$\mathrm{d}\mathbf{Y}_t = [-f_{1-t}\mathbf{Y}_t + g_{1-t}^2\nabla_{\mathbf{x}} \log p_{1-t}(\mathbf{Y}_t)]\,\mathrm{d}t + g_{1-t}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{Y}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{4}$$

where p_t is the distribution of $\mathbf{X}_t$. In practice, the score function $\nabla_{\mathbf{x}} \log p_t(\cdot)$ is unknown and is approximated by a neural network $s_\phi(\mathbf{x}_t, t)$ trained by denoising score matching [Vincent, 2011, Song and Ermon, 2019], that is, by regressing $s_\phi(\mathbf{X}_t, t)$ onto $\nabla_{\mathbf{x}} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)$ with $\mathbf{X}_0 \sim p_{\text{data}}$. Since (3) is linear, this conditional score is available in closed form, as

$$p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(r_t\mathbf{x}_0, r_t^2\sigma_t^2\mathbf{I}), \qquad r_t = \exp\left(\int_0^t f_u\,\mathrm{d}u\right), \qquad \sigma_t^2 = \int_0^t \frac{g_u^2}{r_u^2}\,\mathrm{d}u.$$

Different choices of f_t and g_t in Equation 3 yield the variance exploding (VE) and variance preserving (VP) SDEs, with r_t and σ_t known in closed form (see Table 1 of Karras et al. [2022]). For convenience, we have assumed the VP setting (i.e. $p_1 = \mathcal{N}(\mathbf{0}, \mathbf{I})$). Given f_t and g_t, together with a learned score and schedule, the task is to evolve Equation 4 to draw samples from the generative model.

2.3 Diffusion Predictor-Corrector Framework

Equation 4 can be split into the sum of a deterministic drift and a Langevin diffusion, in which the deterministic component defines the predictor step and the stochastic component the corrector:

$$\mathrm{d}\mathbf{Y}_t = \underbrace{\left(-f_{1-t}\mathbf{Y}_t + \tfrac{g_{1-t}^2}{2}\nabla_{\mathbf{x}} \log p_{1-t}(\mathbf{Y}_t)\right)\mathrm{d}t}_{\text{Probability Flow Prediction ODE}} + \underbrace{\left(\tfrac{g_{1-t}^2}{2}\nabla_{\mathbf{x}} \log p_{1-t}(\mathbf{Y}_t)\,\mathrm{d}t + g_{1-t}\,\mathrm{d}\mathbf{B}_t\right)}_{\text{Langevin Correction SDE}}.$$

This decomposition gives rise to the predictor–corrector framework where at each step we integrate numerically the ODE before applying the Langevin Correction SDE as described in Song et al. [2021]. The rationale is that, if $\mathbf{Y}_{t-\eta} \sim p_{1-(t-\eta)}$ then by integrating exactly the deterministic drift with an exact score over an interval of length η then we obtain $\mathbf{Y}'_t \sim p_{1-t}$ and the Langevin diffusion initialised at such state leaves this distribution invariant and so provide a new sample $\mathbf{Y}_t \sim p_{1-t}$. Despite the terminology, the Langevin "corrector" is itself defined by an SDE and must be discretised as discussed in Section 2.1. Using an Euler–Maruyama scheme, this reads

$$\mathbf{Y}'_{k\eta} = \mathbf{Y}_{(k-1)\eta} + \eta\Big(-f_{1-(k-1)\eta}\mathbf{Y}_{(k-1)\eta} + \tfrac{g_{1-(k-1)\eta}^2}{2}\nabla_{\mathbf{x}} \log p_{1-(k-1)\eta}(\mathbf{Y}_{(k-1)\eta})\Big), \tag{5}$$

$$\mathbf{Y}_{k\eta} = \mathbf{Y}'_{k\eta} + \tfrac{g_{1-k\eta}^2}{2}h\nabla_{\mathbf{x}} \log p_{1-k\eta}(\mathbf{Y}'_{k\eta}) + \sqrt{h}g_{1-k\eta}\mathbf{Z}, \qquad \mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{6}$$

Algorithm 1 MALA for Denoising Diffusion Models

Input: initial sample $\mathbf{x}_t^{(0)}$, score $s(\cdot, t)$, corrector steps K, step size h.
for $i = 0$ **to** $K - 1$ **do**
  Compute $\widetilde{\mathbf{x}}_t^{(i+1)} = \mathbf{x}_t^{(i)} + \frac{h}{2} s(\mathbf{x}_t^{(i)}, t) + \sqrt{h}\mathbf{z}^{(i)}$ with $\mathbf{z}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
  Compute ACCEPT/REJECT using Algorithm 2 or 3 with state $\mathbf{x}_t$ and proposal $\widetilde{\mathbf{x}}_t^{(i+1)}$
  Set $\mathbf{x}_t^{(i+1)} = \widetilde{\mathbf{x}}_t^{(i+1)}$ if ACCEPT; otherwise $\mathbf{x}_t^{(i+1)} = \mathbf{x}_t^{(i)}$.
end for
return $\mathbf{x}_t^{(K)}$.

where (6) is a single ULA step but we can use multiple steps.

In summary, the Langevin corrector is approximated by ULA and so it does not yield exact samples from p_{1-t}. While it may reduce errors introduced by the predictor, it introduces its own discretisation bias. In [Song et al., 2021], the step size is chosen adaptively, scaling inversely with the squared norm of the score, which in the context of ULA introduces additional bias. In the MCMC literature, explicit mechanisms exist to correct such bias and ensure exactness which we describe in the next section.

2.4 Markov chain Monte Carlo

We are interested in sampling from a target density p on $\mathbb{R}^d$, known up to a normalising constant. We describe here a generic MCMC algorithm which can be used to correct the bias of ULA. It generates a Markov chain $(\mathbf{X}^{(i)})_{i \geq 1}$ whose equilibrium distribution is p.

Given the current state $\mathbf{X}^{(i)} = \mathbf{x}$ of the Markov chain, we sample a candidate state $\widetilde{\mathbf{x}}$ from a proposal density $q(\widetilde{\mathbf{x}} \mid \mathbf{x})$. This candidate is accepted with probability $\alpha_{\mathrm{MH}}(\mathbf{x}, \widetilde{\mathbf{x}})$ for the MH algorithm and $\alpha_{\mathrm{B}}(\mathbf{x}, \widetilde{\mathbf{x}})$ for the Barker algorithm, where

$$\alpha_{\mathrm{MH}}(\mathbf{x}, \widetilde{\mathbf{x}}) = 1 \wedge R(\mathbf{x}, \widetilde{\mathbf{x}}), \qquad \alpha_{\mathrm{B}}(\mathbf{x}, \widetilde{\mathbf{x}}) = R(\mathbf{x}, \widetilde{\mathbf{x}}) \left(1 + R(\mathbf{x}, \widetilde{\mathbf{x}})\right)^{-1}, \tag{7}$$

with

$$R(\mathbf{x}, \widetilde{\mathbf{x}}) = H(\mathbf{x}, \widetilde{\mathbf{x}}) r_p(\mathbf{x}, \widetilde{\mathbf{x}}), \qquad H(\mathbf{x}, \widetilde{\mathbf{x}}) = \frac{q(\mathbf{x} \mid \widetilde{\mathbf{x}})}{q(\widetilde{\mathbf{x}} \mid \mathbf{x})}, \qquad r_p(\mathbf{x}, \widetilde{\mathbf{x}}) = \frac{p(\widetilde{\mathbf{x}})}{p(\mathbf{x})}. \tag{8}$$

If the proposal is accepted then we set $\mathbf{X}^{(i+1)} = \widetilde{\mathbf{x}}$, otherwise $\mathbf{X}^{(i+1)} = \mathbf{X}^{(i)}$. The popularity of the MH acceptance function stems from the result of Peskun [1973], which shows that, among reversible accept–reject rules, it minimises the asymptotic variance of ergodic averages. However, Łatuszyński and Roberts [2013] demonstrated that Barker's acceptance function is not far from optimal: in the worst case, it increases the asymptotic variance of Monte Carlo averages by at most a factor of two compared with MH. Although MCMC algorithms based on Barker's acceptance function are no longer strictly speaking *Metropolis–Hastings* methods, we retain the term *Metropolis-adjusted* (instead of *Barker-adjusted*) in this paper for simplicity.

Such MCMC techniques can be used to correct for the bias introduced by the ULA algorithm 2. Consider $g = 1$ for simplicity, then ULA can be thought of as defining a proposal $q(\widetilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}\big(\widetilde{\mathbf{x}}; \mathbf{x} + \frac{h}{2} \nabla \log p(\mathbf{x}), h\mathbf{I}\big)$. We can thus use this proposal within Metropolis–Hastings to sample (asymptotically in the number of iterations) exactly from p. This is a standard approach known as Metropolis-Adjusted Langevin Algorithm (MALA) [Besag, 1994, Roberts and Tweedie, 1996].

In the context of diffusion models, while the proposal-ratio term H_t is computable from score evaluations, both the MH and Barker's acceptance probability rely on a density ratio $p(\widetilde{\mathbf{x}})/p(\mathbf{x})$ which is not accessible as p is the intractable diffused data distribution at diffusion time t; i.e., $p = p_t$ in the notation of Section 2.2. So a direct implementation of MALA is not available from score evaluations alone. In Section 3, we show how we can bypass this key issue.

3 Metropolis-Adjusted Diffusion Models

In this section, we introduce the Metropolis-Adjusted samplers, intended as correctors for diffusion models, that avoid the problem of non-vanishing bias. The challenge with designing these arises from the fact that the target distribution p_t at time t is unknown and only available through its score

$s(t, \mathbf{x}_t) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. In Section 3.1, we re-express this density ratio using a line integral which we then use in Sections 3.2 and 3.3 to show how one can obtain an unbiased estimate of r_{p_t}. We then leverage this to obtain an MCMC algorithm admitting exactly p_t as invariant distribution via a two-coin Bernoulli factory [Gonçalves et al., 2017]. Section 3.4 provides an alternative lightweight estimate of the integral by a single application of Simpson's rule on the integrand. A comparison of both procedures is provided in Figure 2.

3.1 A key density ratio identity

Consider a ULA proposal to sample from p_t; i.e., $q(\widetilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}\big(\widetilde{\mathbf{x}}; \mathbf{x} + \frac{h}{2} s(\mathbf{x}, t), h\mathbf{I}\big)$. Then, as shown in Section 2.4, Metropolis-Adjusted schemes require evaluations of the ratio of proposals,

$$\log H_t(\mathbf{x}, \widetilde{\mathbf{x}}) = \log\left(\frac{q(\mathbf{x} \mid \widetilde{\mathbf{x}})}{q(\widetilde{\mathbf{x}} \mid \mathbf{x})}\right) = \frac{1}{2h}\big\|\widetilde{\mathbf{x}} - \mathbf{x} - \tfrac{h}{2}\, s(\mathbf{x}, t)\big\|^2 - \frac{1}{2h}\big\|\mathbf{x} - \widetilde{\mathbf{x}} - \tfrac{h}{2}\, s(\widetilde{\mathbf{x}}, t)\big\|^2,$$

as well as the target density ratio $r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}}) = p_t(\widetilde{\mathbf{x}})/p_t(\mathbf{x})$. While we can easily compute $H_t(\mathbf{x}, \widetilde{\mathbf{x}})$, r_{p_t} can not be obtained as directly. Here, we present two methods to estimate r_{p_t} and ultimately perform the Metropolis-Adjusted Diffusion corrector in Algorithm 1, when only the score $s(\cdot, t)$ is available. Both methods are based on the identity (see Appendix A for a full derivation):

$$\log(r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})) = \int_0^1 \big\langle s\big(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x}), t\big), \widetilde{\mathbf{x}} - \mathbf{x}\big\rangle \, \mathrm{d}u, \tag{9}$$

which reduces the problem of estimating $r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})$ to estimating a one-dimensional integral.

3.2 An unbiased estimator of the density ratio

We now construct a bounded unbiased estimator of the intractable density ratio $r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})$. This is the key object needed by the Bernoulli factory presented in Section 3.3 which will be used to perform exact adjustment. From (9), a natural unbiased estimator of the line integral is

$$I = \langle s(\mathbf{x} + U(\widetilde{\mathbf{x}} - \mathbf{x}), t), \widetilde{\mathbf{x}} - \mathbf{x}\rangle\,, \quad U \sim \mathcal{U}([0, 1]), \tag{10}$$

since we have $\mathbb{E}[I] = \log r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})$. However, directly exponentiating I does not give an unbiased estimator of $r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})$, and furthermore it does not produce a quantity bounded in $[0, 1]$ as required of the Bernoulli factory technique. Let $\hat{I}_\Delta^{(2)} = \frac{1}{2}\langle s(\widetilde{\mathbf{x}}, t) + s(\mathbf{x}, t), \widetilde{\mathbf{x}} - \mathbf{x}\rangle$ be the trapezoidal estimate of (9). The following analysis can be easily adjusted for other deterministic integral estimates $\hat{I}_\Delta$, but for clarity we fix $\hat{I}_\Delta = \hat{I}_\Delta^{(2)}$ throughout. To ensure boundedness, we make the following assumption.

Assumption 3.1. *There exists a function $C : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ such that, for every $\mathbf{x}, \widetilde{\mathbf{x}} \in \mathbb{R}^d$,*

$$\max_{u \in [0,1]} \Big|\langle s(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x}), t), \widetilde{\mathbf{x}} - \mathbf{x}\rangle - \hat{I}_\Delta^{(2)}\Big| \le C(\mathbf{x}, \widetilde{\mathbf{x}}). \tag{11}$$

In the following lemma, we show that this assumption holds for any data distribution, with a crude choice of C that only requires the score evaluations $s(\mathbf{x}, t)$ and $s(\widetilde{\mathbf{x}}, t)$ (see proof in Appendix B).

Lemma 3.2. *Suppose that $s(\cdot, t) = \nabla \log p_t(\cdot)$. Then Assumption 3.1 holds with*

$$C(\mathbf{x}, \widetilde{\mathbf{x}}) = \tfrac{1}{2}\big\langle s(\widetilde{\mathbf{x}}, t) - s(\mathbf{x}, t), \widetilde{\mathbf{x}} - \mathbf{x}\big\rangle + \tfrac{1}{r_t^2 \sigma_t^2}\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2.$$

In the following section, we will show that the efficiency of the proposed algorithm depends on the tightness of the estimate C. In Appendix D, we show that this estimate can be made less crude using smoothness in the score function.

Under Assumption 3.1, we obtain $|I - \hat{I}_\Delta^{(2)}| \le C(\mathbf{x}, \widetilde{\mathbf{x}})$ almost surely and the truncation error of $\hat{I}_\Delta^{(2)}$

$$\mathbb{E}[I - \hat{I}_\Delta^{(2)}] = \int_0^1 \Big[\big\langle s\big(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x}), t\big), \widetilde{\mathbf{x}} - \mathbf{x}\big\rangle - \hat{I}_\Delta^{(2)}\Big] \, \mathrm{d}u.$$

With $\hat{I}_\Delta^{(2)}$ as a control variate, we obtain unbiased estimates of Equation 9 with unbiased estimates of truncation error. To estimate the exponential of this integral, we use a Poisson-truncated power-series estimator. Let $N \sim \mathrm{Poisson}(2C(\mathbf{x}, \widetilde{\mathbf{x}}))$ and $I_1, \ldots, I_N$ be independent copies of I, and define

$$W = \prod_{j=1}^{N} \left(\frac{1}{2} + \frac{I_j - \hat{I}_\Delta^{(2)}}{2C(\mathbf{x}, \widetilde{\mathbf{x}})}\right). \tag{12}$$

Algorithm 2 Exact MCMC via two coin algorithm

Input: state $\mathbf{x}$, proposal $\widetilde{\mathbf{x}}$, score function $s(\cdot, t)$, step size h, bound $C(\cdot, \cdot)$.
Set $\alpha' = (1 + H_t(\mathbf{x}, \widetilde{\mathbf{x}}) \exp(\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})))^{-1}$
while True **do**
  if $U \leq \alpha'$ for $U \sim \mathcal{U}([0,1])$ **then**
    return REJECT.
  end if
  Sample $N \sim \text{Poisson}(2C(\mathbf{x}, \widetilde{\mathbf{x}}))$
  for $i = 1$ **to** N, $W \leftarrow 1$ **do**
    Sample $U \sim \mathcal{U}([0,1])$.
    $I \leftarrow \langle s(\mathbf{x} + U(\widetilde{\mathbf{x}} - \mathbf{x}), t), \widetilde{\mathbf{x}} - \mathbf{x} \rangle$.
    $W \leftarrow W \cdot \left(\frac{1}{2} + \frac{I - \hat{I}_\Delta^{(2)}}{2C(\mathbf{x}, \widetilde{\mathbf{x}})} \right)$.
  end for
  if $U \leq W$ for $U \sim \mathcal{U}([0,1])$ **then**
    return ACCEPT.
  end if
end while

Algorithm 3 MH estimate by Simpson's 1/3 rule

Input: state $\mathbf{x}$, proposal $\widetilde{\mathbf{x}}$, score function $s(\cdot, t)$, step size h, midpoint $\mathbf{x}_{\texttt{mid}} = \frac{1}{2}(\mathbf{x} + \widetilde{\mathbf{x}})$.
Compute $\widetilde{s} = \frac{1}{6}(s(\mathbf{x}, t) + 4s(\mathbf{x}_{\texttt{mid}}, t) + s(\widetilde{\mathbf{x}}, t))$.
Compute $\widehat{I} = \langle \widetilde{s}, \widetilde{\mathbf{x}} - \mathbf{x} \rangle$.
Compute $\widehat{\alpha} = \min\{1, \exp(\widehat{I}) H_t(\mathbf{x}, \widetilde{\mathbf{x}})\}$.
Sample $U \sim \mathcal{U}([0,1])$.
if $U \leq \widehat{\alpha}$ **then**
  return ACCEPT.
end if
return REJECT.

Estimator	Unbiased	Low Variance
Alg. 2	✓	×
Alg. 3	×	✓

Figure 2: Properties of integral estimators.

The intuition is that the random product expands the exponential series for $\exp\{\log r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}}) - \hat{I}_\Delta^{(2)} - C(\mathbf{x}, \widetilde{\mathbf{x}})\}$, while the rescaling by $C(\mathbf{x}, \widetilde{\mathbf{x}})$ keeps each factor within $[0, 1]$. In the following lemma, we show that this indeed produces an unbiased estimator of r_{p_t} (see proof in Appendix B.2).

Lemma 3.3. *Suppose that Assumption 3.1 holds, then* $W \in [0, 1]$ *and* $e^{C(\mathbf{x}, \widetilde{\mathbf{x}}) + \hat{I}_\Delta^{(2)}} \mathbb{E}[W] = r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})$.

Hence, conditional on W, if we sample $V \sim \mathcal{U}([0,1])$ and declare an accept when $V \leq W$, then the unconditional accept probability is $\mathbb{E}[W]$. Thus the construction gives access to a coin with unknown win probability $e^{-[C(\mathbf{x}, \widetilde{\mathbf{x}}) + \hat{I}_\Delta^{(2)}]} r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})$, without evaluating the density ratio itself. The next subsection uses this coin to obtain an exact Barker accept–reject algorithm.

3.3 A two-coin MCMC algorithm

We now turn the bounded estimator from Section 3.2 into an exact Barker correction. A Bernoulli factory is a randomized procedure which produces a Bernoulli random variable with a desired probability $f(w)$, while only having access to flips of a coin with unknown probability w [Keane and O'Brien, 1994, Nacu and Peres, 2005]. Here, the w-coin is generated by the random variable W in Lemma 3.3, and the desired probability is the Barker acceptance probability $\alpha_{\text{B}}(\mathbf{x}, \widetilde{\mathbf{x}})$ (see (7)). To simplify notation, we write $H = H_t(\mathbf{x}, \widetilde{\mathbf{x}})$, $C = C(\mathbf{x}, \widetilde{\mathbf{x}})$ and $r = r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})$. The two-coin construction proceeds in repeated rounds. In each round, we first reject immediately with probability,

$$\mathbb{P}(\text{reject in round}) = \alpha' = (1 + He^{\hat{I}_\Delta^{(2)} + C})^{-1}. \tag{13}$$

If immediate rejection does not occur, we accept with probability W. This gives us

$$\mathbb{P}(\text{accept in round}) = (1 - \alpha')e^{-[\hat{I}_\Delta^{(2)} + C]} r_{p_t}, \quad \mathbb{P}(\text{restart}) = (1 - \alpha')(1 - e^{-[\hat{I}_\Delta^{(2)} + C]} r_{p_t}).$$

Summing over the resulting geometric sequence of restarts gives the eventual acceptance probability $\alpha_{\text{B}}(\mathbf{x}, \widetilde{\mathbf{x}})$. Hence the algorithm simulates the exact Barker accept–reject decision without evaluating $r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})$. We state this result rigorously in the following theorem (see proof in Appendix B.3).

Theorem 3.4. *Suppose that* $s(\cdot, t) = \nabla \log p_t(\cdot)$ *for some positive differentiable density* p_t *and that Assumption 3.1 holds. Then Algorithm 2 almost surely terminates in finitely many iterations and returns an accept with probability* $\alpha_{\text{B}}(\mathbf{x}, \widetilde{\mathbf{x}})$ *(see* (7)*). Consequently, the adjusted proposal in Algorithm 1 using Algorithm 2 is reversible w.r.t.* p_t *and hence, the algorithm produces samples that converge in total variation distance to* p_t*.*

To determine the number of iterations required for Algorithm 2 to terminate, we state the following proposition which characterises its iteration complexity (see proof in Appendix B.4).

Proposition 3.5. *In the setting of Theorem 3.4, the number of iterations before Algorithm 2 terminates is distributed as a geometric distribution with probability* $(1 + Hr)/(1 + He^{\hat{I}_\Delta^{(2)}+C})$*. Furthermore, the number of score queries has expected value* $(2CHe^{\hat{I}_\Delta^{(2)}+C})/(1 + Hr)$ *and, with probability at least* $1 - \delta$*, it is at most,*

$$4C\frac{1 + He^{\hat{I}_\Delta^{(2)}+C}}{1 + Hr}\log(2/\delta) + 4C + \log(2/\delta).$$

The computational cost of Algorithm 2 is governed by the tightness of C. Since $|\log r - \hat{I}_\Delta^{(2)}| \le C$, we have $He^{\hat{I}_\Delta^{(2)}+C} \le e^{2C}(1 + Hr)$, so the expected number of score queries is at most $2Ce^{2C}$. A loose bound is therefore costly, while for small Langevin step sizes h the regime $C \ll 1$ arises naturally. For the choice of C in Lemma 3.2, C scales with $\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2 = O(hd)$, so the expected number of score queries is $O(hd)$, decaying to zero as the step size is reduced.

3.4 Newton–Cotes quadrature approximation

While Algorithm 2 gives an exact Barker correction, it requires a random number of score evaluations. A cheaper alternative is to approximate the line integral in (9) deterministically with a Newton–Cotes quadrature rule, and then plug the result into the usual Metropolis–Hastings acceptance probability. A Newton–Cotes quadrature rule with $N + 1$ equally spaced points approximates this integral using

$$\hat{I}_\Delta^{(N+1)} = \sum_{i=0}^{N} w_i \left\langle s\left(\mathbf{x} + \tfrac{i}{N}(\widetilde{\mathbf{x}} - \mathbf{x}), t\right), \widetilde{\mathbf{x}} - \mathbf{x}\right\rangle, \tag{14}$$

where $i = 0, 1, \ldots, N$ and $\sum_{i=0}^{N} w_i = 1$. This gives rise to well-known quadrature rules such as the trapezoidal rule and Simpson's 1/3 rule, where truncation errors are described by derivatives of the function $f(u) = \left\langle s\big(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x}), t\big), \widetilde{\mathbf{x}} - \mathbf{x}\right\rangle$ (i.e. the integrand of Equation (9)). For a survey of Newton-Cotes quadrature rules and their truncation errors, we refer the reader to Atkinson [1989].

Example 3.1 (Trapezoidal Rule)**.** Setting $N = 1$ and $w_0 = w_1 = \frac{1}{2}$ yields the estimate

$$\hat{I}_\Delta^{(2)} = \frac{1}{2}\langle s(\mathbf{x}, t) + s(\tilde{\mathbf{x}}, t), \tilde{\mathbf{x}} - \mathbf{x}\rangle.$$

This has truncation error $-\frac{1}{12}f''(u^*)$ for some $u^* \in [0, 1]$, and is of order $\mathcal{O}(h^{3/2})$.

Example 3.2 (Simpson's 1/3 Rule)**.** Setting $N = 2$, $w_0 = w_2 = \frac{1}{6}$ and $w_1 = \frac{4}{6}$ yields the estimate

$$\hat{I}_\Delta^{(3)} = \frac{1}{6}\langle s(\mathbf{x}, t) + 4s(\mathbf{x}_{\texttt{mid}}, t) + s(\widetilde{\mathbf{x}}, t), \widetilde{\mathbf{x}} - \mathbf{x}\rangle, \qquad \mathbf{x}_{\texttt{mid}} = \frac{1}{2}(\mathbf{x} + \widetilde{\mathbf{x}}).$$

This has truncation error $-\frac{1}{2880}f^{(4)}(u^*)$ for some $u^* \in [0, 1]$, and is of order $\mathcal{O}(h^{5/2})$.

With an estimate from Equation 14 in hand, we can use it in place of r_{p_t} in the Metropolis-Hastings acceptance probability $\hat{\alpha}_{\mathrm{MH}}(\mathbf{x}, \widetilde{\mathbf{x}}) = \min\left\{1, \exp(\hat{I}_\Delta)H_t(\mathbf{x}, \widetilde{\mathbf{x}})\right\}$. Empirically, we found Simpson's 1/3 rule effective. This yields Algorithm 3. Other Newton–Cotes quadrature rules could be used to approximate $\hat{I}$. The estimate only requires access to the score and enjoys two advantages over the exact method. The first is that it is more efficient to compute as other quantities including the weights $\{w_i\}_{i=0}^N$ are known. The second is that it provides a closed form estimate of the acceptance probability. The main disadvantage is that it is a biased estimate of Equation (9) due to truncation.

3.5 Implementation considerations

Practical termination. To bound the random evaluation cost of Algorithms 2, we run it for at most K rounds. If neither an accept nor reject decision is reached, we default to Algorithm 3. This forms a hybrid sampling algorithm (see Algorithm 4 in Appendix E.1).

Choosing h. The step size controls a familiar tradeoff between mixing speed and rejection rate. We show that in the high-dimensional limit with a Gaussian target, the optimal step size scales as $h \sim d^{-1/3}$ and yields an asymptotic acceptance rate of 0.347 (see Appendix E.2). This rate provides a practical heuristic: tune h to achieve an observed acceptance rate of roughly one-third.

4 Related work

Bernoulli Factories. For diffusion models, [Chen et al., 2026] used a Bernoulli factory to sample from $p_{t|t+\delta}(x|x') \propto p_t(x)\exp(-||x-\alpha_t x'||^2/(2\sigma_t^2))$ when p_t is only known through its score. This *couples* each predictor step with a rejection sampling step, with the latter acting as the corrector. This differs from our work as MADM corrector steps are optional and separate from predictor steps. Bernoulli factories have also been used in MCMC to sample intractable posteriors [Herbei and Berliner, 2014, Gonçalves et al., 2017, Morina et al., 2022, Vats et al., 2022, Gonçalves et al., 2023].

Soft Metropolis–Hastings Adjustments. Under the assumption that $p_t \approx p_{t+1}$, Feng et al. [2025] similarly suggested using each predictor step as a proposal, and *interpolating* between the proposal and current state with the MH acceptance probability. The latter relaxes the hard update from MH adjustments to a *soft* update. This differs from our work as we make no assumptions on consecutive marginals and apply MH adjustments to *corrector* steps (i.e., $\mathbf{x}_t^{(i+1)}$ only targets p_t).

Metropolis–Hastings Adjustments for Compositional Models. Sjöberg et al. [2026] estimated log density ratios for MH adjustments in compositional models (e.g. guided diffusion) using the trapezoidal rule over Hamiltonian Monte Carlo (HMC) curves and line segments (Equation 9). The former uses HMC proposals, instead of Langevin proposals as in our work. While both our work and theirs estimate Equation 9 with Newton-Cotes quadratures, we note their FIDs on ImageNet are not competitive and comparisons against other image baselines in Karras et al. [2022] are not provided. Separately, Du et al. [2023] uses an energy parameterisation to enable MH adjustments. These differ from our work as we extend the Predictor-Corrector framework of Song et al. [2021] with score-based MH adjustments to corrector steps for *unconditional* and class-conditional models, and show their effectiveness through consistent gains in FID over deterministic sampling baselines.

Learning Acceptance Probabilities. Aloui et al. [2025] proposed *learning* acceptance probabilities α_{MH} by training an *additional* neural network α_θ with a loss to enforce detailed balance. This differs from MADM as calculations of acceptance probabilities are either bypassed (Algorithm 2) or performed directly with suitable approximations of the density ratio (Algorithm 3); enabling the use of MH adjustments while only requiring access to a pre-trained score model.

5 Convergence guarantees under log-concavity

We now analyse the efficiency of the MADM corrector when used within a predictor–corrector sampler, in the simple setting where the data distribution is smooth and strongly log-concave and the exact score $s(\cdot,t) = \nabla \log p_t$ is available. We analyse the sampler that uses N noise levels $1 \geq t_0 > \cdots > t_N = 0$ where at each level t_k, it applies one step of the ancestral sampler of Ho et al. [2020] as the predictor, followed by n_k steps of the MADM corrector, that is, Algorithm 1 with step size h and accept–reject steps given by Algorithm 2. We write ν_N for the law of its output.

In the following result, $\widetilde{\Theta}$ and $\widetilde{\mathcal{O}}$ hide factors that are polylogarithmic in d, κ, $\mathbb{E}\|\mathbf{X}_0\|^2$.

Proposition 5.1. *Suppose that $\log p_{data}$ is twice continuously differentiable and satisfies $m\mathbf{I} \preceq -\nabla^2 \log p_{data} \preceq L\mathbf{I}$ for constants, $0 < m \leq 1 \leq L$, let $\kappa = L/m$ and let $\varepsilon, \delta \in (0, 1/2)$. Then, with $N = \widetilde{\Theta}(L\sqrt{d}\log(\varepsilon^{-1}\delta^{-1}))$ there exists noise levels such that taking,*

$$h^{-1} = \widetilde{\Theta}(L(d+\sqrt{\kappa})\log^2(\varepsilon^{-1}\delta^{-1})), \qquad n_k = \widetilde{\Theta}\big(\kappa(d+\sqrt{\kappa})\log^3(\varepsilon^{-1}\delta^{-1})\big),$$

we have $\|\nu_N - p_{data}\|_{\text{TV}} \leq \varepsilon$ and, with probability at least $1-\delta$, the number of score queries is at most $\widetilde{\mathcal{O}}(L\kappa\sqrt{d}(d+\sqrt{\kappa})\log^5(\varepsilon^{-1}\delta^{-1}))$.

This follows from Propositions F.7 and F.9 applied with accuracy $\varepsilon \wedge \delta/(2N)$; see Appendix F.2 for the choice of noise levels and explicit constants. The key feature of Proposition 5.1 is its polylogarithmic dependence on $1/\varepsilon$. In comparison, diffusion samplers built from predictor-only steps require $\widetilde{\mathcal{O}}(d/\varepsilon^2)$ score queries [Benton et al., 2024] or $\widetilde{\mathcal{O}}(L^2\sqrt{d}/\varepsilon)$ using unadjusted corrections [Chen et al., 2023a]. The polynomial dependence in these bounds is a result of the discretisation bias in Euler-type SDE discretisations. In contrast, each MADM corrector step is exactly reversible with respect to p_{t_k}, so ε dependence enters only through the mixing time of the corrector. The same separation can be seen between ULA and MALA in the sampling literature [Durmus and Moulines, 2017, Wu et al., 2022]. The worse dependence of our bound on d is also fundamental and is a result of the

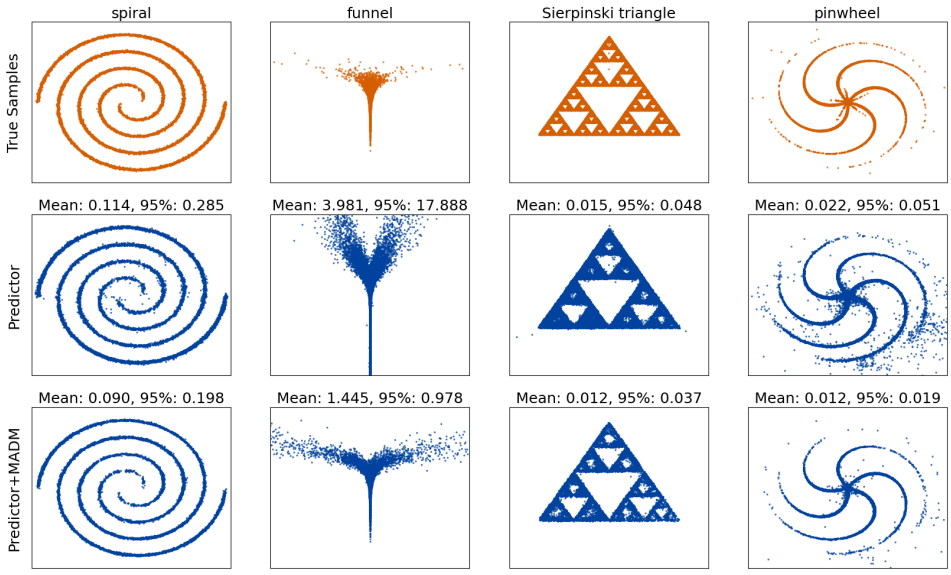

Figure 3: DDM sampling using an ancestral sampler predictor, and MADM correction. Mean distance to the true samples and the 95% sample containment distance are shown.

additional complexity of the Bernoulli factory estimator and the warm start requirement for MALA. Chen et al. [2026] also recently obtained polylogarithmic guarantees by instead removing the bias of each predictor step with rejection sampling, obtaining a query complexity of $\widetilde{\mathcal{O}}(\sqrt{d}\log^3(\varepsilon^{-1}))$.

6 Experiments

6.1 Protocol

Our proposed MADM corrector samplers only require access to a score function. This makes them readily applicable to any pre-trained diffusion model. The aim of our experiments was to illustrate the inherent gap in sample quality in existing PC samplers, and to demonstrate the effectiveness of MADM corrections to remedy this. We considered two types of datasets for our experiments: 2D datasets and image datasets. On 2D datasets, we showed that the ancestral sampler generated samples that are off the support of underlying data distributions, and that MADM corrector steps move these outliers back to the support. On image datasets, we demonstrated numerical improvements in FID by interweaving probability flow ODE steps and MADM corrector steps.

6.2 Synthetic 2D Datasets

We considered the ancestral sampler as the predictor on four 2D datasets: spiral, funnel, Sierpinski triangle, and pinwheel. It is known that the ancestral sampler is a specific discretisation of Equation 4 (see Appendix E of Song et al. [2021]). In other words, we can view one step of the ancestral sampler as equivalent to: one probability flow ODE predictor step and one *unadjusted* Langevin corrector step (i.e. ULA) with a shared step size. Our results with these settings are shown in Figure 3. The second row in Figure 3 illustrates that the ULA step embedded within each ancestral sampler step is not sufficient to move generated samples to the support of underlying data distributions, leaving a large number of outliers. The third row in Figure 3 shows that MADM corrector steps allow us to move samples to the support of underlying data distributions, leaving few to no outliers. This illustrates an improvement in sample quality by adding accept-reject adjustments within the sampling process.

6.3 Image Datasets

We considered the probability flow ODE as the predictor on four standard image datasets: CIFAR-10 [Krizhevsky and Hinton, 2009], FFHQ [Karras et al., 2019], AFHQv2 [Choi et al., 2020] and

Table 1: Fréchet Inception Distance (FID) on image datasets sampled using pre-trained models from Karras et al. [2022]. We consider the probability flow ODE (PF-ODE) predictor equipped with unadjusted Langevin algorithm (ULA) and Metropolis-Adjusted Diffusion Model (MADM) correctors. All FIDs are calculated using 50,000 samples. We highlight the best FID in bold.

ODE Solver	PC Sampler	CIFAR-10	FFHQ	AFHQv2	ImageNet
Euler	PF-ODE	7.62	4.59	2.93	2.41
	PF-ODE + ULA	7.00	4.99	2.85	3.83
	PF-ODE + MADM (Alg. 4)	7.53	4.58	2.93	**2.21**
	PF-ODE + MADM (Alg. 3)	7.52	4.52	2.92	2.25
	PF-ODE (NFE-matched)	**2.69**	**2.78**	**2.19**	2.34
Heun	PF-ODE	1.96	2.47	2.04	2.32
	PF-ODE + ULA	1.97	2.94	2.11	3.64
	PF-ODE + MADM (Alg. 4)	1.96	2.45	2.04	2.16
	PF-ODE + MADM (Alg. 3)	**1.95**	**2.43**	**2.03**	**2.15**
	PF-ODE (NFE-matched)	2.01	2.45	2.05	2.32

ImageNet-64 [Deng et al., 2009]. We used the pre-trained models and default sampling strategy from Karras et al. [2022] and evaluated sample quality using FID. We compared against Heun and Euler ODE solvers. The former served as the natural baseline that balances efficiency and sample quality, while the latter served as a diagnostic to verify that corrector steps improve sample quality when an appreciable error gap exists. Finally, we followed the experimental setup for predictor-corrector (PC) samplers in Song et al. [2021], by using *one* corrector step per predictor step. We present numerical comparisons for ULA, MADM using Algorithms 3 and 4, and the probability flow ODE with predictor steps matched to the number of function evaluations (NFE) as the PC sampler equipped with Algorithm 3 as its MADM corrector. Our FID results with these settings are shown in Table 1.

Unconditional models on CIFAR-10, FFHQ and AFHQv2. In the Euler regime, additional predictor or MADM corrector steps both improve FID, though predictor steps are more beneficial as predictor error dominates in this setting. In the Heun regime, additional predictor steps can deteriorate FID, but one MADM corrector step yielded FIDs competitive to those in Karras et al. [2022], Hatamizadeh et al. [2025] where gains were consistent, albeit sometimes modest. This suggests that MADM improves or matches the sample quality from deterministic sampling.

Class-conditional model on ImageNet-64. Applying one MADM corrector step per Euler predictor step yielded larger gains in FID than using only Heun predictor steps without corrector steps. This suggests that it is more beneficial to use accept-reject adjustments, rather than a higher-order ODE solver, to mitigate discretisation errors when the number of discretisation steps is large.

7 Conclusion

We have introduced MADM as a Metropolis-adjusted corrector framework that naturally extends the Predictor-Corrector framework of diffusion models, to mitigate discretisation bias. We presented the first exact method to compute Barker-adjusted Langevin corrections for diffusion models, based on a two-coin Bernoulli factory algorithm. We also presented an efficient approximation to compute Metropolis-adjusted Langevin corrections, based on Simpson's 1/3 rule. We then drew the connection between the two approaches with the trapezoidal approximation serving as a control variate for the exact method. Empirically, we demonstrated effectiveness of MADM for improving sample quality on both synthetic and image datasets. In the latter, MADM corrector steps were more beneficial when the number of predictor steps is large, yielding competitive FIDs with gains that were consistent, albeit sometimes modest. Regarding limitations, our work assumes perfect score estimation and that the exact method can be computationally expensive due to the random number of score evaluations required. We note that the former is not unique to our work and is an inherent limitation of diffusion models, and the latter can be substantially mitigated with the control variate formulation and hybrid sampling approach. Future work can focus on developing efficient subroutines for our exact method.

Acknowledgements

We acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR) [McIntosh-Smith et al., 2024]. Isambard-AI is operated by the University of Bristol and is funded by the UK Government's Department for Science, Innovation and Technology (DSIT) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023]. Kevin H. Lam gratefully acknowledges his PhD funding from Google DeepMind. Tyler Farghly is supported by the European Research Council Starting Grant DYNASTY - 101039676. Christopher Williams acknowledges support from: the Defence Science and Technology (DST) Group Australia; and supported in part, by the Engineering and Physical Sciences Research Council (EPSRC) through the AI Hub in Generative Models [grant number EP/Y028805/1]. Jun Yang acknowledges support from the Independent Research Fund Denmark (DFF) through the Sapere Aude Starting Grant (5251-00032B).

References

Sanket Agrawal, Dootika Vats, Krzysztof Łatuszyński, and Gareth O Roberts. Optimal scaling of MCMC beyond Metropolis. *Advances in Applied Probability*, 55(2):492–509, 2023.

Ahmed Aloui, Ali Hasan, Juncheng Dong, Zihao Wu, and Vahid Tarokh. Score-based Metropolis-Hastings algorithms. *arXiv preprint arXiv:2501.00467*, 2025.

Kendall E. Atkinson. *An Introduction to Numerical Analysis*. Wiley, New York ;, 2nd ed. edition, 1989.

Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and Geometry of Markov Diffusion Operators*. Springer International Publishing, 2014.

Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly d-linear convergence bounds for diffusion models via stochastic localization. In *The Twelfth International Conference on Learning Representations*, 2024.

Julian Besag. Comments on "Representations of knowledge in complex systems" by U. Grenander and M. I. Miller. *Journal of the Royal Statistical Society Series B*, 56(591-592):4, 1994.

Sergey G Bobkov and Bruno Volzone. On gilles pisier's approach to gaussian concentration, isoperimetry, and poincaré-type inequalities. *Electronic Journal of Probability*, 29(none):1–27, 2024.

Fan Chen, Sinho Chewi, Constantinos Daskalakis, and Alexander Rakhlin. High-accuracy sampling for diffusion models and log-concave distributions. *arXiv preprint arXiv:2602.01338*, 2026.

Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ODE is provably fast. In *Advances in Neural Information Processing Systems*, 2023a.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R. Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023b.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 8489–8510. PMLR, 23–29 Jul 2023.

Alain Durmus and Eric Moulines. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *Annals of Applied Probability*, 27(3):1551–1587, 2017.

Hanqi Feng, Peng Qiu, Meng-Chun Zhang, You Fan, Yiran Tao, and Barnabas Poczos. Soft Metropolis-Hastings correction for generative model sampling. *bioRxiv*, 2025.

Xuefeng Gao, Hoang M Nguyen, and Lingjiong Zhu. Wasserstein convergence guarantees for a general class of score-based generative models. *Journal of Machine Learning Research*, 26(43): 1–54, 2025.

Flávio B Gonçalves, Krzysztof Łatuszyński, and Gareth O Roberts. Exact Monte Carlo likelihood-based inference for jump-diffusion processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(3):732–756, 2023.

Flávio B Gonçalves, Krzysztof Łatuszyński, and Gareth O Roberts. Barker's algorithm for Bayesian inference with intractable likelihoods. *Brazilian Journal of Probability and Statistics*, 31(4): 732–745, 2017.

Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. In *Computer Vision – ECCV 2024*, pages 37–55, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73242-3.

Radu Herbei and L Mark Berliner. Estimating ocean circulation: an MCMC approach with approximated likelihoods via the Bernoulli factory. *Journal of the American Statistical Association*, 109 (507):944–954, 2014.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

Sascha Holl, Gurprit Singh, and Hans-Peter Seidel. Diffusion restore: Real-time markov chain monte carlo light transport. *arXiv preprint arXiv:2605.08916*, 2026. doi: 10.48550/arXiv.2605.08916.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, 2022.

MS Keane and George L O'Brien. A Bernoulli factory. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 4(2):213–219, 1994.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009.

Krzysztof Łatuszyński and Gareth O Roberts. CLTs and asymptotic variance of time-sampled Markov chains. *Methodology and Computing in Applied Probability*, 15(1):237–247, 2013.

L Lovász and M Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures & Algorithms*, 4(4):359–412, 1993.

Simon McIntosh-Smith, Sadaf R. Alam, and Christopher J. Woods. Isambard-ai: a leadership-class supercomputer optimised specifically for artificial intelligence. In *Proceedings of the Cray User Group, CUG 2024, Perth, WA, Australia, May 5-9, 2024*, pages 44–54. ACM, 2024.

Giulio Morina, Krzysztof Łatuszyński, Piotr Nayar, and Alex Wendland. From the Bernoulli factory to a dice enterprise via perfect sampling of Markov chains. *The Annals of Applied Probability*, 32 (1):327–359, 2022.

Alireza Mousavi-Hosseini, Tyler K Farghly, Ye He, Krishna Balasubramanian, and Murat A Erdogdu. Towards a complete analysis of langevin monte carlo: Beyond poincaré inequality. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 1–35. PMLR, 2023.

Şerban Nacu and Yuval Peres. Fast simulation of new coins from old. *Annals of Applied Probability*, pages 93–115, 2005.

Peter H Peskun. Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612, 1973.

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.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various Metropolis–Hastings algorithms. *Statistical Science*, 16(4):351 – 367, 2001.

Gareth O. Roberts and Jeffrey S. Rosenthal. Harris recurrence of Metropolis-within-Gibbs and trans-dimensional Markov chains. *The Annals of Applied Probability*, 16(4):2123–2139, 2006.

Gareth O Roberts and Richard L Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(3):341–363, 1996.

Gareth O. Roberts, Andrew Gelman, and Wally R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.

Anders Sjöberg, Jakob Lindqvist, Magnus Önnheim, Mats Jirstrand, and Lennart Svensson. MCMC correction of score-based diffusion models for model composition. *Entropy*, 28(3), 2026.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Dootika Vats, Flávio B Gonçalves, K Łatuszyński, and Gareth O Roberts. Efficient Bernoulli factory Markov chain Monte Carlo for intractable posteriors. *Biometrika*, 109(2):369–385, 2022.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

Christopher Williams, Andrew Campbell, Arnaud Doucet, and Saifuddin Syed. Score-optimal diffusion schedules. In *Advances in Neural Information Processing Systems*, 2024.

Keru Wu, Scott Schmidler, and Yuansi Chen. Minimax mixing time of the metropolis-adjusted langevin algorithm for log-concave sampling. *Journal of Machine Learning Research*, 23(270): 1–63, 2022.

Jun Yang, Gareth O. Roberts, and Jeffrey S. Rosenthal. Optimal scaling of random-walk Metropolis algorithms on general target distributions. *Stochastic Processes and their Applications*, 130(10): 6094–6132, 2020.

A Derivation of density ratio identity

Recall that $r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}}) = \frac{p_t(\widetilde{\mathbf{x}})}{p_t(\mathbf{x})}$, so it follows that $\log r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}}) = \log p_t(\widetilde{\mathbf{x}}) - \log p_t(\mathbf{x})$. Note that p is differentiable. By the fundamental theorem of calculus, we can write

$$\begin{aligned}\log p_t(\widetilde{\mathbf{x}}) - \log p_t(\mathbf{x}) &= \int_0^1 \big\langle \nabla \log p_t\big(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x})\big), \widetilde{\mathbf{x}} - \mathbf{x} \big\rangle \, \mathrm{d}u \\ &= \int_0^1 \big\langle s\big(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x}), t\big), \widetilde{\mathbf{x}} - \mathbf{x} \big\rangle \, \mathrm{d}u,\end{aligned}$$

where the second equality follows from setting $s(\cdot, t) = \nabla \log p_t$. This completes our derivation of Equation (9).

B Proofs for the exact estimator

For the sake of brevity, we use the shorthand $H = H_t(\mathbf{x}, \tilde{\mathbf{x}})$, $C = C(\mathbf{x}, \tilde{\mathbf{x}})$, $r = r_{p_t}(\mathbf{x}, \tilde{\mathbf{x}})$.

B.1 Proof of Lemma 3.2

As in Section 3.4, let $f(u) = \langle s(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x}), t), \widetilde{\mathbf{x}} - \mathbf{x} \rangle$ denote the integrand of (9), so that $\hat{I}_\Delta^{(2)} = \frac{1}{2}(f(0) + f(1))$.

Recall that p_t is the law of $X_t = r_t X_0 + r_t \sigma_t \boldsymbol{\epsilon}$, where $X_0 \sim p_{\text{data}}$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are independent. Our proof relies on the denoiser $d(\mathbf{y}, t) := \mathbb{E}[X_0 | X_t = \mathbf{y}]$ which is related to the score through Tweedie's identity,

$$s(\mathbf{y}, t) = \frac{r_t \, d(\mathbf{y}, t) - \mathbf{y}}{r_t^2 \sigma_t^2},$$

and its Jacobian is proportional to the posterior covariance,

$$\nabla_{\mathbf{y}} d(\mathbf{y}, t) = \tfrac{r_t}{r_t^2 \sigma_t^2} \mathrm{Cov}(X_0 | X_t = \mathbf{y}) \succeq \mathbf{0}.$$

Using Tweedie's identity, we can write

$$f(u) - \hat{I}_\Delta^{(2)} = \phi(u) - \phi_{\texttt{mid}} - \frac{u - 1/2}{r_t^2 \sigma_t^2} \|\widetilde{\mathbf{x}} - \mathbf{x}\|^2,$$

where $\phi(u) = \frac{r_t}{r_t^2 \sigma_t^2} \langle d(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x}), t), \widetilde{\mathbf{x}} - \mathbf{x} \rangle$ and $\phi_{\texttt{mid}} = \frac{1}{2}(\phi(0) + \phi(1))$. Since the Jacobian of the denoiser is positive semi-definite, we have $\phi'(u) \geq 0$, and so $\phi(0) \leq \phi(u) \leq \phi(1)$. It follows that

$$\begin{aligned}|\phi(u) - \phi_{\texttt{mid}}| &\leq \tfrac{1}{2}(\phi(1) - \phi(0)) \\ &= \tfrac{r_t}{2 r_t^2 \sigma_t^2} \big\langle d(\widetilde{\mathbf{x}}, t) - d(\mathbf{x}, t), \widetilde{\mathbf{x}} - \mathbf{x} \big\rangle \\ &= \tfrac{1}{2} \langle s(\widetilde{\mathbf{x}}, t) - s(\mathbf{x}, t), \widetilde{\mathbf{x}} - \mathbf{x} \rangle + \tfrac{1}{2 r_t^2 \sigma_t^2} \|\widetilde{\mathbf{x}} - \mathbf{x}\|^2,\end{aligned}$$

where the last line uses Tweedie's identity once more. Finally, since $|u - 1/2| \leq 1/2$, the triangle inequality gives

$$\begin{aligned}|f(u) - \hat{I}_\Delta^{(2)}| &\leq |\phi(u) - \phi_{\texttt{mid}}| + \tfrac{1}{2 r_t^2 \sigma_t^2} \|\widetilde{\mathbf{x}} - \mathbf{x}\|^2 \\ &\leq \tfrac{1}{2} \langle s(\widetilde{\mathbf{x}}, t) - s(\mathbf{x}, t), \widetilde{\mathbf{x}} - \mathbf{x} \rangle + \tfrac{1}{r_t^2 \sigma_t^2} \|\widetilde{\mathbf{x}} - \mathbf{x}\|^2.\end{aligned}$$

Since this holds for every $u \in [0, 1]$, Assumption 3.1 holds with the stated choice of C. □

B.2 Proof of Lemma 3.3

Recall that $|I_i - \hat{I}_\Delta^{(2)}| \leq C$ holds almost surely, for all i, and hence,

$$\frac{1}{2} + \frac{I_i - \hat{I}_\Delta^{(2)}}{2C} \in [0, 1].$$

Since W is formed of a product of these variables, we also obtain $W \in [0, 1]$ almost surely.

Since $\mathbb{E}[I_i - \hat{I}_\Delta^{(2)}] = \log r - \hat{I}_\Delta^{(2)}$, and $\{I_i - \hat{I}_\Delta^{(2)}\}_{i=1}^N$ are i.i.d. and also independent of N, we obtain,

$$\begin{aligned}\mathbb{E}[W|N] &= \prod_{i=1}^{N} \mathbb{E}\left[\frac{1}{2} + \frac{I_i - \hat{I}_\Delta^{(2)}}{2C}\right] \\ &= \left(\frac{1}{2} + \frac{\log r - \hat{I}_\Delta^{(2)}}{2C}\right)^N,\end{aligned}$$

where we use the convention $0^0 = 1$. Recalling that $N \sim \text{Poisson}(2C)$, the law of total expectation gives,

$$\begin{aligned}\mathbb{E}[W] &= \sum_{n=0}^{\infty} \frac{(2C)^n \exp(-2C)}{n!} \mathbb{E}[W|N = n] \\ &= \exp(-2C) \sum_{n=0}^{\infty} \frac{1}{n!} \big(C + \log r - \hat{I}_\Delta^{(2)}\big)^n.\end{aligned}$$

Finally, recognising the sum as the Taylor series of the exponential, we obtain,

$$\begin{aligned}\mathbb{E}[W] &= \exp(-2C) \exp(C + \log r - \hat{I}_\Delta^{(2)}) \\ &= \exp(-C) \exp(\log r - \hat{I}_\Delta^{(2)}) \\ &= \exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})]) r_{p_t}(\mathbf{x}, \widetilde{\mathbf{x}})\end{aligned}$$

completing the proof. □

B.3 Proof of Theorem 3.4

In each outer iteration of Algorithm 2, one of three events occurs. The iteration could produce an immediate rejection, which results from a coin flip with probability α'. The iteration could also produce an acceptance, which requires first avoiding rejection and then winning the following acceptance coin flip with probability W. Since the immediate rejection flip and the acceptance flip are independent, the tower property gives this probability as

$$(1 - \alpha')\mathbb{E}[W] = (1 - \alpha') \exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})]) r.$$

Finally, it could occur that neither coin wins and neither a rejection or acceptance is returned, resulting in the loop restarting. This occurs with the remaining probability,

$$1 - \alpha' - (1 - \alpha') \exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})]) r = (1 - \alpha')(1 - \exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})]) r).$$

We begin by briefly analysing the number of iterations (of the outer loop) before termination, which we denote by K. Since K being at least as large as n is equivalent to the event that the first $n - 1$ iterations produced restarts, it follows that,

$$\mathbb{P}(K \geq n) = (1 - \alpha')^{n-1} (1 - \exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})]) r)^{n-1}.$$

Since $H > 0$, we have $\alpha' < 1$ and so the restart probability satisfies

$$(1 - \alpha')(1 - \exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})]) r) \in [0, 1).$$

Hence, we have $\lim_{n\to\infty} \mathbb{P}(K \geq n) = 0$. To show that K is almost-surely finite, we note that for any n,

$$\mathbb{P}(K < \infty) \geq \mathbb{P}(K < n) = 1 - \mathbb{P}(K \geq n).$$

In particular, we have that

$$\mathbb{P}(K < \infty) \geq 1 - \lim_{n\to\infty} \mathbb{P}(K \geq n) = 1.$$

We now compute the probability of acceptance. Let A_n denote the events that the algorithm accepts on the n-th outer iteration. Since A_n necessarily requires a sequence of $n-1$ no decision iterations followed by an acceptance, it follows directly that,

$$\Pr(A_n) = (1-\alpha')^{n-1}(1-\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r)^{n-1}(1-\alpha')\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r.$$

To compute the probability that the algorithm terminates with an acceptance, we must compute the probability of the union $\cup_{n=1}^{\infty} A_n$. Since the events $\{A_n\}_{n=1}^{\infty}$ are disjoint, we obtain the acceptance probability as a geometric series:

$$\begin{aligned}\Pr\left(\bigcup_{n=1}^{\infty} A_n\right) &= \sum_{n=1}^{\infty} \Pr(A_n) \\ &= (1-\alpha')\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r \sum_{n=1}^{\infty}(1-\alpha')^{n-1}(1-\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r)^{n-1}.\end{aligned}$$

Since the sum is a geometric series with rate $(1-\alpha')(1-\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r) \in [0,1)$, it follows that the sum converges with,

$$\begin{aligned}\Pr\left(\bigcup_{n=1}^{\infty} A_n\right) &= (1-\alpha')\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r \frac{1}{1-(1-\alpha')(1-\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r)} \\ &= \frac{(1-\alpha')\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r}{\alpha' + (1-\alpha')\exp(-[\hat{I}_\Delta^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})])r}.\end{aligned}$$

Recalling that $\alpha' = (1 + He^{\hat{I}_\Delta^{(2)}+C})^{-1}$, this gives,

$$\begin{aligned}\Pr\left(\bigcup_{n=1}^{\infty} A_n\right) &= \frac{He^{\hat{I}_\Delta^{(2)}+C}\exp(-[\hat{I}_\Delta^{(2)} + C])r}{1 + He^{\hat{I}_\Delta^{(2)}+C}\exp(-[\hat{I}_\Delta^{(2)} + C])r} \\ &= \frac{Hr}{1+Hr} \\ &= \alpha_B(\mathbf{x}, \tilde{\mathbf{x}}).\end{aligned}$$

Reversibility with respect to p_t follows since the Barker acceptance function $a_B(r) = r/(1+r)$ satisfies $a_B(r) = r a_B(1/r)$ for all $r > 0$, which is a sufficient condition for detailed balance [see, e.g., Agrawal et al., 2023].

For convergence, we use that p_t is a positive density and that the Langevin proposal has full support on $\mathbb{R}^d$, which guarantees $\alpha_B(\mathbf{x}, \tilde{\mathbf{x}}) > 0$ everywhere and hence that the chain is ϕ-irreducible with respect to Lebesgue measure. Then Algorithm 1 is Harris recurrent, which follows from the same line of argument as in Roberts and Rosenthal [2006]. Since p_t is an invariant probability measure, the chain is positive recurrent. Convergence in total variation distance to p_t follows.

□

B.4 Proof of Proposition 3.5

From the proof of Theorem 3.4, the probability of termination in a single outer iteration is,

$$\alpha' + (1-\alpha')e^{-[\hat{I}_\Delta^{(2)}+C]}r = \frac{1}{1+He^{\hat{I}_\Delta^{(2)}+C}} + \frac{Hr}{1+He^{\hat{I}_\Delta^{(2)}+C}} = \frac{1+Hr}{1+He^{\hat{I}_\Delta^{(2)}+C}},$$

and since iterations are independent, the number of iterations K before termination is geometric with this probability, establishing the first claim.

For the second claim, we reframe the setting as an optional stopping problem. Let N_i denote the number of score queries at iteration i where we set $N_i = 0$ if a rejection is returned at that iteration. With this, we obtain that the total number of score queries is given by $\mathbb{E}\left[\sum_{i=1}^{K} N_i\right]$.

For $i > K$, we continue to generate N_i as if the algorithm has not terminated so that $(N_i)_{i=1}^{\infty}$ is an i.i.d. sequence. Let $\mathcal{F}_n$ denote the sigma-algebra generated by all randomness in iterations $1, \ldots, n$.

From this, we see that $(N_i)_{i=1}^{\infty}$ is adapted to the filtration $(\mathcal{F}_i)_{i=1}^{\infty}$ and K is a stopping time. With this, we describe the total number of queries as an optionally stopped sum over an i.i.d. process, and hence we can apply Wald's lemma. Provided $\mathbb{E}[N_1], \mathbb{E}[K] < \infty$, this lemma gives,

$$\mathbb{E}\Big[\sum_{i=1}^{K} N_i\Big] = \mathbb{E}[N_1]\mathbb{E}[K].$$

Since N_1 takes the value 0 with probability α' and is $\mathrm{Poisson}(2C)$ distributed with probability $1-\alpha'$, we obtain the expectation

$$\mathbb{E}[N_1] = 2(1-\alpha')C = \frac{2CHe^{\hat{I}_\Delta^{(2)}+C}}{1+He^{\hat{I}_\Delta^{(2)}+C}}.$$

Since K is geometrically distributed with parameter $(1+Hr)/(1+He^{\hat{I}_\Delta^{(2)}+C})$, we also obtain,

$$\mathbb{E}[K] = \frac{1+He^{\hat{I}_\Delta^{(2)}+C}}{1+Hr}.$$

Thus, we obtain the expected number of queries from the product,

$$\mathbb{E}\Big[\sum_{i=1}^{K} N_i\Big] = \frac{2CHe^{\hat{I}_\Delta^{(2)}+C}}{1+He^{\hat{I}_\Delta^{(2)}+C}} \cdot \frac{1+He^{\hat{I}_\Delta^{(2)}+C}}{1+Hr} = \frac{2CHe^{\hat{I}_\Delta^{(2)}+C}}{1+Hr}.$$

For the final claim, let $\delta \in (0,1)$ and let k be the smallest integer such that $k \geq \frac{1+He^{\hat{I}_\Delta^{(2)}+C}}{1+Hr}\log(2/\delta)$. Thus, for any $q \geq 0$, a union bound gives,

$$\mathbb{P}\Big(\sum_{i=1}^{K} N_i > q\Big) \leq \mathbb{P}(K > k) + \mathbb{P}\Big(\sum_{i=1}^{k} N_i > q\Big).$$

Note that this bound does not require any independence between K and $(N_i)_{i=1}^{\infty}$. For the first term, since K is geometrically distributed, using $1-x \leq e^{-x}$ and the choice of k, we obtain,

$$\mathbb{P}(K > k) = \left(1 - \frac{1+Hr}{1+He^{\hat{I}_\Delta^{(2)}+C}}\right)^k \leq \exp\left(-\frac{1+Hr}{1+He^{\hat{I}_\Delta^{(2)}+C}}k\right) \leq \frac{\delta}{2}.$$

For the second term, we note that N_1 is either 0 or $\mathrm{Poisson}(2C)$ distributed, so that $\mathbb{E}[e^{N_1}] \leq \exp(2C(e-1))$. Hence, setting $q = 4Ck + \log(2/\delta)$, Markov's inequality and the independence of $(N_i)_{i=1}^{\infty}$ give,

$$\begin{aligned}
\mathbb{P}\Big(\sum_{i=1}^{k} N_i > q\Big) &\leq e^{-q}\mathbb{E}[e^{N_1}]^k \\
&\leq \exp\big(2C(e-1)k - 4Ck - \log(2/\delta)\big) \\
&\leq \frac{\delta}{2}.
\end{aligned}$$

Combining these bounds and using that $k \leq \frac{1+He^{\hat{I}_\Delta^{(2)}+C}}{1+Hr}\log(2/\delta)+1$, we obtain that, with probability at least $1-\delta$,

$$\sum_{i=1}^{K} N_i \leq q \leq 4C\frac{1+He^{\hat{I}_\Delta^{(2)}+C}}{1+Hr}\log(2/\delta) + 4C + \log(2/\delta).$$

□

C Results for Newton–Cotes quadrature rules

C.1 Other common quadrature rules

Example C.1 (Simpson's 3/8 rule)**.** Setting $N = 3$, $w_0 = w_3 = \frac{1}{8}$ and $w_1 = w_2 = \frac{3}{8}$ yields the estimate

$$\hat{I}_\Delta^{(4)} = \frac{1}{8}\langle s(\mathbf{x},t) + 3s(\mathbf{y},t) + 3s(\mathbf{z},t) + s(\tilde{\mathbf{x}},t), \tilde{\mathbf{x}} - \mathbf{x}\rangle, \quad \mathbf{y} = \mathbf{x} + \frac{1}{3}(\tilde{\mathbf{x}} - \mathbf{x}), \quad \mathbf{z} = \mathbf{x} + \frac{2}{3}(\tilde{\mathbf{x}} - \mathbf{x}).$$

This has truncation error $-\frac{1}{6480}f^{(4)}(u^*)$ for some $u^* \in [0,1]$, and is of order $\mathcal{O}(h^{5/2})$.

C.2 Derivation of error bounds from truncation error

Let $\mathbf{v} = \tilde{\mathbf{x}} - \mathbf{x}$. The results in this section are based on the following lemma.

Lemma C.1. *Let* $\tilde{\mathbf{x}} = \mathbf{x} + \frac{h}{2}s(\mathbf{x}, t) + \sqrt{h}\mathbf{z}$ *where* $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. *Suppose* $s(\mathbf{x}, t)$ *is locally bounded. Then* $\|\tilde{\mathbf{x}} - \mathbf{x}\|$ *has* $\mathcal{O}(h^{1/2})$.

Proof. By triangle inequality, we have

$$\|\mathbf{v}\| = \|\tilde{\mathbf{x}} - \mathbf{x}\| \leq \frac{h}{2}\|s(\mathbf{x}, t)\| + \sqrt{h}\|\mathbf{z}\|.$$

Since $s(\mathbf{x}, t)$ is locally bounded, the first term on the RHS of the inequality has order $\mathcal{O}(h)$. The second term on the RHS of the inequality has order $\mathcal{O}(h^{1/2})$. Since $h^{1/2}$ dominates h as $h \to 0$, it follows that $\|\mathbf{v}\|$ has order $\mathcal{O}(h^{1/2})$. □

Recall from Section 3.4 that

$$f : [0, 1] \to \mathbb{R}, \quad f(u) = \big\langle s\big(\mathbf{x} + u\mathbf{v}, t\big), \mathbf{v}\big\rangle$$

where f is the integrand of Equation (9). Truncation errors from the Newton–Cotes quadrature rules introduced in Section 3.4 are expressed by derivatives of f. More specifically, they can be expressed by Fréchet derivatives of s evaluated at $\mathbf{x} + u^*\mathbf{v}$. The n-th Fréchet derivative $D^n s(\mathbf{x}) : \underbrace{\mathbb{R}^d \times \cdots \times \mathbb{R}^d}_{n \text{ times}} \to \mathbb{R}^d$ is a multi-linear map takes n arguments from $\mathbb{R}^d$ and outputs a value from $\mathbb{R}^d$.

Trapezoidal rule. Recall from Example 3.1 that the truncation error of the trapezoidal rule is $-\frac{1}{12}f''(u^*)$ for some $u^* \in [0, 1]$. This gives us

$$\begin{aligned} |f''(u^*)| &= |\langle D^2 s(\mathbf{x} + u^*\mathbf{v})[\mathbf{v}, \mathbf{v}], \mathbf{v}\rangle| \\ &\leq \|D^2 s(\mathbf{x} + u^*\mathbf{v})[\mathbf{v}, \mathbf{v}]\|\|\mathbf{v}\| \\ &\leq \|D^2 s(\mathbf{x} + u^*\mathbf{v})\|_{\text{op}}\|\mathbf{v}\|^3 \end{aligned}$$

where the first inequality follows from Cauchy-Schwarz and the second follows from the definition of operator norms for multi-linear maps. By Lemma C.1, it follows that $|f''(u^*)|$ has $\mathcal{O}(h^{3/2})$.

Simpson's 1/3 and 3/8 rules. Recall from Examples 3.2 and C.1 that the truncation error of both Simpson's 1/3 and 3/8 rules are proportional $f^{(4)}(u^*)$ for some $u^* \in [0, 1]$. Following the same arguments as for the trapezoidal rule, this gives us

$$\begin{aligned} |f^{(4)}(u^*)| &= |\langle D^4 s(\mathbf{x} + u^*\mathbf{v})[\mathbf{v}, \mathbf{v}, \mathbf{v}, \mathbf{v}], \mathbf{v}\rangle| \\ &\leq \|D^4 s(\mathbf{x} + u^*\mathbf{v})[\mathbf{v}, \mathbf{v}, \mathbf{v}, \mathbf{v}]\|\|\mathbf{v}\| \\ &\leq \|D^4 s(\mathbf{x} + u^*\mathbf{v})\|_{\text{op}}\|\mathbf{v}\|^5. \end{aligned}$$

By Lemma C.1, it follows that $|f^{(4)}(u^*)|$ has $\mathcal{O}(h^{5/2})$.

D Choosing the quantity C

The validity of Algorithm 2 rests on a valid estimate of $C(\mathbf{x}, \widetilde{\mathbf{x}})$ and, as shown in Proposition 3.5, the computational cost depends on the tightness of this quantity. Choosing a suitable $C(\mathbf{x}, \tilde{\mathbf{x}})$ requires some additional knowledge about the score function as it bounds the score function along some, typically small, line segment. Thus, we require some predictability of how the score function changes along this line segment via smoothness or boundedness of the score network.

Bounded data support. Suppose that the data distribution is known to be supported on the centred ball of radius b. By Tweedie's identity, the score admits the form,

$$\nabla \log p_t = \frac{r_t\, d(\mathbf{x}, t) - \mathbf{x}}{r_t^2 \sigma_t^2}, \qquad d(\mathbf{x}, t) := \mathbb{E}[X_0 | X_t = \mathbf{x}].$$

Thus, the score function can be controlled using the fact that $\|d(\mathbf{x},t)\| \le b$. Similarly, given a generic score function $s(\mathbf{x},t)$ that is, for example, derived from a trained neural network, it may be parameterised such that the associated denoising function $\hat{d}(\mathbf{x},t)$ is appropriately compact. Indeed, if we assume that $\|\hat{d}(\cdot,t)\| \le b$ for some $b > 0$, it follows that for any $\mathbf{x}, \widetilde{\mathbf{x}} \in \mathbb{R}^d$ and $u \in [0,1]$,

$$\begin{aligned}
|\langle s(\mathbf{x}+u(\widetilde{\mathbf{x}}-\mathbf{x}),t), \widetilde{\mathbf{x}}-\mathbf{x}\rangle| &= \frac{1}{r_t^2\sigma_t^2}|r_t\langle d(\mathbf{x}+u(\widetilde{\mathbf{x}}-\mathbf{x}),t), \widetilde{\mathbf{x}}-\mathbf{x}\rangle - \langle \mathbf{x}+u(\widetilde{\mathbf{x}}-\mathbf{x}), \widetilde{\mathbf{x}}-\mathbf{x}\rangle| \\
&\le \frac{1}{r_t\sigma_t^2}\|d(\mathbf{x}+u(\widetilde{\mathbf{x}}-\mathbf{x}),t)\|\|\widetilde{\mathbf{x}}-\mathbf{x}\| + \frac{1}{r_t^2\sigma_t^2}|\langle \mathbf{x}+u(\widetilde{\mathbf{x}}-\mathbf{x}), \widetilde{\mathbf{x}}-\mathbf{x}\rangle| \\
&\le \frac{br_t + \max\{\|\mathbf{x}\|, \|\widetilde{\mathbf{x}}\|\}}{r_t^2\sigma_t^2}\|\widetilde{\mathbf{x}}-\mathbf{x}\|.
\end{aligned}$$

Since this holds for every $u \in [0,1]$, it also bounds $|\hat{I}_\Delta^{(2)}|$, and thus setting $C(\mathbf{x},\widetilde{\mathbf{x}})$ to be twice the right-hand side verifies Assumption 3.1.

Lipschitz score. The previous approach relied on boundedness in the score. Here we utilise first-order smoothness, by supposing that the score function $s(\mathbf{x},t)$ is Lipschitz in $\mathbf{x}$. This means that there exists some constant $L \ge 0$ such that,

$$\|s(\mathbf{x},t) - s(\widetilde{\mathbf{x}},t)\| \le L\|\mathbf{x}-\widetilde{\mathbf{x}}\|,$$

for all $\mathbf{x}, \widetilde{\mathbf{x}}$. As in Section 3.4, let $f(u) = \langle s(\mathbf{x}+u(\widetilde{\mathbf{x}}-\mathbf{x}),t), \widetilde{\mathbf{x}}-\mathbf{x}\rangle$, so that $\hat{I}_\Delta^{(2)} = \frac{1}{2}(f(0)+f(1))$. By the Cauchy–Schwarz inequality, we have $|f(u)-f(0)| \le Lu\|\widetilde{\mathbf{x}}-\mathbf{x}\|^2$ and $|f(u)-f(1)| \le L(1-u)\|\widetilde{\mathbf{x}}-\mathbf{x}\|^2$ for $u \in [0,1]$. Hence,

$$\left|f(u) - \hat{I}_\Delta^{(2)}\right| \le \tfrac{1}{2}|f(u)-f(0)| + \tfrac{1}{2}|f(u)-f(1)| \le \tfrac{L}{2}\|\widetilde{\mathbf{x}}-\mathbf{x}\|^2.$$

Thus, Assumption 3.1 holds with constant,

$$C(\mathbf{x},\widetilde{\mathbf{x}}) = \frac{L}{2}\|\widetilde{\mathbf{x}}-\mathbf{x}\|^2.$$

Note that in fact, this proof only requires the weaker condition given by,

$$|\langle s(\mathbf{x},t) - s(\widetilde{\mathbf{x}},t), \mathbf{x}-\widetilde{\mathbf{x}}\rangle| \le L\|\mathbf{x}-\widetilde{\mathbf{x}}\|^2.$$

E Implementation considerations

E.1 Practical termination

Algorithm 2 gives an exact method and avoids discretisation bias, but may require many score evaluations. Algorithm 3 is biased since it perturbs the acceptance probability, but it is computationally much cheaper. To get the best of both worlds, we propose a hybrid sampling algorithm: attempt Algorithm 2 K times to get an accept or reject decision; use Algorithm 3 if this fails. This controls the worst-case computational cost while retaining an exact update whenever the Bernoulli factory terminates in time. In our experiments, we used this hybrid sampling strategy for synthetic 2D datasets, where it is feasible, and showed that on image datasets, using only Algorithm 3 still yields performance gains.

E.2 Optimal scaling

Optimal scaling theory addresses the choice of step size h to achieve efficient exploration of the target: if proposed moves are too small, the chain mixes slowly, whereas overly large proposals lead to frequent rejections and poor exploration. The goal is to strike an optimal balance by selecting a proposal step size h that maximizes sampling efficiency. We establish the optimal scaling of h in Algorithm 2 by maximizing the Expected Squared Jumping Distance (ESJD) in the stationary phase:

$$\max_h \text{ESJD}(h) := \max_h \mathbb{E}_{\mathbf{x}_t^{(i)} \sim p_t}[\|\mathbf{x}_t^{(i+1)} - \mathbf{x}_t^{(i)}\|^2].$$

The justification for using ESJD as an optimization criterion comes from its connection to diffusion limits [Roberts et al., 1997]. When weak convergence to a diffusion limit is established, the ESJD

Algorithm 4 Hybrid MADM algorithm

Input: $\mathbf{x}$, $\widetilde{\mathbf{x}}$, $s(\cdot, t)$, h, $C(\cdot, \cdot)$, `use_exact` $\in \{$`True`, `False`$\}$.
Set $\alpha' = (1 + H_t(\mathbf{x}, \widetilde{\mathbf{x}}) \exp(\hat{I}_{\Delta}^{(2)} + C(\mathbf{x}, \widetilde{\mathbf{x}})))^{-1}$
if `use_exact` **then**
  for $k = 0, \ldots, K-1$ **do**
    if $U \leq \alpha'$ for $U \sim \mathcal{U}([0,1])$ **then**
      return REJECT.
    end if
    Compute W.
    if $U \leq W$ for $U \sim \mathcal{U}([0,1])$ **then**
      return ACCEPT.
    end if
  end for
end if
return Alg. 3

of the Markov chain converges to the quadratic variation of the limiting diffusion [Roberts and Rosenthal, 2001, Yang et al., 2020]. A larger ESJD generally indicates that the chain is exploring the state space more efficiently, avoiding both overly conservative moves and excessive rejection.

We establish the optimal scaling theory for the Barker-adjusted Langevin algorithm in Algorithm 2. For clarity, we present the result for multivariate Gaussian targets. The argument extends in a standard way to broader classes of target distributions [Yang et al., 2020], with the same limiting optimal acceptance rate.

Proposition E.1. *Assume that p_t is a d-dimensional standard multivariate Gaussian and that the score is exact, i.e., $s(\cdot, t) = \nabla \log p_t(\cdot)$. Choosing h to maximize the ESJD for Algorithm 2, then the optimal scaling for h is $h = h_d \sim d^{-1/3}$, and the optimal expected acceptance rate converges to* 0.347 *(to three decimal places) as $d \to \infty$.*

Note that if h scales with d faster (or slower) than order $d^{-1/3}$, the limiting acceptance probability becomes $1/2$ (or 0), which cannot be optimal for maximizing the ESJD. Therefore, we will write $h = h_d = \ell^2 d^{-1/3}$ for some $\ell > 0$. The proof follows the same line of argument as in Roberts and Rosenthal [1998], with the acceptance function modified according to Agrawal et al. [2023].

First, for notational clarity, we suppress the time index in the target p_t. By assumption,

$$p_t(\mathbf{x}) = \prod_{i=1}^{d} \phi(x_i), \qquad \text{where } \phi(x_i) = (2\pi)^{-1/2} \exp(-x_i^2/2).$$

Under a perfect score, the resulting exact algorithm reduces to the Barker-adjusted Langevin algorithm: proposals are generated via ULA and accepted according to Barker's rule. For consistency in notation, we also suppress the subscript B in the acceptance probability α_B and write α_d to highlight its dependence on the dimension. That is,

$$\alpha_d(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{1}{1 + \exp(-W_d)} \tag{15}$$

where $W_d = \log\left(\frac{p_t(\tilde{\mathbf{x}}) q(\mathbf{x}|\tilde{\mathbf{x}})}{p_t(\mathbf{x}) q(\tilde{\mathbf{x}}|\mathbf{x})}\right)$ is the log-acceptance ratio, and the ULA proposal

$$\tilde{x}_i = x_i - \frac{h}{2} x_i + \sqrt{h} z_i, \quad i = 1, \ldots, d, \tag{16}$$

where $x_i, z_i \overset{iid}{\sim} \mathcal{N}(0,1)$ in stationarity. Note that while the assumption of a multivariate Gaussian target simplifies the analysis of ULA proposal, the argument extends readily to more general settings [Yang et al., 2020]. We restrict attention to the Gaussian case only for simplicity.

We are interested in maximizing the ESJD, with step size scales with dimension as $h = \ell^2 d^{-1/3}$ for a constant $\ell > 0$ to yield a non-degenerate limit:

$$\text{ESJD}_d(h) = \mathbb{E}[\|\mathbf{x} - \tilde{\mathbf{x}}\|^2 \alpha_d(\mathbf{x}, \tilde{\mathbf{x}})] = d \cdot \mathbb{E}\left[(\tilde{x}_1 - x_1)^2 \alpha_d(\mathbf{x}, \tilde{\mathbf{x}})\right] \tag{17}$$

As we re-parameterize h by ℓ via $h = \ell^2 d^{-1/3}$, we can define an efficiency function over ℓ:

$$J_d(\ell) := d^{-2/3} \cdot \text{ESJD}_d(h). \tag{18}$$

Then maximizing ESJD(h) over h is equivalent to maximizing $J_d(\ell)$ over ℓ up to a constant $d^{-2/3}$.

Next, we establish the limiting distribution of W_d. As shown in standard algebraic expansions for Gaussian targets, the log-ratio simplifies perfectly to:

$$W_d = \sum_{i=1}^{d} \xi_{i,d}, \quad \text{where} \quad \xi_{i,d} = \frac{h}{8}(x_i^2 - \tilde{x}_i^2) \tag{19}$$

Lemma E.2. *As $d \to \infty$, $W_d \xrightarrow{\mathcal{D}} \mathcal{N}\left(-\frac{\ell^6}{32}, \frac{\ell^6}{16}\right)$.*

Proof. The proof follows the same line of arguments as the proof for MALA in Roberts and Rosenthal [1998]. Substituting the ULA proposal into $\xi_{i,d}$, we obtain:

$$\xi_{i,d} = \frac{h}{8}\left[h x_i^2 - \frac{h^2}{4} x_i^2 - 2\sqrt{h}\left(1 - \frac{h}{2}\right) x_i z_i - h z_i^2\right]. \tag{20}$$

Taking expectations using $\mathbb{E}[x_i^2] = 1, \mathbb{E}[z_i^2] = 1$, and $\mathbb{E}[x_i z_i] = 0$:

$$\mu_d = \mathbb{E}[\xi_{i,d}] = \frac{h}{8}\left(h - \frac{h^2}{4} - h\right) = -\frac{h^3}{32}. \tag{21}$$

For the variance, the dominant term is the cross-term $-\frac{h^{3/2}}{4} x_i z_i$. Thus:

$$\text{Var}(\xi_{i,d}) = \mathbb{E}\left[\left(-\frac{h^{3/2}}{4} x_i z_i\right)^2\right] + \mathcal{O}(h^4) = \frac{h^3}{16} + \mathcal{O}(h^4). \tag{22}$$

Because $h = \ell^2 d^{-1/3}$, the sum of means is $\mathbb{E}[W_d] = d\left(-\frac{h^3}{32}\right) = -\frac{\ell^6}{32}$. The sum of variances is $s_d^2 = \sum_{i=1}^{d} \text{Var}(\xi_{i,d}) = d\left(\frac{h^3}{16}\right) + d\mathcal{O}(h^4) \to \frac{\ell^6}{16}$. Finally, to establish asymptotic normality, we verify the Lyapunov condition for the fourth moment,

$$\sum_{i=1}^{d} \mathbb{E}\left[|\xi_{i,d} - \mathbb{E}\xi_{i,d}|^4\right] = O(dh^6) = O(d^{-1}) \to 0,$$

which completes the proof. □

Next, we establish the scaling limit of ESJD and the expected acceptance rate.

Proposition E.3 (Optimal Scaling Limit)**.** *As $d \to \infty$, the scaled ESJD converges to:*

$$\lim_{d\to\infty} J_d(\ell) = \ell^2 \cdot A(\ell), \quad A(\ell) := \mathbb{E}\left[\frac{1}{1 + \exp(-W)}\right] \tag{23}$$

where $W \sim \mathcal{N}(-\sigma^2/2, \sigma^2)$ and $\sigma^2 = \ell^6/16$.

Proof. The proof follows the same line of argument as in Roberts and Rosenthal [1998], Agrawal et al. [2023]. Observe that in stationarity, the d components are identically distributed. Thus, the full ESJD is exactly d times the single-coordinate ESJD:

$$J_d(\ell) = d^{-2/3} \cdot d \cdot \mathbb{E}\left[(\tilde{x}_1 - x_1)^2 \alpha_d(\mathbf{x}, \tilde{\mathbf{x}})\right] = d^{1/3}\mathbb{E}\left[(\tilde{x}_1 - x_1)^2 \alpha_d(\mathbf{x}, \tilde{\mathbf{x}})\right] \tag{24}$$

Write the single-coordinate squared jump distance as $(\tilde{x}_1 - x_1)^2 = h z_1^2 + R_{1,d}$, where $R_{1,d} = \mathcal{O}(h^{3/2})$ almost surely. Multiplying by $d^{1/3}$, we note $d^{1/3}h = \ell^2$. Thus, $d^{1/3}(\tilde{x}_1 - x_1)^2 \to \ell^2 z_1^2$ almost surely.

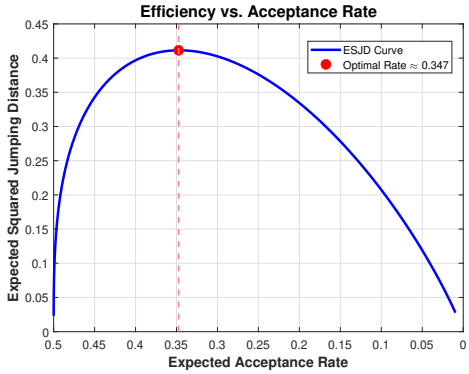

Figure 4: Optimal acceptance rate for Algorithm 2

Let $W_{d,-1} = \sum_{i=2}^{d} \xi_{i,d}$. Because $\xi_{1,d} \xrightarrow{\mathbb{P}} 0$, by Slutsky's Theorem, $W_{d,-1} \xrightarrow{\mathcal{D}} W$. Furthermore, since $W_{d,-1}$ is strictly independent of z_1, the joint distribution converges: $(z_1, W_{d,-1}) \xrightarrow{\mathcal{D}} (z_1, W)$, where $z_1 \perp\!\!\!\perp W$. By Slutsky's Theorem again on the pair $(z_1, W_d) = (z_1, W_{d,-1}) + (0, \xi_{1,d})$, we have $(z_1, W_d) \xrightarrow{\mathcal{D}} (z_1, W)$.

Let $g(w) = (1 + \exp(-w))^{-1}$ represent Barker's function. Because g is continuous and bounded, $g(w) \in (0, 1)$, the function $f(z, w) = z^2 g(w)$ is continuous. By the Continuous Mapping Theorem:

$$V_d := d^{1/3}(\tilde{x}_1 - x_1)^2 \alpha_d(\mathbf{x}, \tilde{\mathbf{x}}) \xrightarrow{\mathcal{D}} \ell^2 z_1^2 g(W). \tag{25}$$

To pass the limit inside the expectation, we require the sequence V_d to be uniformly integrable. Indeed, since $0 \leq \alpha_d \leq 1$, we have $|V_d| \leq d^{1/3}[\frac{h^2}{4} x_1^2 - h^{3/2} x_1 z_1 + h z_1^2] \leq \frac{\ell^4}{4} x_1^2 + \ell^3 |x_1 z_1| + \ell^2 z_1^2$. Because $x_1, z_1 \sim \mathcal{N}(0, 1)$, the dominating variable is integrable, which implies uniform integrability of $\{V_d\}_{d \geq 1}$. It follows that:

$$\lim_{d \to \infty} J_d(\ell) = \lim_{d \to \infty} \mathbb{E}[V_d] = \mathbb{E}\left[\ell^2 z_1^2 g(W)\right] = \ell^2 \mathbb{E}[z_1^2] \mathbb{E}[g(W)] = \ell^2 \mathbb{E}\left[\frac{1}{1 + \exp(-W)}\right]. \tag{26}$$

□

Finally, Proposition E.3 allows one to determine the optimal acceptance rate via numerical simulations. See Figure 4 for the efficiency curve, $\ell^2 \cdot A(\ell)$ versus $A(\ell)$. By choosing the optimal ℓ, which corresponds to the optimal step size h that maximizes the ESJD, the expected acceptance rate converges to 0.347 (to three decimal places) as $d \to \infty$.

F Convergence analysis

F.1 Convergence of MADM under a warm start

In this section, we establish convergence guarantees of the MADM corrector algorithm at a fixed noise level when initialised from a warm start. We begin with a general result for Barker-adjusted Markov chains.

As in Section 2.4, we take p to be a positive target density on $\mathbb{R}^d$ and q to be a proposal density. We write P_{B} for the transition kernel of the Markov chain with proposal q and acceptance probability α_{B} given in (7). That is, for every $\mathbf{x} \in \mathbb{R}^d$ and measurable $A \subseteq \mathbb{R}^d$,

$$P_{\mathrm{B}}(\mathbf{x}, A) = \int_A \alpha_{\mathrm{B}}(\mathbf{x}, \widetilde{\mathbf{x}}) q(\widetilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\widetilde{\mathbf{x}} + \left(1 - \int_{\mathbb{R}^d} \alpha_{\mathrm{B}}(\mathbf{x}, \widetilde{\mathbf{x}}) q(\widetilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\widetilde{\mathbf{x}}\right) \mathbf{1}_A(\mathbf{x}),$$

and we define P_{MH} analogously, with α_{B} replaced by α_{MH}. We refer to these as the Barker-adjusted and Metropolis-adjusted kernels respectively. Both are reversible with respect to p, so that p is a stationary distribution for each.

We use the notion of s-conductance introduced by Lovász and Simonovits [1993]. For a transition kernel P with stationary distribution p, the flow of P out of a measurable set $A \subseteq \mathbb{R}^d$ is given by,

$$\Phi_P(A) = \int_A P(\mathbf{x}, A^c) p(\mathbf{x}) \, \mathrm{d}\mathbf{x}.$$

For $s \in (0, 1/2]$, the s-conductance of P is then defined as,

$$\Phi_s(P) = \inf \left\{ \frac{\Phi_P(A)}{p(A) - s} : s < p(A) \leq \frac{1}{2} \right\}.$$

The following lemma is a modification of Corollary 1.6 in Lovász and Simonovits [1993] refined for Barker-adjusted Markov chains. Note that, unlike the result of Lovász and Simonovits [1993], we do not rely on $1/2$-laziness. We find that such an assumption is not necessary for Barker-adjusted chains and that the resulting conductance quantity is that of the lazy Metropolis-adjusted chain.

Lemma F.1. *Let $(\mathbf{X}^{(i)})_{i \geq 0}$ be a Markov chain with transition kernel P_{B}, let $s \in (0, 1/2]$ and let*

$$M_s = \sup \left\{ |\mathbb{P}(\mathbf{X}^{(0)} \in A) - p(A)| : p(A) \leq s \right\}.$$

Then, for every $i \geq 0$ and measurable $A \subseteq \mathbb{R}^d$,

$$|\mathbb{P}(\mathbf{X}^{(i)} \in A) - p(A)| \leq M_s + \frac{M_s}{s} \left(1 - \frac{\Phi_s(P_{\mathrm{MH}})^2 \wedge 1}{8} \right)^i.$$

Proof. For the sake of brevity, we write μ_i for the law of $\mathbf{X}^{(i)}$. Following [Lovász and Simonovits, 1993, Section 1.C], for $u \in [0, 1]$ we define,

$$h_i(u) = \sup_g \left\{ \int g \, \mathrm{d}\mu_i - u \right\},$$

where the supremum is over all measurable functions $g : \mathbb{R}^d \to [0, 1]$ such that $\int g(\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x} = u$. Lovász and Simonovits [1993] show that h_i satisfies the following properties, none of which rely on laziness: (i) h_i is concave on $[0, 1]$; (ii) for every $u \in [0, 1]$ the supremum is attained by the indicator function of a measurable set A with $p(A) = u$; and (iii) $h_i(u)$ is non-increasing in i.

Fix $i \geq 1$, $s \leq u \leq 1/2$ and let A be a measurable set such that $p(A) = u$ and $h_i(u) = \mu_i(A) - u$, which exists by (ii). For $\mathbf{x}, \widetilde{\mathbf{x}} \in \mathbb{R}^d$, let

$$\theta(\mathbf{x}, \widetilde{\mathbf{x}}) = \big(1 - \alpha_{\mathrm{B}}(\mathbf{x}, \widetilde{\mathbf{x}})\big) \mathbf{1}_A(\mathbf{x}) + \alpha_{\mathrm{B}}(\mathbf{x}, \widetilde{\mathbf{x}}) \mathbf{1}_A(\widetilde{\mathbf{x}}),$$

denote the probability that the chain lands in A given the current state $\mathbf{x}$ and the proposal $\widetilde{\mathbf{x}}$, so that $P_{\mathrm{B}}(\mathbf{x}, A) = \int \theta(\mathbf{x}, \widetilde{\mathbf{x}}) q(\widetilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\widetilde{\mathbf{x}}$. We split this function as, $P_{\mathrm{B}}(\mathbf{x}, A) = \frac{1}{2}(g_1(\mathbf{x}) + g_2(\mathbf{x}))$ where,

$$g_1(\mathbf{x}) = \int \big((2\theta(\mathbf{x}, \widetilde{\mathbf{x}}) - 1) \vee 0 \big) q(\widetilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\widetilde{\mathbf{x}},$$

$$g_2(\mathbf{x}) = \int \big(2\theta(\mathbf{x}, \widetilde{\mathbf{x}}) \wedge 1 \big) q(\widetilde{\mathbf{x}} \mid \mathbf{x}) \, \mathrm{d}\widetilde{\mathbf{x}},$$

and we note that both functions take values in $[0, 1]$. For $j \in \{1, 2\}$, let $u_j = \int g_j(\mathbf{x}) p(\mathbf{x}) \, \mathrm{d}\mathbf{x}$ so that,

$$\begin{aligned} u_1 + u_2 &= 2 \int P_{\mathrm{B}}(\mathbf{x}, A) p(\mathbf{x}) \, \mathrm{d}\mathbf{x} \\ &= 2u. \end{aligned}$$

Since g_j is measurable and takes values in $[0, 1]$, we obtain that

$$\begin{aligned} h_i(u) &= \int P_{\mathrm{B}}(\mathbf{x}, A) \, \mathrm{d}\mu_{i-1}(\mathbf{x}) - u \\ &= \frac{1}{2} \left(\int g_1 \, \mathrm{d}\mu_{i-1} - u_1 \right) + \frac{1}{2} \left(\int g_2 \, \mathrm{d}\mu_{i-1} - u_2 \right) \\ &\leq \frac{1}{2} h_{i-1}(u_1) + \frac{1}{2} h_{i-1}(u_2). \end{aligned}$$

Next, we use that $(2\theta \wedge 1) - \theta = \theta \wedge (1-\theta)$ to obtain,

$$\begin{aligned}
u_2 - u &= \int \big(g_2(\mathbf{x}) - P_{\mathrm{B}}(\mathbf{x}, A)\big)p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
&= \iint \big(\theta(\mathbf{x},\widetilde{\mathbf{x}}) \wedge (1-\theta(\mathbf{x},\widetilde{\mathbf{x}}))\big)p(\mathbf{x})q(\widetilde{\mathbf{x}} \mid \mathbf{x})\,\mathrm{d}\widetilde{\mathbf{x}}\,\mathrm{d}\mathbf{x} \\
&= \left(\int_A \int_{A^c} + \int_{A^c}\int_A\right)\big(\alpha_{\mathrm{B}}(\mathbf{x},\widetilde{\mathbf{x}}) \wedge (1-\alpha_{\mathrm{B}}(\mathbf{x},\widetilde{\mathbf{x}}))\big)p(\mathbf{x})q(\widetilde{\mathbf{x}} \mid \mathbf{x})\,\mathrm{d}\widetilde{\mathbf{x}}\,\mathrm{d}\mathbf{x}.
\end{aligned}$$

By (7), we have $\alpha_{\mathrm{B}} \wedge (1-\alpha_{\mathrm{B}}) = \alpha_{\mathrm{MH}}/(1+R)$. We exchange the roles of $\mathbf{x}$ and $\widetilde{\mathbf{x}}$ in the second integral and use that, by the definition of R in (8),

$$\begin{aligned}
p(\widetilde{\mathbf{x}})q(\mathbf{x} \mid \widetilde{\mathbf{x}}) &= R(\mathbf{x},\widetilde{\mathbf{x}})p(\mathbf{x})q(\widetilde{\mathbf{x}} \mid \mathbf{x}), \qquad R(\widetilde{\mathbf{x}},\mathbf{x}) = R(\mathbf{x},\widetilde{\mathbf{x}})^{-1}, \\
\alpha_{\mathrm{MH}}(\widetilde{\mathbf{x}},\mathbf{x}) &= R(\mathbf{x},\widetilde{\mathbf{x}})^{-1}\alpha_{\mathrm{MH}}(\mathbf{x},\widetilde{\mathbf{x}}),
\end{aligned}$$

to obtain,

$$\begin{aligned}
u_2 - u &= \int_A \int_{A^c} \left(\frac{\alpha_{\mathrm{MH}}(\mathbf{x},\widetilde{\mathbf{x}})}{1+R(\mathbf{x},\widetilde{\mathbf{x}})} + R(\mathbf{x},\widetilde{\mathbf{x}})\frac{\alpha_{\mathrm{MH}}(\widetilde{\mathbf{x}},\mathbf{x})}{1+R(\widetilde{\mathbf{x}},\mathbf{x})}\right)p(\mathbf{x})q(\widetilde{\mathbf{x}} \mid \mathbf{x})\,\mathrm{d}\widetilde{\mathbf{x}}\,\mathrm{d}\mathbf{x} \\
&= \int_A \int_{A^c} \left(\frac{\alpha_{\mathrm{MH}}(\mathbf{x},\widetilde{\mathbf{x}})}{1+R(\mathbf{x},\widetilde{\mathbf{x}})} + R(\mathbf{x},\widetilde{\mathbf{x}})\frac{R(\mathbf{x},\widetilde{\mathbf{x}})^{-1}\alpha_{\mathrm{MH}}(\mathbf{x},\widetilde{\mathbf{x}})}{1+R(\mathbf{x},\widetilde{\mathbf{x}})^{-1}}\right)p(\mathbf{x})q(\widetilde{\mathbf{x}} \mid \mathbf{x})\,\mathrm{d}\widetilde{\mathbf{x}}\,\mathrm{d}\mathbf{x} \\
&= \int_A \int_{A^c} \alpha_{\mathrm{MH}}(\mathbf{x},\widetilde{\mathbf{x}})p(\mathbf{x})q(\widetilde{\mathbf{x}} \mid \mathbf{x})\,\mathrm{d}\widetilde{\mathbf{x}}\,\mathrm{d}\mathbf{x}.
\end{aligned}$$

Thus, by the definition of s-conductance and using $p(A) = u \in [s, 1/2]$ (the case $u = s$ being trivial),

$$u_2 - u = \Phi_{P_{\mathrm{MH}}}(A) \geq \Phi_s(P_{\mathrm{MH}})(u-s).$$

Since $u_1 + u_2 = 2u$, we also obtain that $u - u_1 \geq \Phi_s(P_{\mathrm{MH}})(u-s)$. Thus, we have that $u_1 \leq u - (\Phi_s(P_{\mathrm{MH}}) \wedge 1)(u-s) \leq u + (\Phi_s(P_{\mathrm{MH}}) \wedge 1)(u-s) \leq u_2$ and hence, by the concavity of h_{i-1},

$$h_i(u) \leq \frac{1}{2}\Big(h_{i-1}\big(u - (\Phi_s(P_{\mathrm{MH}}) \wedge 1)(u-s)\big) + h_{i-1}\big(u + (\Phi_s(P_{\mathrm{MH}}) \wedge 1)(u-s)\big)\Big).$$

When $1/2 \leq u \leq 1-s$, the same argument applies, using that $\Phi_{P_{\mathrm{MH}}}(A) = \Phi_{P_{\mathrm{MH}}}(A^c)$ by reversibility and $p(A^c) = 1-u$, which gives the analogous inequality,

$$h_i(u) \leq \frac{1}{2}\Big(h_{i-1}\big(u - (\Phi_s(P_{\mathrm{MH}}) \wedge 1)(1-u-s)\big) + h_{i-1}\big(u + (\Phi_s(P_{\mathrm{MH}}) \wedge 1)(1-u-s)\big)\Big).$$

An upper bound on h_i then follows from the induction argument in Theorem 1.4 of Lovász and Simonovits [1993]. Here, it is shown that for any c_1, c_2 such that,

$$h_0(u) \leq c_1 + c_2\Big(\sqrt{u-s} \wedge \sqrt{1-s-u}\Big), \qquad \text{for all } s \leq u \leq 1-s,$$

it follows that, for all $i \geq 0$, $s \leq u \leq 1-s$,

$$h_i(u) \leq c_1 + c_2\Big(\sqrt{u-s} \wedge \sqrt{1-s-u}\Big)\left(1 - \frac{\Phi_s(P_{\mathrm{MH}})^2 \wedge 1}{8}\right)^i.$$

Thus, to complete the proof, we verify that the base case with $c_1 = M_s$ and $c_2 = M_s/s$ holds for all $s \leq u \leq 1-s$. Let $u \leq s$ and let A be a measurable set with $p(A) = u$ so that $h_0(u) = \mu_0(A) - p(A) \leq M_s$. Similarly, for $u \geq 1-s$, taking complements,

$$h_0(u) = \sup_{p(A)=u} \big(\mu_0(A) - p(A)\big) = \sup_{p(B)=1-u} \big(p(B) - \mu_0(B)\big) \leq M_s.$$

For $s < u < 1-s$, we use the concavity of h_0 as well as the fact that $h_0(0) \geq 0$ and $h_0(1) = 0$, to obtain,

$$h_0(u) \leq (u/s)h_0(s), \qquad h_0(u) \leq ((1-u)/s)h_0(1-s).$$

Hence, since $h_0(s), h_0(1-s) \le M_s$, we obtain,

$$\begin{aligned} h_0(u) &\le \frac{M_s}{s}\big(u \wedge (1-u)\big) \\ &= M_s + \frac{M_s}{s}\big((u-s) \wedge (1-s-u)\big) \\ &\le M_s + \frac{M_s}{s}\Big(\sqrt{u-s} \wedge \sqrt{1-s-u}\Big). \end{aligned}$$

The bound in the statement then follows by noting that, for any measurable $A \subseteq \mathbb{R}^d$, $\mu_i(A) - p(A) \le h_i(p(A))$ and $p(A) - \mu_i(A) = \mu_i(A^c) - p(A^c) \le h_i(1-p(A))$, and that $h_i(u) \le h_0(u) \le M_s$ for $u \notin [s, 1-s]$ by (iii). □

We now recall a lower bound on the s-conductance of P_{MH} due to Wu et al. [2022], for the case where q is the Langevin proposal used in Algorithm 1 and p is strongly log-concave. Their result is stated in terms of the isoperimetric constant of p, which, for an m-strongly log-concave density, is at least $\log(2)\sqrt{m}$ [see Wu et al., 2022, proof of Theorem 1].

Lemma F.2. *Suppose that $\log p$ is twice continuously differentiable with $m\mathbf{I} \preceq -\nabla^2 \log p \preceq L\mathbf{I}$, and let P_{MH} be the Metropolis-adjusted kernel with proposal $q(\cdot \mid \mathbf{x}) = \mathcal{N}(\mathbf{x} + \frac{h}{2}\nabla \log p(\mathbf{x}), h\mathbf{I})$. Then, there exist universal constants $c_1, c_2 > 0$ such that, for every $s \in (0, 1/2]$ and*

$$h \le \frac{c_1}{L\sqrt{d}\log^2(dL/(ms))},$$

we have $\Phi_s(P_{\mathrm{MH}}) \ge c_2\sqrt{hm}$.

Explicit values of c_1, c_2 can be obtained from conditions (33), (34) and the bound (40) in the proof of Theorem 3 of Wu et al. [2022]. Note that their step size is half of ours, and that their bound concerns the lazy version of P_{MH}, whose s-conductance is half that of P_{MH}.

Assumption F.3. *The function $\log p_{data}$ is twice continuously differentiable and there exist $0 < m \le L$ such that $m\mathbf{I} \preceq -\nabla^2 \log p_{data}(\mathbf{x}) \preceq L\mathbf{I}$ for every $\mathbf{x} \in \mathbb{R}^d$.*

Recall from Section 2.2 that p_t is the law of $r_t(\mathbf{X}_0 + \sigma_t \mathbf{Z})$, where $\mathbf{X}_0 \sim p_{\text{data}}$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are independent. Under Assumption F.3, p_t satisfies the same assumption with constants [see Gao et al., 2025, proof of Proposition 7 and Lemma 9],

$$m_t = \frac{m}{r_t^2(1 + m\sigma_t^2)}, \qquad L_t = \frac{L \wedge \sigma_t^{-2}}{r_t^2}.$$

For a measure μ, we write the total variation distance as $\|\mu - p\|_{\mathrm{TV}} = \sup_A |\mu(A) - p(A)|$, where the supremum is over measurable sets $A \subseteq \mathbb{R}^d$. We also write the chi-squared divergence as

$$\chi^2(\mu \,\|\, p) = \begin{cases} \int \big(\frac{\mathrm{d}\mu}{\mathrm{d}p}(\mathbf{x}) - 1\big)^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x}, & \text{if } \mu \ll p, \\ \infty, & \text{otherwise.} \end{cases} \tag{27}$$

Combining Lemmas F.1 and F.2, we obtain the following convergence guarantee for the MADM corrector at a fixed noise level.

Proposition F.4. *Fix $t \in [0, 1]$ and let $(\mathbf{X}^{(i)})_{i \ge 0}$ be the iterates of Algorithm 1 with the exact score $\nabla \log p_t$ and where the accept-reject step is performed by Algorithm 2. Suppose that Assumption F.3 holds and $\chi^2(\mu_0 \,\|\, p_t) \le D$ where we let μ_i denote the law of $\mathbf{X}^{(i)}$. Given $\varepsilon > 0$ with $\varepsilon^2 \le 2D$, suppose that*

$$h \le \frac{c_1}{L_t\sqrt{d}\log^2(4dL_tD/(m_t\varepsilon^2))},$$

where c_1 is the constant of Lemma F.2. Then, there exists a universal constant $c_3 > 0$ such that $\|\mu_i - p_t\|_{\mathrm{TV}} \le \varepsilon$ whenever,

$$i \ge \frac{1}{c_3 h m_t} \log\left(\frac{4D}{\varepsilon^2}\right).$$

Proof. By Theorem 3.4, each accept-reject step accepts with probability α_{B}. Hence, $(\mathbf{X}^{(i)})_{i\geq 0}$ is a Markov chain with transition kernel P_{B}, with target $p = p_t$ and proposal $q(\cdot \mid \mathbf{x}) = \mathcal{N}(\mathbf{x} + \frac{h}{2}\nabla \log p_t(\mathbf{x}), h\mathbf{I})$. Thus, we may apply Lemma F.1. To bound M_s, we note that for any measurable A, the Cauchy–Schwarz inequality gives,

$$\begin{aligned}|\mu_0(A) - p_t(A)| &= \left| \int_A \left(\frac{\mathrm{d}\mu_0}{\mathrm{d}p_t}(\mathbf{x}) - 1 \right) p_t(\mathbf{x})\, \mathrm{d}\mathbf{x} \right| \\ &\leq \sqrt{D\, p_t(A)}.\end{aligned}$$

Thus, setting $s = \varepsilon^2/(4D)$, which lies in $(0, 1/2]$ since $\varepsilon^2 \leq 2D$, we obtain $M_s \leq \sqrt{sD} = \varepsilon/2$.

Since p_t satisfies Assumption F.3 with constants m_t and L_t, Lemma F.2 gives $\Phi_s(P_{\mathrm{MH}}) \geq c_2\sqrt{hm_t}$ and hence there exists a universal constant $c_3 > 0$ such that,

$$\frac{\Phi_s(P_{\mathrm{MH}})^2 \wedge 1}{8} \geq \frac{c_2^2 h m_t \wedge 1}{8} \geq c_3 h m_t.$$

Applying Lemma F.1 with $p = p_t$, we obtain that, for every measurable A,

$$|\mu_i(A) - p_t(A)| \leq \frac{\varepsilon}{2} + \frac{2D}{\varepsilon}\big(1 - c_3 h m_t\big)^i \leq \frac{\varepsilon}{2} + \frac{2D}{\varepsilon} e^{-c_3 h m_t i},$$

which is at most ε for i as in the statement. □

F.2 Oracle complexity of the MADM-based sampler

We now analyse the complexity of the MADM-based predictor-corrector sampler in setting of Assumption F.3. We work in the VP setting of Section 2.2 and fix noise levels $1 \geq t_0 > t_1 > \cdots > t_N \geq 0$. For $k \in \{1, \ldots, N\}$, let $\beta_k = 1 - r^2_{t_{k-1}}/r^2_{t_k}$, so that if $\mathbf{X} \sim p_{t_k}$ and $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ are independent, then $\sqrt{1-\beta_k}\mathbf{X} + \sqrt{\beta_k}\mathbf{Z} \sim p_{t_{k-1}}$. For each noise level t_k, we perform a predictor step, followed by n_k steps of the MADM corrector. The predictor is the ancestral sampler of Ho et al. [2020] with the exact score, whose step from noise level t_{k-1} to t_k has transition kernel,

$$Q_k(\mathbf{x}, \cdot) = \mathcal{N}\left(\frac{\mathbf{x} + \beta_k \nabla \log p_{t_{k-1}}(\mathbf{x})}{\sqrt{1-\beta_k}}, \beta_k \mathbf{I} \right).$$

The corrector at level k is Algorithm 1, where the accept-reject step is performed by Algorithm 2. By Theorem 3.4, its transition kernel is the Barker-adjusted kernel P_k with target p_{t_k} and proposal $q_k(\cdot \mid \mathbf{x}) = \mathcal{N}(\mathbf{x} + \frac{h_k}{2}\nabla \log p_{t_k}(\mathbf{x}), h_k \mathbf{I})$. Since the sampler is initialised from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, the law of its output at level k is given by,

$$\nu_k = \nu_{k-1} Q_k P_k^{n_k}, \qquad \nu_0 = \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

The following lemma decomposes the error of the sampler into the error of the initialisation and the error made at each level when the sampler is started from the exact marginal. In particular, the warm start needed at level k concerns $p_{t_{k-1}} Q_k$ rather than the law $\nu_{k-1} Q_k$ of the sampler.

Lemma F.5. *For every $N \geq 1$,*

$$\|\nu_N - p_{t_N}\|_{\mathrm{TV}} \leq \|\nu_0 - p_{t_0}\|_{\mathrm{TV}} + \sum_{k=1}^{N} \big\| p_{t_{k-1}} Q_k P_k^{n_k} - p_{t_k} \big\|_{\mathrm{TV}}.$$

Proof. Fix $k \in \{1, \ldots, N\}$. By the triangle inequality,

$$\|\nu_k - p_{t_k}\|_{\mathrm{TV}} \leq \big\| \nu_{k-1} Q_k P_k^{n_k} - p_{t_{k-1}} Q_k P_k^{n_k} \big\|_{\mathrm{TV}} + \big\| p_{t_{k-1}} Q_k P_k^{n_k} - p_{t_k} \big\|_{\mathrm{TV}}.$$

For any transition kernel P, probability measures μ, ν and measurable A, since $P(\cdot, A)$ takes values in $[0, 1]$,

$$|\mu P(A) - \nu P(A)| = \left| \int P(\mathbf{x}, A)\, \mathrm{d}(\mu - \nu)(\mathbf{x}) \right| \leq \|\mu - \nu\|_{\mathrm{TV}}.$$

Hence, the first term is at most $\|\nu_{k-1} - p_{t_{k-1}}\|_{\mathrm{TV}}$ and the result follows by induction on k. □

Next, we bound the warm start at each level. The key observation is that, by Tweedie's identity, the predictor step Q_k has the same mean as the reverse transition of the forward process, so that it only errs in the shape of this transition.

Lemma F.6. *Suppose that Assumption F.3 holds, let $k \in \{1, \dots, N\}$ and let $\delta_k = |1-m_{t_k}| \vee |L_{t_k}-1|$. If $\beta_k \delta_k \le 1/4$, then*

$$\log\big(1 + \chi^2(p_{t_{k-1}} Q_k \,\|\, p_{t_k})\big) \le 7 d \beta_k^2 \delta_k^2.$$

Proof. For brevity, we write $\beta = \beta_k$ and $\delta = \delta_k$. We begin by noting that we can describe both $p_{t_{k-1}} Q_k$ and p_{t_k} as mixture distributions over $\mathbf{y} \sim p_{t_{k-1}}$. Indeed, for any measurable $A \subset \mathbb{R}^d$,

$$p_{t_{k-1}} Q_k(A) = \int Q_k(\mathbf{y}, A) p_{t_{k-1}}(\mathrm{d}\mathbf{y}), \qquad p_{t_k}(A) = \int p_{t_k|t_{k-1}}(A|\mathbf{y}) p_{t_{k-1}}(\mathrm{d}\mathbf{y}).$$

Thus, by the joint convexity of the chi-squared divergence, we obtain,

$$\chi^2(p_{t_{k-1}} Q_k \,\|\, p_{t_k}) \le \int \chi^2(Q_k(\mathbf{y}, \cdot) \,\|\, p_{t_k|t_{k-1}}(\cdot|\mathbf{y})) p_{t_{k-1}}(\mathrm{d}\mathbf{y}).$$

Therefore, it suffices to bound $\log(1 + \chi^2(Q_k(\mathbf{y}, \cdot) \,\|\, p_{t_k|t_{k-1}}(\cdot|\mathbf{y})))$ uniformly in $\mathbf{y} \in \mathbb{R}^d$.

Fix $\mathbf{y} \in \mathbb{R}^d$. By Tweedie's identity, we have that the mean of $p_{t_k|t_{k-1}}(\cdot|\mathbf{y})$ is,

$$\widehat{\mathbf{x}} = \frac{\mathbf{y} + \beta \nabla \log p_{t_{k-1}}(\mathbf{y})}{\sqrt{1-\beta}},$$

which coincides with the mean of $Q_k(\mathbf{y}, \cdot) = \mathcal{N}(\widehat{\mathbf{x}}, \beta \mathbf{I})$. Moreover, by Bayes' rule, $p_{t_k|t_{k-1}}(\cdot|\mathbf{y})$ has density proportional to $p_{t_k}(\mathbf{x}) \exp(-\|\mathbf{y} - \sqrt{1-\beta}\mathbf{x}\|^2/(2\beta))$, and so its density with respect to $Q_k(\mathbf{y}, \cdot)$ satisfies,

$$\frac{\mathrm{d} p_{t_k|t_{k-1}}(\cdot|\mathbf{y})}{\mathrm{d} Q_k(\mathbf{y}, \cdot)} = \frac{e^{-V}}{\mathbb{E}_{\mathbf{x} \sim Q_k(\mathbf{y}, \cdot)}\big[e^{-V(\mathbf{x})}\big]}, \qquad V(\mathbf{x}) = -\log p_{t_k}(\mathbf{x}) + \frac{1}{2\beta}\big\|\mathbf{y} - \sqrt{1-\beta}\mathbf{x}\big\|^2 - \frac{1}{2\beta}\|\mathbf{x} - \widehat{\mathbf{x}}\|^2.$$

Thus, we upper bound the chi-squared divergence by,

$$\begin{aligned}
1 + \chi^2\big(Q_k(\mathbf{y}, \cdot) \,\|\, p_{t_k|t_{k-1}}(\cdot|\mathbf{y})\big) &= \mathbb{E}_{\mathbf{x} \sim Q_k(\mathbf{y}, \cdot)}\left[\frac{\mathrm{d} Q_k(\mathbf{y}, \cdot)}{\mathrm{d} p_{t_k|t_{k-1}}(\cdot|\mathbf{y})}(\mathbf{x})\right] \\
&= \mathbb{E}_{\mathbf{x} \sim Q_k(\mathbf{y}, \cdot)}\big[e^{V(\mathbf{x})}\big] \mathbb{E}_{\mathbf{x} \sim Q_k(\mathbf{y}, \cdot)}\big[e^{-V(\mathbf{x})}\big] \\
&= \mathbb{E}\big[e^{V(\mathbf{x}) - V(\mathbf{x}')}\big],
\end{aligned}$$

where $\mathbf{x}, \mathbf{x}'$ are independent samples of $Q_k(\mathbf{y}, \cdot)$. Using the Gaussian inequality of Maurey and Pisier [e.g. see Bobkov and Volzone, 2024], we thus obtain,

$$\begin{aligned}
1 + \chi^2\big(Q_k(\mathbf{y}, \cdot) \,\|\, p_{t_k|t_{k-1}}(\cdot|\mathbf{y})\big) &\le \mathbb{E}\left[\exp\left(\frac{\pi}{2}\langle \nabla V(\mathbf{x}), \mathbf{x}' - \widehat{\mathbf{x}}\rangle\right)\right] \\
&\le \mathbb{E}\left[\exp\left(\frac{\pi^2 \beta}{8}\|\nabla V(\mathbf{x})\|^2\right)\right].
\end{aligned}$$

Since $\nabla^2 V = -\nabla^2 \log p_{t_k} - \mathbf{I}$ and p_{t_k} satisfies Assumption F.3 with constants m_{t_k} and L_{t_k}, we have $-\delta \mathbf{I} \preceq \nabla^2 V \preceq \delta \mathbf{I}$ and thus, $\|\nabla V(\mathbf{x})\|^2 \le 2\|\nabla V(\widehat{\mathbf{x}})\|^2 + 2\delta^2 \|\mathbf{x} - \widehat{\mathbf{x}}\|^2$. Thus, we obtain the upper bound,

$$\begin{aligned}
\log\Big(1 + \chi^2\big(Q_k(\mathbf{y}, \cdot) \,\|\, p_{t_k|t_{k-1}}(\cdot|\mathbf{y})\big)\Big) &\le \frac{\pi^2 \beta}{4}\|\nabla V(\widehat{\mathbf{x}})\|^2 + \log \mathbb{E}\left[\exp\left(\frac{\pi^2 \beta \delta^2}{4}\|\mathbf{x} - \widehat{\mathbf{x}}\|^2\right)\right] \\
&\le \frac{\pi^2 \beta}{4}\|\nabla V(\widehat{\mathbf{x}})\|^2 - \frac{d}{2}\log\left(1 - \frac{\pi^2 \beta^2 \delta^2}{2}\right),
\end{aligned}$$

where in the last line we use that $\mathbf{x} - \widehat{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \beta \mathbf{I})$ and $\pi^2 \beta^2 \delta^2 / 2 < 1$.

It remains to bound $\|\nabla V(\widehat{\mathbf{x}})\|$, for which we use that the means of $Q_k(\mathbf{y}, \cdot)$ and $p_{t_k|t_{k-1}}(\cdot|\mathbf{y})$ coincide. By integration by parts, the score of $p_{t_k|t_{k-1}}(\cdot|\mathbf{y})$ has mean zero under it, that is,

$$\mathbb{E}_{\mathbf{x} \sim p_{t_k|t_{k-1}}(\cdot|\mathbf{y})}\big[\nabla V(\mathbf{x}) + (\mathbf{x} - \widehat{\mathbf{x}})/\beta\big] = \mathbf{0},$$

and since the mean of $p_{t_k|t_{k-1}}(\cdot|\mathbf{y})$ is $\widehat{\mathbf{x}}$, it follows that $\mathbb{E}_{\mathbf{x}\sim p_{t_k|t_{k-1}}(\cdot|\mathbf{y})}[\nabla V(\mathbf{x})] = \mathbf{0}$. Hence,

$$\|\nabla V(\widehat{\mathbf{x}})\| = \big\|\mathbb{E}_{\mathbf{x}\sim p_{t_k|t_{k-1}}(\cdot|\mathbf{y})}[\nabla V(\widehat{\mathbf{x}}) - \nabla V(\mathbf{x})]\big\| \le \delta\, \mathbb{E}_{\mathbf{x}\sim p_{t_k|t_{k-1}}(\cdot|\mathbf{y})}\|\mathbf{x} - \widehat{\mathbf{x}}\|.$$

The negative log-density of $p_{t_k|t_{k-1}}(\cdot|\mathbf{y})$ has Hessian $-\nabla^2 \log p_{t_k} + ((1-\beta)/\beta)\mathbf{I} \succeq ((1-\beta\delta)/\beta)\mathbf{I}$, and so, by the Brascamp–Lieb inequality [Bakry et al., 2014, e.g. see],

$$\mathbb{E}_{\mathbf{x}\sim p_{t_k|t_{k-1}}(\cdot|\mathbf{y})}\|\mathbf{x} - \widehat{\mathbf{x}}\|^2 \le d\beta/(1-\beta\delta).$$

Thus, $\|\nabla V(\widehat{\mathbf{x}})\|^2 \le d\beta\delta^2/(1-\beta\delta)$. Combining these bounds and using that $\beta\delta \le 1/4$ and $-\log(1-u) \le u/(1-u)$ for $u \in [0,1)$, we conclude that,

$$\begin{aligned}\log\Big(1 + \chi^2\big(Q_k(\mathbf{y},\cdot)\,\|\,p_{t_k|t_{k-1}}(\cdot|\mathbf{y})\big)\Big) &\le \frac{\pi^2 d\beta^2\delta^2}{4(1-\beta\delta)} + \frac{\pi^2 d\beta^2\delta^2}{4(1-\pi^2\beta^2\delta^2/2)} \\ &\le 7d\beta^2\delta^2.\end{aligned}$$

Since this bound is uniform in $\mathbf{y}$, the result follows. □

Combining Lemmas F.5 and F.6 with Proposition F.4, we obtain the following convergence guarantee for the full sampler. For simplicity, we assume that $m \le 1 \le L$ and write $\kappa = L/m$.

Proposition F.7. *Suppose that Assumption F.3 holds with $m \le 1 \le L$ and let $\varepsilon \in (0,1)$. Let t_0 be such that $r_{t_0}^2 = \varepsilon^2/(2(\mathbb{E}\|\mathbf{X}_0\|^2 + d))$, let $N = \lceil 4L\sqrt{d}\log(1/r_{t_0}^2)\rceil$ and choose the remaining noise levels such that $r_{t_k}^2 = r_{t_0}^{2(1-k/N)}$. Suppose also that the step size $h_k = h$ and the number of corrector steps $n_k = n$ do not depend on k and satisfy,*

$$h \le \frac{c_1}{2L\sqrt{d}\log^2(896\kappa d^2L^2(\mathbb{E}\|\mathbf{X}_0\|^2 + d)/\varepsilon^4)}, \qquad n \ge \frac{1}{c_3 hm}\log\left(\frac{448dL^2(\mathbb{E}\|\mathbf{X}_0\|^2 + d)}{\varepsilon^4}\right),$$

where c_1 and c_3 are the constants of Lemma F.2 and Proposition F.4. Then, $\|\nu_N - p_{data}\|_{\mathrm{TV}} \le \varepsilon$.

Here, we assume that $r_1^2 \le \varepsilon^2/(2(\mathbb{E}\|\mathbf{X}_0\|^2 + d))$, so that t_0 exists. Taking the largest admissible step size, the total number of iterations $N(n+1)$ is $O(L\kappa d)$, up to logarithmic factors.

Proof. By Lemma F.5, we have that,

$$\|\nu_N - p_{\text{data}}\|_{\mathrm{TV}} \le \|\nu_0 - p_{t_0}\|_{\mathrm{TV}} + \sum_{k=1}^{N}\big\|p_{t_{k-1}}Q_kP_k^n - p_{t_k}\big\|_{\mathrm{TV}},$$

so it suffices to show that the first term is at most $\varepsilon/2$ and that each term in the sum is at most $\varepsilon/(2N)$.

For the first term, we write $r = r_{t_0}$, so that p_{t_0} is the law of $r\mathbf{X}_0 + \sqrt{1-r^2}\mathbf{Z}$. Since the KL divergence is convex in its first argument, comparing each Gaussian $\mathcal{N}(r\mathbf{x}, (1-r^2)\mathbf{I})$ with $\nu_0 = \mathcal{N}(\mathbf{0},\mathbf{I})$ gives,

$$\begin{aligned}\mathrm{KL}(p_{t_0}\,\|\,\nu_0) &\le \frac{1}{2}\mathbb{E}\big[r^2\|\mathbf{X}_0\|^2 - dr^2 - d\log(1-r^2)\big] \\ &\le \frac{r^2}{2}\big(\mathbb{E}\|\mathbf{X}_0\|^2 + d\big),\end{aligned}$$

where in the last line we use that $-\log(1-x) \le x/(1-x)$ and $r^2 \le 1/2$. Hence, by Pinsker's inequality, $\|\nu_0 - p_{t_0}\|_{\mathrm{TV}} \le \sqrt{\mathrm{KL}(p_{t_0}\,\|\,\nu_0)/2} \le \varepsilon/2$.

For the terms in the sum, fix $k \in \{1,\dots,N\}$. By the choice of noise levels, we have $\beta_k = \beta := 1 - r_{t_0}^{2/N}$, so that $(1-\beta)^N = r_{t_0}^2$. Since $m \le 1 \le L$ and $r_t^2(1+\sigma_t^2) = 1$, the expressions for m_t and L_t give,

$$m \le m_{t_k} \le 1 \le L_{t_k} \le 1 + L,$$

so that the quantity δ_k of Lemma F.6 satisfies $\delta_k \le L$. Hence, since $\beta \le -\log(1-\beta) = \log(1/r_{t_0}^2)/N \le 1/(4L\sqrt{d})$ by the choice of N, we have $\beta\delta_k \le 1/(4\sqrt{d}) \le 1/4$. Thus, from Lemma F.6 it follows that,

$$\chi^2(p_{t_{k-1}}Q_k\,\|\,p_{t_k}) \le e^{7d\beta^2\delta_k^2} - 1 \le 14dL^2\beta^2 =: D.$$

We now apply Proposition F.4 at noise level t_k, with initial law $\mu_0 = p_{t_{k-1}} Q_k$ and accuracy $\varepsilon/(2N)$. This first requires that $\varepsilon^2 \le 8N^2 D = 112 dL^2 N^2 \beta^2$, which holds since $\varepsilon < 1$ and $N\beta \ge \log(2)/2$. The remaining conditions of Proposition F.4 read,

$$h \le \frac{c_1}{L_{t_k}\sqrt{d}\log^2(16 d L_{t_k} N^2 D/(m_{t_k}\varepsilon^2))}, \qquad n \ge \frac{1}{c_3 h m_{t_k}} \log\left(\frac{16N^2 D}{\varepsilon^2}\right).$$

Since $N\beta \le \log(1/r_{t_0}^2) \le 1/r_{t_0}$, we have $16N^2 D = 224 dL^2 N^2 \beta^2 \le 448 dL^2(\mathbb{E}\|\mathbf{X}_0\|^2 + d)/\varepsilon^2$. Thus, since also $L_{t_k} \le 2L$, $L_{t_k}/m_{t_k} \le 2\kappa$ and $m_{t_k} \ge m$, these conditions are implied by those in the statement. Hence,

$$\left\| p_{t_{k-1}} Q_k P_k^n - p_{t_k} \right\|_{\mathrm{TV}} \le \frac{\varepsilon}{2N},$$

completing the proof. □

We now convert the number of iterations in Proposition F.7 into a bound on the number of score queries. By Proposition 3.5, the number of score queries made by Algorithm 2 is governed by the bound C, so we first give a choice of C under Assumption F.3 and bound its size at the proposal of Algorithm 1.

Lemma F.8. *Suppose that Assumption F.3 holds and fix $t \in [0,1]$. Then, Assumption 3.1 holds for the score $\nabla \log p_t$ with,*

$$C(\mathbf{x}, \widetilde{\mathbf{x}}) = \frac{L_t}{2}\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2.$$

Furthermore, if $\widetilde{\mathbf{x}} = \mathbf{x} + \frac{h}{2}\nabla \log p_t(\mathbf{x}) + \sqrt{h}\mathbf{z}$ for some $\mathbf{z} \in \mathbb{R}^d$, then,

$$C(\mathbf{x}, \widetilde{\mathbf{x}}) \le \frac{L_t h^2}{4}\|\nabla \log p_t(\mathbf{x})\|^2 + L_t h \|\mathbf{z}\|^2.$$

Proof. For $u \in [0,1]$, let $f(u) = \langle \nabla \log p_t(\mathbf{x} + u(\widetilde{\mathbf{x}} - \mathbf{x})), \widetilde{\mathbf{x}} - \mathbf{x}\rangle$, so that $\hat{I}_\Delta^{(2)} = \frac{1}{2}(f(0) + f(1))$. Since p_t satisfies Assumption F.3 with constants m_t and L_t, we have $|f'(u)| \le L_t\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2$ and thus, $|f(u) - f(0)| \le L_t\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2 u$ and $|f(u) - f(1)| \le L_t\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2(1-u)$. Hence,

$$\left|f(u) - \hat{I}_\Delta^{(2)}\right| \le \frac{1}{2}|f(u) - f(0)| + \frac{1}{2}|f(u) - f(1)| \le \frac{L_t}{2}\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2,$$

which establishes the first claim. The second claim then follows from the first by using that $\|\mathbf{a}+\mathbf{b}\|^2 \le 2\|\mathbf{a}\|^2 + 2\|\mathbf{b}\|^2$. □

Proposition F.9. *Suppose that the conditions of Proposition F.7 hold, that at each level k the accept-reject step of the corrector uses the bound C of Lemma F.8 with $t = t_k$, and let $\delta \in (0,1)$. Suppose further that,*

$$h \le \frac{1}{32L(d + \ell + \sqrt{\kappa\ell})}, \qquad \ell = \log(8Nn/\delta).$$

Then, with probability at least $1 - \delta/2 - N\varepsilon$, the total number of score queries made by the sampler is at most $15Nn\ell$.

Since ℓ depends on n only logarithmically, taking the largest admissible step size, h is of order $1/(L(d + \sqrt{\kappa}))$ and the total number of score queries is $O(L\kappa\sqrt{d}(d + \sqrt{\kappa}))$, up to logarithmic factors. In particular, this is $O(L\kappa d^{3/2})$ when $\kappa \le d^2$. Moreover, since the conditions depend on ε only logarithmically, taking $\varepsilon \le \delta/(2N)$ ensures that the bound holds with probability at least $1 - \delta$, at the cost of logarithmic factors.

Proof. At each level, the predictor evaluates the score once, and each corrector step evaluates the score at the current state and at the proposal. Thus, writing $J_{k,i}$ for the number of score queries made by Algorithm 2 in the i-th corrector step at level k, the total number of score queries is at most,

$$N + 2Nn + \sum_{k=1}^{N}\sum_{i=1}^{n} J_{k,i}.$$

We begin by bounding $J_{k,i}$ in terms of C. Fix k and i, let $\mathbf{x}$ be the state of the chain before the i-th corrector step at level k and let $\widetilde{\mathbf{x}}$ be the proposal. By Lemma F.8, $C = C(\mathbf{x}, \widetilde{\mathbf{x}})$ satisfies Assumption 3.1, so that Proposition 3.5 applies. Writing H and r as in Section 3.3, we have $1 + He^{\hat{I}_\Delta^{(2)}+C} \le e^{2C}(1 + Hr)$ since $|\log r - \hat{I}_\Delta^{(2)}| \le C$. Hence, conditionally on $\mathbf{x}$ and $\widetilde{\mathbf{x}}$, with probability at least $1 - \delta/(8Nn)$,

$$J_{k,i} \le 4Ce^{2C} \log(16Nn/\delta) + 4C + \log(16Nn/\delta).$$

In particular, if $C \le 1/2$, then $J_{k,i} \le (2e+1)\log(16Nn/\delta) + 2 \le 12\ell$, where we use that $\ell \ge \log 4$.

Thus, since $N + 2Nn + 12Nn\ell \le 15Nn\ell$ by $\ell \ge \log 4$, it suffices to show that, with probability at least $1 - \delta/2 - N\varepsilon$, we have $C \le 1/2$ and the above bound on $J_{k,i}$ holds at every corrector step.

To this end, we keep k and i fixed and let $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be the Gaussian increment of the proposal, so that $\widetilde{\mathbf{x}} = \mathbf{x} + \frac{h}{2}\nabla \log p_{t_k}(\mathbf{x}) + \sqrt{h}\mathbf{z}$. Let $T = \sqrt{2dL} + 4L\sqrt{\ell/m}$ and $S_k = \{\mathbf{y} \in \mathbb{R}^d : \|\nabla \log p_{t_k}(\mathbf{y})\| > T\}$, and consider the events,

$$A_{k,i} = \{\mathbf{x} \in S_k\}, \qquad Z_{k,i} = \big\{\|\mathbf{z}\| > \sqrt{d} + \sqrt{2\ell}\big\},$$

and let $B_{k,i}$ be the event that the above bound on $J_{k,i}$ fails. We also let $E_k = \bigcup_{i=1}^n (A_{k,i} \cup Z_{k,i} \cup B_{k,i})$ and, for a probability measure μ, we write $\mathbb{P}_\mu$ for the probability of an event when level k is started from μ, that is, when the predictor at level k is applied to a sample from μ.

Outside of $A_{k,i} \cup Z_{k,i}$, Lemma F.8 and $L_{t_k} \le 2L$ give,

$$\begin{aligned} C &\le \frac{Lh^2}{2}\|\nabla \log p_{t_k}(\mathbf{x})\|^2 + 2Lh\|\mathbf{z}\|^2 \\ &\le \frac{Lh^2T^2}{2} + 4Lh(d + 2\ell) \\ &\le \frac{1}{8} + \frac{1}{4} \le \frac{1}{2}, \end{aligned}$$

where in the last line we use that, by the condition on h, $hT \le 1/(2\sqrt{L})$ and $4Lh(d + 2\ell) \le 1/4$. Hence, outside of E_k, we have $C \le 1/2$ and the above bound on $J_{k,i}$ holds at every corrector step at level k.

We now bound the probability of E_k when level k is started from $p_{t_{k-1}}$. Since $\mathbf{z} \mapsto \|\mathbf{z}\|$ is 1-Lipschitz and $\mathbb{E}\|\mathbf{z}\| \le \sqrt{d}$, Gaussian concentration [see Bakry et al., 2014] gives $\mathbb{P}_{p_{t_{k-1}}}(Z_{k,i}) \le e^{-\ell} = \delta/(8Nn)$, and by the bound on $J_{k,i}$ above, $\mathbb{P}_{p_{t_{k-1}}}(B_{k,i}) \le \delta/(8Nn)$. For $A_{k,i}$, we first bound $p_{t_k}(S_k)$. Since p_{t_k} is m_{t_k}-strongly log-concave and $\mathbf{y} \mapsto \|\nabla \log p_{t_k}(\mathbf{y})\|$ is L_{t_k}-Lipschitz, Gaussian concentration gives, for $\mathbf{Y} \sim p_{t_k}$ and $\rho = 4L\sqrt{\ell/m}$,

$$\begin{aligned} p_{t_k}(S_k) &\le \mathbb{P}\big(\|\nabla \log p_{t_k}(\mathbf{Y})\| \ge \mathbb{E}\|\nabla \log p_{t_k}(\mathbf{Y})\| + \rho\big) \\ &\le \exp\left(-\frac{m_{t_k}\rho^2}{2L_{t_k}^2}\right) \\ &\le \left(\frac{\delta}{8Nn}\right)^2, \end{aligned}$$

where in the first line we use that, by integration by parts, $\mathbb{E}\|\nabla \log p_{t_k}(\mathbf{Y})\|^2 = -\mathbb{E}[\Delta \log p_{t_k}(\mathbf{Y})] \le dL_{t_k} \le 2dL$, and in the last line we use that $m_{t_k} \ge m$ and $L_{t_k} \le 2L$. To transfer this bound to the chain, recall from the proof of Proposition F.7 that $\chi^2(p_{t_{k-1}}Q_k \,\|\, p_{t_k}) \le D = 14dL^2\beta^2$, where $\beta \le 1/(4L\sqrt{d})$ so that $D \le 1$. Let $\mu_{k,i} = p_{t_{k-1}}Q_kP_k^{i-1}$ denote the law of $\mathbf{x}$ when level k is started from $p_{t_{k-1}}$. Since $p_{t_k}P_k = p_{t_k}$, the data processing inequality gives $\chi^2(\mu_{k,i} \,\|\, p_{t_k}) \le D \le 1$, and so, using the Cauchy–Schwarz inequality as in the proof of Proposition F.4,

$$\mathbb{P}_{p_{t_{k-1}}}(A_{k,i}) = \mu_{k,i}(S_k) \le p_{t_k}(S_k) + \sqrt{p_{t_k}(S_k)} \le \frac{\delta}{4Nn}.$$

Thus, by a union bound over the corrector steps at level k,

$$\mathbb{P}_{p_{t_{k-1}}}(E_k) \le \sum_{i=1}^n \left(\frac{\delta}{4Nn} + \frac{\delta}{8Nn} + \frac{\delta}{8Nn}\right) = \frac{\delta}{2N}.$$

Finally, the sampler starts level k from ν_{k-1} rather than $p_{t_{k-1}}$. Writing $K_k(\mathbf{y}, E_k)$ for the probability of E_k when level k is started from $\mathbf{y}$, which takes values in $[0,1]$, we obtain as in the proof of Lemma F.5,

$$\begin{aligned}\left|\mathbb{P}_{\nu_{k-1}}(E_k) - \mathbb{P}_{p_{t_{k-1}}}(E_k)\right| &= \left|\int K_k(\mathbf{y}, E_k)\,\mathrm{d}(\nu_{k-1} - p_{t_{k-1}})(\mathbf{y})\right| \\ &\le \|\nu_{k-1} - p_{t_{k-1}}\|_{\mathrm{TV}}.\end{aligned}$$

By Lemma F.5 and the bounds in the proof of Proposition F.7, $\|\nu_{k-1} - p_{t_{k-1}}\|_{\mathrm{TV}} \le \varepsilon/2 + (k-1)\varepsilon/(2N) \le \varepsilon$. Hence, by a union bound over k, the probability that some E_k occurs is at most $\delta/2 + N\varepsilon$, completing the proof. □

As noted in Section 5, the key feature of Proposition F.9 is the polylogarithmic dependence on $1/\varepsilon$ compared to results on diffusion samplers built from predictor-only steps which require $\widetilde{\mathcal{O}}(d/\varepsilon^2)$ score queries [Benton et al., 2024], or $\widetilde{\mathcal{O}}(L^2 d/\varepsilon^2)$ when the score is L-Lipschitz [Chen et al., 2023b], and the dimension dependence can be reduced to $\widetilde{\mathcal{O}}(L^2\sqrt{d}/\varepsilon)$ by using the probability flow ODE as the predictor together with an underdamped Langevin corrector [Chen et al., 2023a]. This is the same mechanism that separates ULA, whose complexity is polynomial in $1/\varepsilon$ [Durmus and Moulines, 2017, Mousavi-Hosseini et al., 2023], from MALA, whose complexity is polylogarithmic in $1/\varepsilon$ [Wu et al., 2022]. Recently, Chen et al. [2026] also obtained polylogarithmic guarantees for diffusion models by instead removing the bias of each predictor step with rejection sampling. Their guarantees hold under minimal assumptions on the data and with approximate scores, using $\widetilde{\mathcal{O}}(d_\star)$ score queries, where $d_\star \le d$ is an intrinsic dimension of the data, which equals d for data with full support, as under Assumption F.3.

The price of exact correction in Proposition 5.1 is a worse dependence on d and κ. The factor $d^{3/2}$ is the product of the $\sqrt{d}$ noise levels needed for the predictor to provide a warm start and the factor d in the number of corrector steps n_k. The latter comes from the cost of Algorithm 2, which grows exponentially in C (Proposition 3.5). At a typical proposal, $C = \frac{L_t}{2}\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2$ is of order Lhd, so we require $h \lesssim 1/(Ld)$, whereas MALA, which evaluates the density, mixes with step sizes of order $1/(L\sqrt{d})$ in $\widetilde{\mathcal{O}}(\kappa\sqrt{d})$ steps [Wu et al., 2022]. In the worst case, this cannot be improved by a better choice of C, since for isotropic Gaussian data the integrand in (9) is linear in u and varies by $L_t\|\widetilde{\mathbf{x}} - \mathbf{x}\|^2$ up to a constant factor. However, Chen et al. [2026] recover the rate of MALA for log-concave sampling using only gradients, so this extra factor is a cost of our estimator of the density ratio rather than of score-based access. The dependence on κ is the price of removing the discretisation error by mixing. At noise levels close to the data, p_t is as ill-conditioned as p_{data}, and the number of steps needed for a Langevin-type corrector to mix grows with κ, whether or not it is Metropolis-adjusted.

G Experiment details

Synthetic 2D Datasets In these experiments, we used the ancestral sampler from DDPM [Ho et al., 2020] as the predictor. This uses the discretisation

$$\mathbf{x}_t = \frac{1}{\sqrt{1-\beta_{t+1}}}(\mathbf{x}_{t+1} + \beta_{t+1} s_\phi(\mathbf{x}_{t+1}, t+1)) + \sqrt{\beta_{t+1}}\mathbf{z}_{i+1}$$

where the forward noising process assumes a discrete Markov chain $\{\mathbf{x}_0, \dots, \mathbf{x}_T\}$ constructed as $p(\mathbf{x}_{t+1}|\mathbf{x}_t) \sim \mathcal{N}(\mathbf{x}_{t+1}; \sqrt{1-\beta_t}\mathbf{x}_t, \beta_t \mathbf{I})$ with $0 < \beta_1, \dots \beta_T < 1$. The diffusion sampling hyperparameters for these experiments are listed in Table 2.

Image Datasets All experiments were run on a single NVIDIA H100 GPU. In these experiments, we used the probability flow ODE as the predictor by using pre-trained models (config F) from [Karras et al., 2022]. We used the same number of predictor steps (18 for CIFAR-10, 40 for FFHQ and AFHQv2, 256 for ImageNet). We use the default sampling strategy in their codebase, which uses the scale function $s(t) = 1$ and schedule $\sigma(t) = t$. This yields a simplified expression for the probability flow ODE (Equation 4 in [Karras et al., 2022])

$$\mathrm{d}\mathbf{Y}_t = \sigma(t_{\max} - t)\nabla_{\mathbf{x}} \log p_{t_{\max}-t}(\mathbf{Y}_t)\,\mathrm{d}t.$$

where the ODE runs forward in time (i.e. t goes from 0 to $t_{\max}$). The diffusion sampling hyperparameters for these experiments are listed in Table 3.

Table 2: Diffusion sampling hyperparameters for experiments on synthetic 2D datasets.

Dataset	Predictor Steps	MADM steps per Predictor Step	Step Size h at p_t
spiral	40	20	$10^{-1} \times \beta_{t+1}$
funnel	10	20	β_{t+1}
sierpinski triangle	20	30	$10^{-2} \times \beta_{t+1}$
pinwheel	20	30	$10^{-2} \times \beta_{t+1}$

Table 3: Diffusion sampling hyperparameters for experiments on image datasets. Run times are based on a batch size of 64.

Dataset	ODE Solver	Predictor Steps	MADM Steps	MADM Step Size h at p_t	Run Time
CIFAR-10	Heun	18	-	-	3 secs
CIFAR-10	Heun	18	1	$10^{-2} \times \sigma(t)$	3 secs
CIFAR-10	Euler	18	-	-	2 secs
CIFAR-10	Euler	18	1	$10^{-2} \times \sigma(t)$	3 secs
FFHQ	Heun	40	-	-	10 secs
FFHQ	Heun	40	1	$10^{-2} \times \sigma(t)$	24 secs
FFHQ	Euler	40	-	-	8 secs
FFHQ	Euler	40	1	$10^{-2} \times \sigma(t)$	20 secs
AFHQv2	Heun	40	-	-	10 secs
AFHQv2	Heun	40	1	$10^{-2} \times \sigma(t)$	24 secs
AFHQv2	Euler	40	-	-	8 secs
AFHQv2	Euler	40	1	$10^{-2} \times \sigma(t)$	20 secs
ImageNet	Heun	256	-	-	3 mins
ImageNet	Heun	256	1	$10^{-2} \times \sigma(t)$	4 mins
ImageNet	Euler	256	-	-	2 mins
ImageNet	Euler	256	1	$10^{-2} \times \sigma(t)$	3 mins

H Additional diagnostics on CIFAR-10

H.1 Conservativeness of learned score

Let $J_s = \nabla_x s_\theta$ and $K_s = (J_s - J_s^\top)/2$. We report

$$R_{\text{skew}} = 100 \frac{\|K_s\|_F^2}{\|J_s\|_F^2} = 50 \frac{\sum_{i<j} (\partial_i s_{\theta,j} - \partial_j s_{\theta,i})^2}{\|J_s\|_F^2}.$$

This is the percentage of Jacobian energy in the non-conservative component. A conservative score has $R_{\text{skew}} = 0$. We report this in Table 4. The score is therefore approximately conservative over the probed states: the skew component accounts for at most 0.66% of total Jacobian energy.

Table 4: Percentage of Jacobian energy in the non-conservative component. We estimate this without constructing the Jacobian, by using two Rademacher Jacobian-product probes at each of 16 states per noise level.

σ	57.6	5.32	0.585	0.0599	0.00753	0.002
R_{skew}	$1.7\times10^{-6}\%$	0.0030%	0.109%	0.214%	0.664%	0.398%

H.2 MADM robustness across proposal scales

Many efforts have sought to integrate MCMC-based corrections into continuous diffusion models without Metropolis correction, for instance more broadly see Holl et al. [2026]. However, dismissing discretisation error overlooks its severe vulnerability under scaling. Comparing the Unadjusted

Langevin Algorithm (ULA) and Metropolis-adjusted approaches like MADM across moderate and aggressive proposal scales illustrates this exact failure: while both methods perform similarly at moderate scales, a fourfold increase in the common scale causes ULA's performance to deteriorate sharply (from 1.97 to 7.27 in FID). Conversely, accept–reject adjustments (such as Algorithm 4) maintain robustness—retaining an FID of 1.99—by actively rejecting harmful proposals. This empirical contrast underscores the danger of glossing over discretisation bias, demonstrating that proper correction mechanisms are indispensable for maintaining structural stability and preventing severe performance collapse.

Table 5: Fréchet Inception Distance (FID) on CIFAR-10 sampled using the pre-trained model from Karras et al. [2022]. We consider the probability flow ODE (PF-ODE) predictor equipped with unadjusted Langevin algorithm (ULA) and Metropolis-Adjusted Diffusion Model (MADM) correctors. All FIDs are calculated using 50,000 samples. We highlight the best FID in bold.

PC Sampler/Proposal scale	2×10^{-2}	8×10^{-2}	16×10^{-2}
PF-ODE + ULA	1.97	7.27	27.03
PF-ODE + MADM (Alg. 4)	1.97	**1.99**	**1.99**

I Licenses

Codebases:

- Elucidating the Design Space of Diffusion-Based Generative Models [Karras et al., 2022]: Attribution-NonCommercial-ShareAlike 4.0 International.

Datasets:

- CIFAR-10 [Krizhevsky and Hinton, 2009]: MIT license.
- FFHQ [Karras et al., 2019]: Creative Commons BY-NC-SA 4.0 license.
- AFHQv2 [Choi et al., 2020]: MIT license.
- ImageNet [Deng et al., 2009]: Unknown license.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main theoretical and experimental contributions are clearly stated in the abstract and demonstrated in the paper. They reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in Section 7.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All proofs are provided in the technical appendices and appropriately referenced.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide full experiment details in Appendix G and the code to run our experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code needed to run our experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We provide full experiment details in Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Our main metric is FID score for which it is standard practice to report it calculated on the first 50,000 images generated from the model. Standard deviations could be bootstrapped from this set but this is not standard practice. Furthermore, the computational cost to sample large image models 50,000 times multiple times is prohibitive for our academic compute cluster.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide full details of compute resources in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: After careful review of the NeurIPS Code of Ethics, it is clear that the research conducted in this paper conforms with the Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper is mostly theoretical and methodological. We do not see immediate societal impact of our work. However, we acknowledge that our algorithm might suffer from the same societal biases as any other generative models. Indeed, it could improve the quality of generative models and hence, be used to generate deepfakes for disinformation.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets used in our experiments are open-source and have been properly referenced in the paper and Appendix I.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components. However, we acknowledge that LLMs were used to check the soundness of proofs and theorems.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.