
An Efficient Minibatch Acceptance Test for Metropolis-Hastings

Anonymous Author 1
Unknown Institution 1

Anonymous Author 2
Unknown Institution 2

Anonymous Author 3
Unknown Institution 3

Abstract

Markov chain Monte Carlo (MCMC) methods have many applications in machine learning. We are particularly interested in their application to modeling very large datasets, where it is impractical to perform Metropolis-Hastings tests on the full data. Previous work on reducing the cost of Metropolis-Hastings tests yield variable data consumed per sample, with only constant factor reductions versus using the full dataset for each sample. Here we present a method that can be tuned to provide arbitrarily small batch sizes, by adjusting either proposal step size or temperature. Our approach uses the natural noise present in minibatch likelihood estimates to furnish the randomness in a Metropolis-Hastings test. Our test uses the noise-tolerant Barker acceptance test with a novel additive correction variable. The resulting test can be combined with minibatch proposals to yield updates with the same complexity as a simple SGD update. In this paper we derive the test, analyze its performance, discuss its implementation, and present several experiments. We demonstrate several order-of-magnitude speedup over previous work, and show for the first time that small expected minibatch sizes are possible.

1 INTRODUCTION

Markov chain Monte Carlo (MCMC) sampling is a powerful method for computation on intractable distributions. We are interested in large dataset applications, where the goal is to sample a posterior distribution $p(\theta \mid x_1, \dots, x_N)$ of parameter θ for large N . The Metropolis-Hastings method (M-H) generates sample

candidates from a proposal distribution q which is in general different from the target distribution p , and decides whether to accept or reject based on an acceptance test. The acceptance test is usually a Metropolis test [Metropolis et al., 1953, Hastings, 1970].

Many state-of-the-art machine learning methods, and deep learning in particular, are based on minibatch updates (such as SGD) to a model. Minibatch updates produce many improvements to the model for each pass over the dataset, and have high sample efficiency. In contrast, conventional M-H requires calculations over the full dataset to produce a new sample. Recent results from Korattikara et al. [2014] and Bardenet et al. [2014] perform approximate (bounded error) acceptance tests using subsets of the full dataset. The amount of data consumed for each test varies significantly from one minibatch to the next. By contrast, Maclaurin and Adams [2014], Bardenet et al. [2015] perform exact tests but require a lower bound on parameter distribution across its domain. The amount of data reduction depends on the accuracy of this bound, and such bounds are only available for relatively simple distributions.

Here we derive a new test which incorporates the variability in minibatch statistics as *a natural part of the test* and requires less data per iteration than prior work. We use a Barker test function [Barker, 1965], which makes our test naturally error tolerant. The idea of using a noise-tolerant Barker’s test function was suggested but not explored empirically in Bardenet et al. [2015] section 6.3. But the asymptotic test statistic CDF and the Barker function are different, which leads to fixed errors for the approach in Bardenet et al. [2015]. Here, we show that the difference between the distributions can be corrected with an additive random variable. This leads to a test which is fast, and whose error can be made arbitrarily small.

Our test is applicable when the variance (over data samples) of the log acceptance probability is small enough (less than 1). It’s not clear at first why this quantity should be bounded, but we will show that it is “natural” for well-specified models running Metropolis-Hastings sampling with optimal proposals [Roberts and Rosenthal, 2001] on a full dataset. When we reduce the

amount of data for the test, the variance goes up. We have to reduce variance in one of several ways. Either:

- Increase the temperature of the target distribution. Log likelihoods scale as $1/T$, and so the variance of the likelihood ratio will vary as $1/T^2$. Our model is no longer well-specified (we are doing inference at a temperature different from that assumed during data generation), but higher temperature can be advantageous for parameter exploration.
- Increase the minibatch size when needed for certain minibatches. Log acceptance variance scales as $1/k$ vs the minibatch size k . Our test is adaptive like earlier works, but unlike them, the distribution of minibatch size is Gaussian, not long-tailed. Increased minibatch size also reduces the error rate for the test.
- For continuous probability distributions, reduce the proposal step size and variance (for stochastic proposals) compared to an optimal proposal. The variance of the log acceptance probability scales as the square of proposal step size.

It is worth discussing at this point the typical goals of M-H sampling on large datasets. By the Bernstein-von Mises Theorem, the posterior distribution for a Bayesian inference task has variance that scales inversely with the N data samples. Simply sampling from it is one application, but an efficient proposal [Roberts and Rosenthal, 2001] has similar variance to the target and will diffuse to it extremely slowly. For applications to neural networks or models where the posterior is multimodal [Choromanska et al., 2015], samplers will likely get trapped in one of the modes. A common solution is to anneal the sampler, running first at high temperature (scaling log likelihoods by $1/T$) to flatten the likelihood landscape. This in turn reduces the variance of the log acceptance probability and allows our acceptance test to be applied. Our samples can cover the search space densely with small steps rather than taking a few sparse steps towards an optimum. In this mode, Metropolis-Hastings can be used in similar fashion to Stochastic Gradient Descent. The goal in SGD is to make gradual progress to a posterior mode with each step, taking small steps so that the cumulative displacement has progressively lower variance.

The contributions of this paper are as follows:

- We develop a new, more efficient (in samples per test) minibatch acceptance test with quantifiable error bounds. The test uses a novel additive correction variable to implement a Barker test based on minibatch mean and variance.
- We compare performance of our new test and prior approaches on several datasets. We demonstrate

orders of magnitude improvements in efficiency (measured as data consumed per test), and that it does not suffer from long-tailed minibatch sizes.

2 PRELIMINARIES

In the Metropolis-Hastings method [Gilks and Spiegelhalter, 1996, Brooks et al., 2011], a difficult-to-compute probability distribution $p(\theta)$ is sampled using a Markov chain $\theta_1, \dots, \theta_n$. The sample θ_{t+1} at time $t+1$ is generated using a candidate θ' from a (simpler) proposal distribution $q(\theta' | \theta_t)$, filtered by an acceptance test. The acceptance test is usually a Metropolis test. The Metropolis test has acceptance probability:

$$\alpha(\theta_t, \theta') = \frac{p(\theta')q(\theta_t | \theta')}{p(\theta_t)q(\theta' | \theta_t)} \wedge 1 \quad (1)$$

where $a \wedge b$ denotes $\min(a, b)$. With probability $\alpha(\theta_t, \theta')$, we accept θ' and set $\theta_{t+1} = \theta'$, otherwise set $\theta_{t+1} = \theta_t$. The test is often implemented with an auxiliary random variable $u \sim \mathcal{U}(0, 1)$ with a comparison $u < \alpha(\theta_t, \theta')$; here, $\mathcal{U}(a, b)$ denotes the uniform distribution on the interval $[a, b]$. For simplicity, we drop the subscript t for the current sample θ_t and denote it as θ .

The acceptance test guarantees detailed balance, which means $p(\theta)p(\theta' | \theta) = p(\theta')p(\theta | \theta')$, where $p(\theta' | \theta)$ is the probability of a transition from state θ to θ' . Here, $p(\theta' | \theta) = q(\theta' | \theta)\alpha(\theta, \theta')$. This condition, together with ergodicity, guarantees that the Markov chain has a unique stationary distribution $\pi(\theta) = p(\theta)$. For Bayesian inference, we would like to sample from a parameter distribution for θ based on some observed data, so the target distribution is $p(\theta | x_1, \dots, x_N)$. The acceptance probability is now:

$$\alpha(\theta, \theta') = \frac{p_0(\theta') \prod_{i=1}^N p(x_i | \theta') q(\theta | \theta')}{p_0(\theta) \prod_{i=1}^N p(x_i | \theta) q(\theta' | \theta)} \wedge 1 \quad (2)$$

where $p_0(\theta)$ is a prior, and $p(x_i | \theta)$ are the probabilities of the observations. Computing samples this way requires the use of all N training data points, but this is very expensive for large datasets. To address this challenge, Korattikara et al. [2014], Bardenet et al. [2014] perform approximate Metropolis-Hastings tests using sequential hypothesis testing. During each iteration, they start with a small minibatch of data and test whether the sample θ' should be accepted based on an approximate version of the test $u < \alpha(\theta, \theta')$. If the approximate test cannot make a decision with sufficient confidence, then the minibatch size is increased and the test repeats. This process continues until a decision. The bounds depend on either an asymptotic Central Limit Theorem [Korattikara et al., 2014] or a concentration bound [Bardenet et al., 2014]. The latter requires direct bounds on the log likelihood ratio, which

for general distributions requires knowing $p(x_i | \theta)$ and $p(x_i | \theta')$ for all N samples. In addition, both methods suffer the drawback of resolving small log likelihood ratio differences between the minibatch and full batch versions. In the worst case, all N data points may be needed, as we show in Section 2.1.

Following Bardenet et al. [2014], we write the test $u < \alpha(\theta, \theta')$ equivalently as $\Lambda(\theta, \theta') > \psi(u, \theta, \theta')$, where¹

$$\Lambda(\theta, \theta') = \sum_{i=1}^N \log \left(\frac{p(x_i | \theta')}{p(x_i | \theta)} \right) \quad (3)$$

and $\psi(u, \theta, \theta') = \log \left(u \frac{q(\theta' | \theta) p_0(\theta)}{q(\theta | \theta') p_0(\theta')} \right).$

To reduce computational effort, an unbiased estimate of $\Lambda(\theta, \theta')$ based on a minibatch can be used:

$$\Lambda^*(\theta, \theta') = \frac{N}{b} \sum_{i=1}^b \log \left(\frac{p(x_i | \theta')}{p(x_i | \theta)} \right) \quad (4)$$

Finally, it will be convenient for our analysis to define $\Lambda_i(\theta, \theta') = N \log \left(\frac{p(x_i | \theta')}{p(x_i | \theta)} \right)$. Thus, $\Lambda(\theta, \theta')$ is the mean of $\Lambda_i(\theta, \theta')$ over the entire dataset, and $\Lambda^*(\theta, \theta')$ is the mean of $\Lambda_i(\theta, \theta')$ over its minibatch.

Since minibatches contains randomly selected samples x_i , the values Λ_i are i.i.d. random variables². By the Central Limit Theorem, we expect $\Lambda^*(\theta, \theta')$ to be approximately Gaussian. The acceptance test then becomes a statistical test of the hypothesis that $\Lambda(\theta, \theta') > \psi(u, \theta, \theta')$ by establishing that $\Lambda^*(\theta, \theta')$ is substantially larger than $\psi(u, \theta, \theta')$. In Korattikara et al. [2014] an asymptotic central limit argument was used to derive this gap, while in Bardenet et al. [2014] a concentration bound was used. In both cases, the resulting tests were shown to give useful reductions in number of samples required over using the full dataset, but there were no worst-case bounds, which we explore in the following section.

2.1 A Worst-Case Gaussian Example

Let x_1, \dots, x_N be i.i.d. $\mathcal{N}(\theta, 1)$ with known variance $\sigma^2 = 1$ and (unknown) mean $\theta = 0.5$. We use a uniform prior on θ . The log likelihood ratio is

$$\Lambda^*(\theta, \theta') = N(\theta' - \theta) \left(\frac{1}{b} \sum_{i=1}^b x_i - \theta - \frac{\theta' - \theta}{2} \right) \quad (5)$$

which is normally distributed over selection of the Normal samples x_i . Since the x_i have unit variance, their

¹Our definitions differ from those in Bardenet et al. [2014] by a factor of N to simplify our analysis later.

²The analysis assumes sampling with replacement although implementations on typical large datasets will approximate this by sampling without replacement.

mean has variance $1/b$, and the variance of $\Lambda^*(\theta, \theta')$ is $\sigma^2(\Lambda^*) = (\theta' - \theta)^2 N^2 / b$. In order to pass a hypothesis test that $\Lambda > \psi$, there needs to be a large enough gap (several $\sigma(\Lambda^*)$) between $\Lambda^*(\theta, \theta')$ and $\psi(u, \theta, \theta')$.

The posterior is a Gaussian centered on the sample mean μ , and with variance $1/N$ (i.e., $\mathcal{N}(\mu, 1/N)$). In one dimension, an efficient proposal distribution has the same variance as the target distribution [Roberts and Rosenthal, 2001], so we use a proposal based on $\mathcal{N}(\theta, 1/N)$. It is symmetric $q(\theta' | \theta) = q(\theta | \theta')$, and since we assumed a uniform prior, $\psi(u, \theta, \theta') = \log u$. Our worst-case scenario is specified in Lemma 1.

Lemma 1. *For the model in Section 2.1, there exists a fixed (independent of N) constant c such that with probability $\geq c$ over the joint distribution of (θ, θ', u) , the tests from Korattikara et al. [2014], Bardenet et al. [2014] consume all N samples.*

Proof. See Supplementary Material, Section A. \square

Similar results can be shown for other distributions and proposals by identifying regions in product space (θ, θ', u) such that the hypothesis test needs to separate nearly-equal values. It follows that the accelerated M-H tests from prior work require at least a constant fraction $\geq c$ in the amount of data consumed per test compared to full-data tests, so their speed-up is $\leq 1/c$.

The problem is that these methods use tail bounds to separate Δ away from zero, but for certain input/random u combinations, Δ can be arbitrarily close to zero. We will avoid this by using the *approximately normal* variation in Δ^* to *replace* the variation due to u .

2.2 MCMC Posterior Inference

There is a separate line of MCMC work drawing principles from statistical physics. By viewing random variables as particles in a system, one can apply Hamiltonian Monte Carlo (HMC) [Neal, 2010] methods which generate high acceptance *and* distant proposals when run on full batches of data. Recently Langevin Dynamics [Welling and Teh, 2011, Ahn et al., 2012] has been applied to Bayesian estimation on minibatches of data. This simplified dynamics uses local proposals and avoids M-H tests by using small proposal steps whose acceptance approaches 1 in the limit. However, the constraint on proposal step size is severe, and the state space exploration reduces to a random walk. Full minibatch HMC for minibatches was described in Chen et al. [2014] which allows momentum-augmented proposals with larger step sizes. However, step sizes are still limited by the need to run accurately without M-H tests. By providing an M-H test with similar cost to standard gradient steps, our work opens the door to

applying those methods with much more aggressive step sizes without loss of accuracy.

3 A NEW MH ACCEPTANCE TEST

3.1 Log-Likelihood Ratios

For our new M-H test, we denote the exact and approximate log likelihood ratios as Δ and Δ^* , respectively. First, Δ is defined as

$$\Delta(\theta, \theta') = \log \frac{p_0(\theta') \prod_{i=1}^N p(x_i | \theta') q(\theta | \theta')}{p_0(\theta) \prod_{i=1}^N p(x_i | \theta) q(\theta' | \theta)}, \quad (6)$$

where p_0, p , and q match the corresponding functions within Equation (2). We separate out terms dependent and independent of the data x as:

$$\begin{aligned} \Delta(\theta, \theta') &= \sum_{i=1}^N \log \frac{p(x_i | \theta')}{p(x_i | \theta)} - \psi(1, \theta, \theta') \\ &= \Lambda(\theta, \theta') - \psi(1, \theta, \theta'). \end{aligned} \quad (7)$$

A minibatch estimator of Δ , denoted as Δ^* , is

$$\begin{aligned} \Delta^*(\theta, \theta') &= \frac{N}{b} \sum_{i=1}^b \log \frac{p(x_i | \theta')}{p(x_i | \theta)} - \psi(1, \theta, \theta') \\ &= \Lambda^*(\theta, \theta') - \psi(1, \theta, \theta') \end{aligned} \quad (8)$$

Note that Δ and Δ^* are evaluated on the full dataset and a minibatch of size b respectively. The term N/b means $\Delta^*(\theta, \theta')$ is an unbiased estimator of $\Delta(\theta, \theta')$.

The key to our test is a smooth acceptance function. We consider functions other than the classical Metropolis test that satisfy the detailed balance condition needed for accurate posterior estimation. A class of suitable functions is specified as follows:

Lemma 2. *If $g(s)$ is any function such that $g(s) = \exp(s)g(-s)$, then the acceptance function $\alpha(\theta, \theta') \triangleq g(\Delta(\theta, \theta'))$ satisfies detailed balance.*

This result is used in Barker [1965] to define the Barker acceptance test. As a sanity check, choosing $g(s) = \exp(s) \wedge 1$ — a function satisfying the requirement of Lemma 2 — produces the classical Metropolis acceptance test $\alpha(\theta, \theta') = g(\Delta(\theta, \theta')) = \frac{p(\theta')q(\theta|\theta')}{p(\theta)q(\theta'|\theta)} \wedge 1$. In fact, $g(s) = \exp(s) \wedge 1$ is the optimal acceptance function in terms of acceptance rate, since it accepts with probability 1 for $\Delta > 0$.

3.2 Barker (Logistic) Acceptance Function

For our new MH test we use the Barker logistic [Barker, 1965] function: $g(s) = (1 + \exp(-s))^{-1}$. Straightforward arithmetic shows that it satisfies the condition

in Lemma 2. While it is slightly less efficient than the Metropolis test when used on the full dataset, we will see that its smoothness allows it to naturally tolerate substantial variance in its input argument. This in turn will lead to a much more efficient test on subsets of data.

Assume we begin with the current sample θ and a candidate sample θ' , and that $V \sim \mathcal{U}(0, 1)$ is a uniform random variable. We accept θ' if $g(\Delta(\theta, \theta')) > V$, and reject otherwise. Since $g(s)$ is monotonically increasing, its inverse $g^{-1}(s)$ is well-defined and unique. So an equivalent test is to accept θ' iff

$$\Delta(\theta, \theta') > X = g^{-1}(V) \quad (9)$$

where X is a random variable with the logistic distribution (its CDF is the logistic function). To see this notice that $\frac{dV}{dX} = g'$, that g' is the density corresponding to a logistic CDF, and finally that $\frac{dV}{dX}$ is the density of X . The density of X is symmetric, so we can equivalently test whether

$$\Delta(\theta, \theta') + X > 0 \quad (10)$$

for a logistic random variable X .

3.3 A Minibatch Acceptance Test

We now describe acceptance testing using the minibatch estimator $\Delta^*(\theta, \theta')$. From Equation (8), $\Delta^*(\theta, \theta')$ can be represented as a constant term plus the mean of b IID terms $\Lambda_i(\theta, \theta')$ of the form $N \log \frac{p(x_i | \theta')}{p(x_i | \theta)}$. As b increases, $\Delta^*(\theta, \theta')$ therefore has a distribution which approaches a normal distribution by the Central Limit Theorem. We now describe this using an asymptotic argument and defer specific bounds between the CDFs of $\Delta^*(\theta, \theta')$ and a Gaussian to Section 5.

In the limit, since Δ^* is normally distributed about its mean Δ , we can write

$$\Delta^* = \Delta + X_{\text{norm}}, \quad X_{\text{norm}} \sim \bar{\mathcal{N}}(0, \sigma^2(\Delta^*)), \quad (11)$$

where $\bar{\mathcal{N}}(0, \sigma^2(\Delta^*))$ denotes a distribution which is approximately normal with variance $\sigma^2(\Delta^*)$. But to perform the test in Equation (10) we want $\Delta + X$ for a logistic random variable X (call it X_{log} from now on). In Bardenet et al. [2015] it was proposed to use Δ^* in a Barker test anyway and tolerate the fixed error caused by this approximation.

Our approach is to instead decompose X_{log} as

$$X_{\text{log}} = X_{\text{norm}} + X_{\text{corr}}, \quad (12)$$

where we assume $X_{\text{norm}} \sim \mathcal{N}(0, \sigma^2)$ and that X_{corr} is a zero-mean “correction” variable with density $C_\sigma(X)$. The two variables are added (i.e., their distributions convolve) to form X_{log} . This decomposition requires an

appropriate X_{corr} distribution. We show in Section 4 that it is possible to use deconvolution to derive a highly accurate representation of $C_\sigma(X)$. Using X_{corr} samples from $C_\sigma(X)$, the acceptance test is now

$$\Delta + X_{\text{log}} = (\Delta + X_{\text{norm}}) + X_{\text{corr}} = \Delta^* + X_{\text{corr}} > 0. \quad (13)$$

Therefore, assuming the variance of Δ^* is small enough, if we have an estimate of Δ^* from the current data minibatch, we test acceptance by adding a random variable X_{corr} and then accept θ' if the result is positive (and reject otherwise).

If $\tilde{\mathcal{N}}(0, \sigma^2(\Delta^*))$ is exactly $\mathcal{N}(0, \sigma^2(\Delta^*))$, the above test is exact, and as we show in Section 5, if there is a maximum error ϵ between the CDF of $\tilde{\mathcal{N}}(0, \sigma^2(\Delta^*))$ and the CDF of $\mathcal{N}(0, \sigma^2(\Delta^*))$, then our test has an error of at most ϵ relative to the full batch version.

4 COMPUTING THE CORRECTION DISTRIBUTION

Our proposed test in Equation (13) requires knowing the distribution of the correction variable X_{corr} such that $X_{\text{norm}} + X_{\text{corr}} = X_{\text{log}}$, where $X_{\text{norm}} \sim \mathcal{N}(0, \sigma^2)$ and X_{log} has a standard logistic CDF, $(1 + \exp(-X))^{-1}$. In Section 5, we show that the accuracy of the test depends on the absolute error between the CDFs of $X_{\text{norm}} + X_{\text{corr}}$ and X_{log} . Consequently, we need to minimize this in our construction of X_{corr} . More formally, let $\Phi_{s_X} = \Phi(X/s_X)$ where Φ is the standard normal CDF³, $S(X)$ be the logistic function, and $C_\sigma(X)$ be the *density* of the correction X_{corr} distribution. Our goal, based on Equation (12), is to solve the following optimization problem:

$$C_\sigma^* = \arg \min_{C_\sigma} |\Phi_\sigma * C_\sigma - S| \quad (14)$$

where $*$ denotes convolution.

For computation of C_σ , we assume that its input Y and another variable X lie in the intervals $[-V, V]$ and $[-2V, 2V]$, respectively. We discretize the convolution by discretizing X and Y into $4N + 1$ and $2N + 1$ values respectively. If $i \in \{-2N, \dots, 2N\} = \mathcal{I}$ and $j \in \{-N, \dots, N\} = \mathcal{J}$, then we can write $X_i = i(V/N)$ and $Y_j = j(V/N)$, and the objective can be written as:

$$C_\sigma^* = \arg \min_{C_\sigma} \max_{i \in \mathcal{I}} \left| \sum_{j \in \mathcal{J}} \Phi_\sigma(X_i - Y_j) C_\sigma(Y_j) - S(X_i) \right|.$$

We now define a matrix M and vectors u and v such that $M_{ij} = \Phi_\sigma(X_i - Y_j)$, $u_j = C_\sigma(Y_j)$, and $v_i =$

³Hence, Φ_{s_X} is the CDF of a zero-mean Gaussian with standard deviation s_X .

Input : Number of samples T , minibatch size m , error bound δ , pre-computed correction $C_1(X)$ distribution, initial sample θ_1 .
Output : A chain of T samples $\{\theta_1, \dots, \theta_T\}$ from $p(\theta)$;
for $t = \{1, \dots, T\}$ **do**
 -Propose a candidate θ' from proposal $q(\theta' | \theta_t)$;
 -Draw a minibatch of m points x_i , compute $\Delta^*(\theta_t, \theta')$ and sample variance $s_{\Delta^*}^2$;
 -Estimate moments $E|\Lambda_i - \Lambda|$ and $E|\Lambda_i - \Lambda|^3$ from the sample, and error ϵ from Corollary 1;
while $s_{\Delta^*}^2 \geq 1$ **or** $\epsilon > \delta$ **do**
 | -Draw m more samples to augment the minibatch, update Δ^* , $s_{\Delta^*}^2$ and ϵ estimates;
end
 -Draw $X_{\text{nc}} \sim \mathcal{N}(0, 1 - s_{\Delta^*}^2)$ and X_{corr} from the correction distribution $C_1(X)$;
if $\Delta^* + X_{\text{nc}} + X_{\text{corr}} > 0$ **then**
 | -Accept the candidate, $\theta_{t+1} = \theta'$;
else
 | -Reject and re-use the old sample, $\theta_{t+1} = \theta_t$;
end
end

Algorithm 1: Our acceptance test for MCMC.

Table 1: Error in $X_{\text{norm}} + X_{\text{corr}}$ versus X_{log}

N	σ	λ	L_∞ error
4000	0.9	1	1.0e-4
4000	0.8	0.03	5.0e-6

$S(X_i)$, where the indices i and j are appropriately translated to be non-negative indices for M , u , and v . Thus, the problem is now to minimize $\|Mu - v\|_\infty$ with the constraint that $u > 0$ since it represents a density. We approximate this with a least squares solution:

$$u^* = \arg \min_u \|Mu - v\|_2^2 + \lambda \|u\|_2^2, \quad (15)$$

with regularization λ . The solution is well-known from the normal equations ($u^* = (M^T M + \lambda I)^{-1} M^T v$) and in practice yields an acceptable L_∞ norm.

With this approach, there is no guarantee that $u^* \geq 0$. However, we have some flexibility in the choice of σ in Equation (14). As we decrease the variance of X_{norm} , the variance of X_{corr} grows by the same amount and is in fact the result of convolution with a Gaussian whose variance is the difference. Thus as σ decreases, $C_\sigma(X)$ grows and approaches the derivative of a logistic function at $\sigma = 0$. It retains some very weak negative values for $\sigma > 0$ but removal of those values leads to very small error. Table 1 shows that the errors between $X_{\text{norm}} + X_{\text{corr}}$ and X_{log} can be made very small, approaching single floating precision (about 10^{-7}).

Algorithm 1 describes our procedure. A few points:

- It uses an adaptive step size so as to use the smallest possible average minibatch size. Unlike previous work however (and as we show in Section 6) the size distribution is short-tailed.
- An additional normal variable X_{nc} is added to Δ^* to produce a variable with unit variance. This is not mathematically necessary, but allows us to use a single correction distribution C_1 with $\sigma = 1$ for X_{corr} , saving on memory footprint.
- The sample variance $s_{\Delta^*}^2$ is proportional to $\|\theta' - \theta\|_2^2$ whose distribution for Normal proposals is the square of a normal variable.

5 ANALYSIS

We now derive error bounds for our M-H test, and for the approximate target distribution that it generates. From Table 1, we know that it is possible to generate the correction samples X_{corr} with a CDF error approaching single-precision floating point error. We therefore treat X_{corr} as a sample from the exact correction distribution and we will not analyze its errors.

In the most similar prior works, Korattikara et al. [2014] uses asymptotic arguments based on the CLT to argue that its approximate acceptance test error tends to zero as batch size increases, but no quantitative bounds are given. In Bardenet et al. [2014], explicit bounds are given, but they depend on bounding:

$$C_{\theta, \theta'} = \max_{1 \leq i \leq N} |\log p(x_i | \theta') - \log p(x_i | \theta)|. \quad (16)$$

Such bounds can be derived efficiently for models such as logistic regression, but it is unclear how to derive them for a complex model such as a neural network. In general, since a new θ' value is obtained each iteration, one would need to iterate through all the $p(x_i | \theta')$ terms, defeating the purpose of minibatch MCMC. In contrast, we use quantitative forms of the Central Limit Theorem which rely on measurable statistics from a single minibatch. Thus a sampler using our approach does not need to see data beyond the current minibatch.

In Section 5.1, we present bounds on the absolute and relative error (in terms of the CDFs) of the distribution of Δ^* vs. a Gaussian. We then show in Section 5.2 that these bounds are preserved after the addition of other random variables (e.g., X_{nc} and X_{corr}). It then follows that the acceptance test has the same error bound.

5.1 Bounding the Error of Δ^* from Gaussian

We use the following quantitative central-limit result:

Lemma 3. *Let X_1, \dots, X_n be a set of zero-mean, independent, identically-distributed random variables with*

sample mean \bar{X} and sample variance s_X^2 where:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i, \quad s_X = \frac{1}{n} \left(\sum_{i=1}^n (X_i - \bar{X})^2 \right)^{\frac{1}{2}}. \quad (17)$$

This implies $t = \bar{X}/s_X$ has an approximate Student's distribution which approaches a normal distribution in the limit. Then

$$\sup_x |\Pr(t < x) - \Phi(x)| \leq \frac{6.4E|X|^3 + 2E|X|}{\sqrt{n}}. \quad (18)$$

Proof. The following bound is given immediately after Corollary 2 from Novak [2005]:

$$-6.4E|X|^3 - 2E|X| \leq \sup_x |\Pr(t < x) - \Phi(x)|\sqrt{n} \leq 1.36E|X|^3.$$

This bound applies to $x \geq 0$. Applying the bound to $-x$ when $x < 0$ and combining with $x > 0$, we obtain the weaker but unqualified bound in Equation (18). \square

Lemma 3 demonstrates that as long as we know the first and third absolute moments $E|X|$ and $E|X|^3$, we can bound the error of the normal approximation, which decays as $O(n^{-\frac{1}{2}})$. Making the change of variables $y = xs_X$, Equation (18) becomes

$$\sup_y \left| \Pr(\bar{X} < y) - \Phi\left(\frac{y}{s_X}\right) \right| \leq \frac{6.4E|X|^3 + 2E|X|}{\sqrt{n}} \quad (19)$$

showing that the distribution of \bar{X} approaches the normal distribution $\mathcal{N}(0, s_X)$ whose variance is s_X , measured from the sample.

To apply this to our test, let $X_i = \Lambda_i(\theta, \theta') - \Lambda(\theta, \theta')$, so that the X_i are zero-mean, i.i.d. variables. If instead of all n samples, we only extract a subset of b samples corresponding to our minibatch, we can connect \bar{X} with our Δ^* term:

$$\bar{X} = \Delta^*(\theta, \theta') - \Delta(\theta, \theta'), \quad (20)$$

so that $s_X = s_{\Delta^*}$. This results in the following:

Corollary 1. *We can now substitute into Equation (19) and displace by the mean, giving:*

$$\begin{aligned} \sup_y \left| \Pr(\Delta^* < y) - \Phi\left(\frac{y - \Delta}{s_{\Delta^*}}\right) \right| &\leq \frac{6.4E|X|^3 + 2E|X|}{\sqrt{b}} \\ &= \epsilon(\theta, \theta', b). \end{aligned}$$

Corollary 1 shows that the distribution of Δ^* approximates a Normal distribution with mean Δ and variance $s_{\Delta^*}^2$. Furthermore, it bounds the error with *estimable quantities*: both $E|X|$ and $E|X|^3$ can be estimated

as means of $|\Lambda_i - \Lambda|$ and $|\Lambda_i - \Lambda|^3$, respectively, on each minibatch. We expect this will often be accurate enough on minibatches with hundreds or thousands of points, but otherwise bootstrap CIs can be computed from those sequences. Since the bounds are monotone in $E|X|$ and $E|X|^3$, using upper bootstrap CI limits will provide high-confidence error bounds.

5.2 Error Bounds are Preserved After Adding Random Variables

We next relate the CDFs of distributions and show that bounds are preserved after adding random variables.

Lemma 4. *Let $P(x)$ and $Q(x)$ be two cumulative distributions satisfying $\sup_x |P(x) - Q(x)| \leq \epsilon$ with x in some real range. Let $R(y)$ be the density of another random variable y . Let P' be the convolution $P * R$ and Q' be the convolution $Q * R$. Then $P'(z)$ (resp. $Q'(z)$) is the CDF of sum $z = x + y$ of independent random variables x with CDF $P(x)$ (resp. $Q(x)$) and y with density $R(y)$. Then*

$$\sup_x |P'(x) - Q'(x)| \leq \epsilon \quad (21)$$

Proof. See Supplementary Material, Section B. \square

From Lemma 4, we have the following Corollary:

Corollary 2. *If $\sup_y |\Pr(\Delta^* < y) - \Phi(\frac{y-\Delta}{s_{\Delta^*}})| \leq \epsilon(\theta, \theta', b)$, then*

$$\sup_y |\Pr(\Delta^* + X_{\text{nc}} + X_{\text{corr}} < y) - S(y - \Delta)| \leq \epsilon(\theta, \theta', b)$$

where $S(x)$ is the standard logistic function, and X_{nc} and X_{corr} are generated as per Algorithm 1.

Proof. See Supplementary Material, Section C. \square

From Section 3, as the distribution of Δ^* approaches a Gaussian, our new MH test becomes more accurate. Corollary 2 shows that the bounds from Section 5.1 are preserved after the addition of the random variables we use, showing that our test should remain accurate.

In fact we can do better than $O(n^{-1/2})$ error for the quantitative CLT (showing the error decreases as $O(n^{-1})$) by using a more precise limit distribution under an additional assumption. We review the details in the Supplementary Material, Section D, and leave the implementation details to future work.

6 EXPERIMENTS

We conduct two sets of experiments to explore the benefits of our minibatch MH test and to benchmark it with

previous work. In Section 6.1, we show that our test enables samples to converge to the posterior distribution of a heated Gaussian mixture model. In Section 6.2, we analyze its efficiency on logistic regression.

6.1 Mixture of Gaussians

We start with a simple Gaussian mixture model, borrowing an experiment from Welling and Teh [2011]. The parameter is 2-D, $\theta = \langle \theta_1, \theta_2 \rangle$, and the parameter/data generation process is

$$\begin{aligned} \theta &\sim \mathcal{N}(0, \text{diag}(\sigma_1^2, \sigma_2^2)) \\ x_i &\sim 0.5 \cdot \mathcal{N}(\theta_1, \sigma_x^2) + 0.5 \cdot \mathcal{N}(\theta_1 + \theta_2, \sigma_x^2). \end{aligned} \quad (22)$$

We set $\sigma_1^2 = 10$, $\sigma_2^2 = 1$ and $\sigma_x^2 = 2$. We fix $\theta = \langle 0, 1 \rangle$. The original paper sampled 100 data points from this distribution, and estimated the posterior in the parameters. We are interested in performance on larger problems and so sampled 1,000,000 points to form the posterior of $p(\theta) \prod_{i=1}^{1,000,000} p(x_i | \theta)$ with the same prior from Equation (22). This produces a much sharper posterior with two very narrow peaks. Our goal is to reproduce the original posterior, so we adjust the temperature to $T = 10,000$. Taking logs, we get the target as shown in the far left of Figure 1.

We run MCMC with our MH test and benchmark with Korattikara et al. [2014] and Bardenet et al. [2014], all with initial minibatch size $m = 100$. For Korattikara et al. [2014], we increment minibatches by 100 and for Bardenet et al. [2014], we increase the minibatch size geometrically with a ratio of $\gamma = 1.5$. These methods have different approaches to repeated testing: the former uses an aggressive testing approach, explicitly modeling correlation between tests and allowing many more tests for a given δ , while the latter uses discounting, allocating a fraction of the test error δ to each minibatch test.

The tolerance for making a decision in Korattikara et al. [2014] is $\epsilon = 0.005$, and the error bound control parameters in Bardenet et al. [2014] are $p = 2$ and $\delta = 0.01$. All methods (including ours) use the same random walk proposer with covariance $\Sigma = \text{diag}(0.15, 0.15)$. Since the proposer generates a random walk, all shaping of distribution of the samples is due to the MH test. All methods are run 5000 times to collect 5000 samples.

Figure 1 shows scatter plots of the resulting θ samples for the three methods, with darker regions indicating a greater density of points. There are no obvious differences, so we measure the similarity between each set of samples and the actual posterior.

We discretize the posterior coordinates into bins (non-overlapping ranges) with respect to the two components of the parameter θ . The probability of a sample

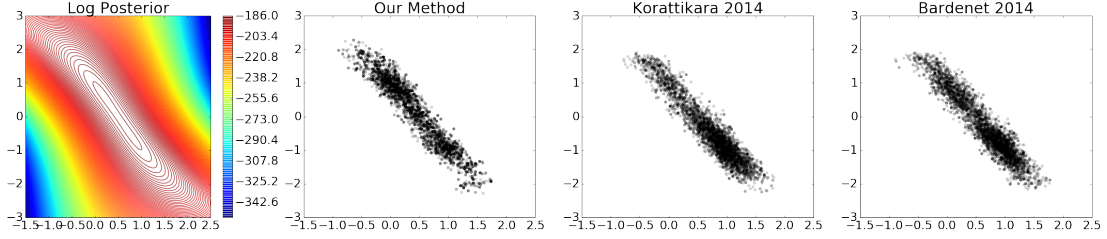


Figure 1: The log posterior contours and scatter plots of sampled θ values using different methods.

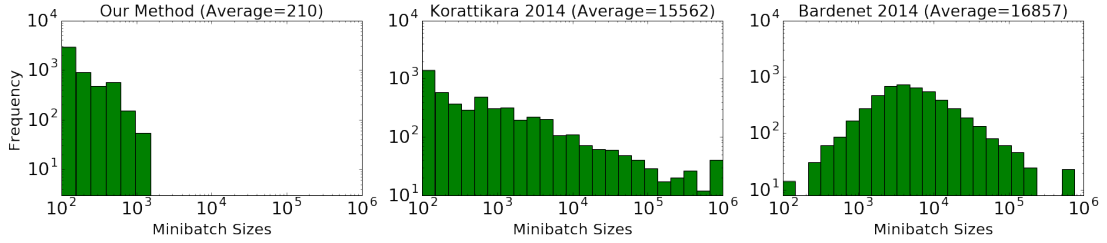


Figure 2: Minibatch sizes used in Section 6.1's experiment. The axes have the same (log-log scale) range.

falling into bin i (denoted as P_i) is the integral of the true posterior probability over the area of that bin. A single sample from any of the MH methods should therefore be multinomial with distribution P , and a set of n (ideally independent) samples should follow a $\text{Multinomial}(P, n)$ distribution. Since the ideal distribution is simple, we can use it to measure the quality of the sample distributions rather than use general purpose tests like KL-divergence or likelihood-ratio. The latter are problematic with zero counts in some bins as we have here.

In practice, it is difficult to compute multinomial probabilities for large n , in that case the per-bin distributions are well-approximated by Poisson distributions with parameter $\lambda_i = P_i n$. Given a set of samples $\{\theta_1, \dots, \theta_T\}$, let c_j denote the number of individual samples θ_i that fall in bin j out of N_{bins} total. Then we have

$$\log p(c_1, \dots, c_{N_{\text{bins}}} \mid P_1, \dots, P_{N_{\text{bins}}}) = \sum_{j=1}^{N_{\text{bins}}} c_j \log(nP_j) - nP_j - \log(\Gamma(c_j + 1)). \quad (23)$$

The results for each model are shown in Table 2.

While the Poisson results provide general guidance on the relative accuracy of the models, it is difficult to interpret the scores, so we perform significance tests to show the difference between the MCMC-sampled distributions and the ground-truth. We perform an approximate test using the Chi-Squared distribution as the test statistic, which we also show in Table 2.

Both the Poisson values and the Chi-Squared test statistic result in similar conclusions: our method is slightly superior to Korattikara et al. [2014], but slightly worse

than Bardenet et al. [2014]. We now show, however, that minibatch size results suggest that our method dominates in terms of speed and efficiency.

Figure 2 shows a histogram of the final minibatch sizes used by the three methods in each iteration. Our method consumes significantly less data; most minibatch sizes are smaller than 1000, and the average size is 210. The other methods occasionally need to consume a large proportion of the entire data set of one million elements; the average minibatch sizes are 15562 and 16857 for Korattikara 2014 and Bardenet 2014, respectively. The average minibatch sizes accurately predict the running times of these methods since all methods have a running time proportional to the total data consumed, with one caveat: the method of Bardenet et al. [2014] requires a global bound on the term $C_{\theta, \theta'}$ (see Equation (16)) over the entire the data set at every iteration in order to calculate the error bound. This bound must be either computed analytically using properties of the specific dataset, or it must be checked at each sample if a black-box test is required. We found that the sample code provided by the authors of Bardenet et al. [2014] in fact computed this bound explicitly for each sample generated, i.e. the implementation traversed the entire dataset for each sample, providing no performance advantage over the complete test. For this reason, we omitted the time to compute $C_{\theta, \theta'}$ from our measurements of the running time of the sample implementation of Bardenet et al. [2014], and consider only the time to process the minibatches.

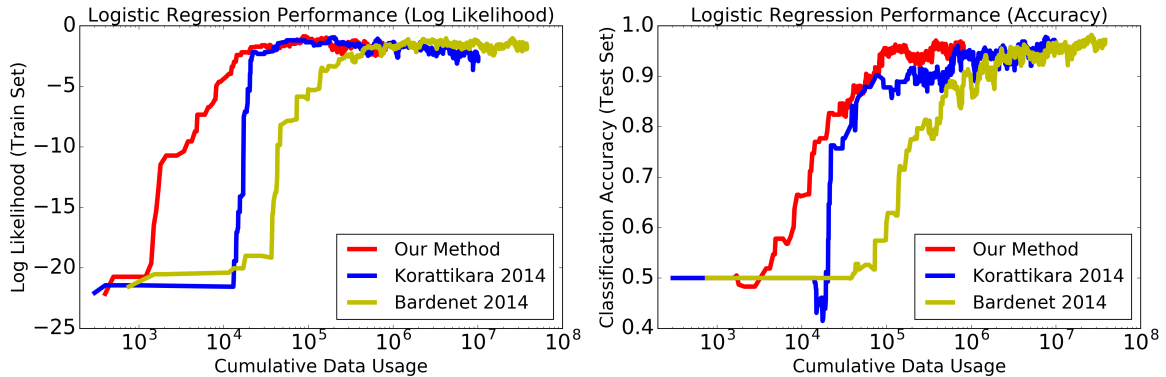


Figure 3: Logistic regression performance (accuracy/log likelihood) based on cumulative data usage.

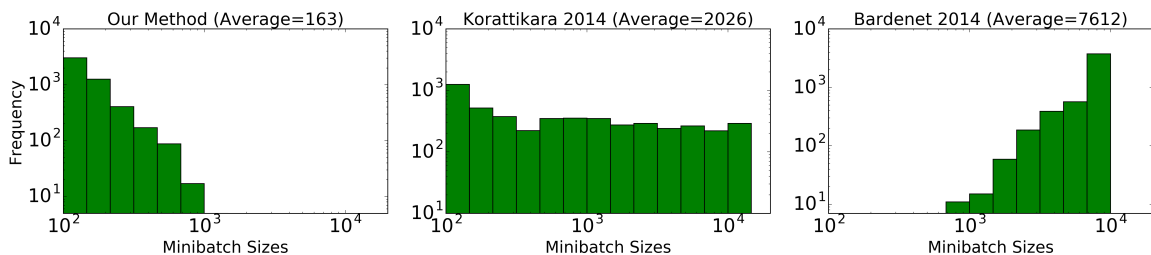


Figure 4: Counts of minibatch sizes in the MNIST logistic regression experiment (analogous to Figure 2).

Table 2: Gaussian Mixture Model Statistics

Metric	Ours	Korat.'14	Barde.'14
Equation 23	-1430.0	-1578.9	-1232.7
Chi-Squared	3313.9	3647.7	2444.1

6.2 Logistic Regression

We next use logistic regression for the binary classification of 1s versus 7s in the MNIST dataset [LeCun and Cortes]. The data has 12007 training and 1000 testing elements (we used a random subset of the test data). The pixel values are scaled to be in $(0, 1)$. We impose a flat prior on θ . For the proposer, we again use a random walk, this time with covariance matrix $0.05I$ for the 784×784 identity matrix I . The posterior temperature is set at $T = 1000$. We run our MH test and compare with Korattikara 2014 and Bardenet 2014; minibatch sizes and other parameters are set to be the same value as with the Gaussian Mixtures experiment from Section 6.1.

Figure 3 shows the training log likelihood and the testing prediction accuracy, each as a function of the cumulative training data points processed.⁴ To gener-

ate the curves, for each of the 5000 sampled vectors θ_t , $t \in \{1, \dots, 5000\}$, we use θ_t as the parameter for logistic regression. We see that our minibatch MH test is more efficient compared to the other two methods, because it achieves similar or superior performance while consuming fewer data points.

Figure 4, in a similar manner as Figure 2, shows the histogram of minibatch sizes for all three methods on a log-log scale. With an initial minibatch size of 100, Korattikara 2014 and Bardenet 2014 achieve average minibatch size of 2026 and 7612 respectively over the MNIST classification task, while our method achieves that of 163, showing significantly better performance than the benchmark test methods.

7 CONCLUSIONS

In this paper, we have derived a new MH test for minibatch MCMC methods. We demonstrated how a simple deconvolution process allows us to use a minibatch approximation to the full data tests. We experimentally show the benefits of our test on Gaussian mixtures and logistic regression. Straightforward directions for future work include running more experiments with a particular focus on investigation of the variance precondition. More elaborate extensions include testing on deep neural networks and combining our results with Hamiltonian Monte Carlo methods, providing a recipe

⁴Note that the curves do not span the same length over the x-axis, because our test consumes fewer samples throughout the MCMC procedure, so the corresponding curve will “end” before the other two.

for how to use our algorithm (following the framework of Ma et al. [2015]), or integrating parallel MCMC [Angelino et al., 2014, Ahn et al., 2014] concepts.

References

- S. Ahn, A. K. Balan, and M. Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.
- S. Ahn, B. Shahbaba, and M. Welling. Distributed stochastic gradient MCMC. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.
- E. Angelino, E. Kohler, A. Waterland, M. Seltzer, and R. P. Adams. Accelerating MCMC via parallel predictive prefetching. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.
- R. Bardenet, A. Doucet, and C. Holmes. Towards scaling up markov chain monte carlo: an adaptive subsampling approach. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.
- R. Bardenet, A. Doucet, and C. Holmes. On markov chain monte carlo methods for tall data. *arXiv preprint arXiv:1505.02827*, 2015.
- A. A. Barker. Monte-carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18:119–133, 1965.
- V. Bentkus, F. Gotze, and W.R.vanZwet. An edgeworth expansion for symmetric statistics. *Annals of Statistics*, 25(2), 1997.
- S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.
- T. Chen, E. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.
- A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surface of multilayer networks. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics, AISTATS*, 2015.
- W. Gilks and D. Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman & Hall/CRC, 1996.
- W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57: 97–109, 1970.
- A. Korattikara, Y. Chen, and M. Welling. Austerity in MCMC land: Cutting the metropolis-hastings budget. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.
- Y. LeCun and C. Cortes. MNIST handwritten digit database. URL <http://yann.lecun.com/exdb/mnist/>.
- Y. Ma, T. Chen, and E. Fox. A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems 28*, 2015.
- D. Maclaurin and R. P. Adams. Firefly monte carlo: Exact MCMC with subsets of data. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21, 1953.
- R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.
- Y. Novak. On self-normalized sums and students statistic. *Theory of Probability and its Applications*, 49(2): 336–344, 2005.
- G. O. Roberts and J. S. Rosenthal. Optimal scaling for various metropolishastings algorithms. *Statistical Science*, 16(4):351367, 2001.
- M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

Supplementary Material

A Proof of Lemma 1

Choose $(\theta' - \theta) \in \pm \frac{1}{\sqrt{N}}[0.5, 1]$ (event 1) and $(\theta - 0.5) \in \pm \frac{1}{\sqrt{N}}[0.5, 1]$ filtered for matching sign (event 2). As discussed in Lemma 1, both $q(\theta' | \theta)$ and $p(\theta | x_1, \dots, x_N)$ have variance $1/N$. If we denote Φ as the CDF of the standard normal distribution, then the former event occurs with probability $p_0 = 2(\Phi(\sqrt{N} \frac{1}{\sqrt{N}}) - \Phi(\sqrt{N} \frac{0.5}{\sqrt{N}})) = 2(\Phi(1) - \Phi(0.5)) \approx 0.2997$. The latter event, because we restrict signs, occurs with probability $p_1 = \Phi(1) - \Phi(0.5) \approx 0.14988$.

These events together guarantee that $\Lambda^*(\theta, \theta')$ is negative by inspection of equation (25) below. This implies that we can find a $u \in (0, 1)$ so that $\psi(u, \theta, \theta') = \log u < 0$ equals $E[\Lambda^*(\theta, \theta')]$. Specifically, choose u_0 to satisfy $\log u_0 = E[\Lambda^*(\theta, \theta')]$. Using $E[x_i] = 0.5$ and Equation (5), we see that

$$\log u_0 = N(\theta' - \theta) \frac{1}{b} \cdot E \left[\sum_{i=1}^b x_i - \theta - \frac{\theta' - \theta}{2} \right] \quad (24)$$

$$\log u_0 = -N(\theta' - \theta) \left(\theta - 0.5 + \frac{\theta' - \theta}{2} \right). \quad (25)$$

Next, consider the minibatch acceptance test $\Lambda^*(\theta, \theta') \not\approx \psi(u, \theta, \theta')$ used in Korattikara et al. [2014] and Bardenet et al. [2014], where $\not\approx$ means “significantly different from” under the distribution over samples of x_i . This turns out to be

$$\begin{aligned} \Lambda^*(\theta, \theta') &\not\approx \psi(u_0, \theta, \theta') \\ \iff N(\theta' - \theta) \cdot \frac{1}{b} \sum_{i=1}^b x_i - \theta - \frac{\theta' - \theta}{2} &\not\approx \log u_0 \\ \iff \frac{1}{b} \sum_{i=1}^b x_i - \left(\theta + \frac{\theta' - \theta}{2} + \frac{\log u_0}{N(\theta' - \theta)} \right) &\not\approx 0 \\ \iff \frac{1}{b} \sum_{i=1}^b x_i - 0.5 &\not\approx 0. \end{aligned} \quad (26)$$

Since the x_i have mean 0.5, the resulting test with our chosen u_0 will never correctly succeed and must use all N data points. Furthermore, if we sample values of u near enough to u_0 , the terms in parenthesis will not be sufficiently different from 0.5 to allow the test to succeed.

The choices above for θ and θ' guarantee that

$$\log u_0 \in -[0.5, 1][0.75, 1.5] = [-1.5, -0.375]. \quad (27)$$

Next, consider the range of u values near u_0 :

$$\log u \in \log u_0 + [-0.5, 0.375]. \quad (28)$$

The size of the range in u is at least $\exp([-2, -1.125]) \approx [0.13534, 0.32465]$ and occurs with probability at least $p_2 = 0.18932$. With u in this range, we rewrite the test as:

$$\frac{1}{b} \sum_{i=1}^b x_i - 0.5 \not\approx \frac{\log u/u_0}{N(\theta' - \theta)} \quad (29)$$

so that, as in Equation (26), the LHS has expected value zero. Given our choice of intervals for the variables, we can compute the range for the right hand side (RHS) assuming⁵ that $\theta' - \theta > 0$:

$$\begin{aligned} \min\{\text{RHS}\} &= \frac{-0.5}{\sqrt{N} \cdot 0.5} = -\frac{1}{\sqrt{N}} \\ \text{and } \max\{\text{RHS}\} &= \frac{0.375}{\sqrt{N} \cdot 0.5} = \frac{0.75}{\sqrt{N}} \end{aligned} \quad (30)$$

Thus, the RHS is in $\frac{1}{\sqrt{N}}[-1, 0.75]$. The standard deviation of the LHS given the interval constraints is at least $0.5/\sqrt{b}$. Consequently, the gap between the LHS and RHS in Equation (29) is at most $2\sqrt{b/N}$ standard deviations, limiting the range in which the test will be able to “succeed” without requiring more samples.

The samples θ , θ' and u are drawn independently and so the probability of the conjunction of these events is $c = p_0 p_1 p_2 = 0.0085$.

B Proof of Lemma 4

We first observe that

$$P'(z) - Q'(z) = \int_{-\infty}^{+\infty} (P(z-x) - Q(z-x))R(x)dx,$$

and since $\sup_x |P(x) - Q(x)| \leq \epsilon$ it follows that for all z :

$$\begin{aligned} -\epsilon &= \int_{-\infty}^{+\infty} -\epsilon R(x)dx \\ &\leq \int_{-\infty}^{+\infty} (P(z-x) - Q(z-x))R(x)dx \\ &\leq \int_{-\infty}^{+\infty} \epsilon R(x)dx = \epsilon, \end{aligned}$$

as desired.

C Proof of Corollary 2

We apply Lemma 4 twice. First take:

$$\begin{aligned} P(y) &= \Pr(\Delta^* < y) \\ \text{and } Q(y) &= \Phi\left(\frac{y - \Delta}{s_{\Delta^*}}\right) \end{aligned} \quad (31)$$

⁵If $\theta' - \theta < 0$, then the range would be $\frac{1}{\sqrt{N}}[-0.75, 1]$ but this does not matter for the purposes of our analysis.

and convolve with the distribution of X_n which has density $\phi(X/\sigma_n)$ where $\sigma_n^2 = 1 - s_{\Delta^*}^2$. This yields the next iteration of P and Q :

$$\begin{aligned} P'(y) &= \Pr(\Delta^* + X_{\text{nc}} < y) \\ \text{and } Q'(y) &= \Phi(y - \Delta) \end{aligned} \quad (32)$$

Now we convolve with the distribution of X_{corr} :

$$\begin{aligned} P''(y) &= \Pr(\Delta^* + X_{\text{nc}} + X_{\text{corr}} < y) \\ \text{and } Q''(y) &= S(y - \Delta) \end{aligned} \quad (33)$$

Both steps preserve the error bound $\epsilon(\theta, \theta', b)$. Finally $S(y - \Delta)$ is a logistic CDF centered at Δ , and so $S(y - \Delta) = \Pr(\Delta + X_{\log} < y)$ for a logistic random X_{\log} . We conclude that the probability of acceptance for the actual test $\Pr(\Delta^* + X_{\text{nc}} + X_{\text{corr}} > 0)$ differs from the exact test $\Pr(\Delta + X_{\log} > 0)$ by at most ϵ .

D Improved Error Bounds Based on Skew Estimation

We demonstrate how we can show $O(n^{-1})$ error for the quantitative CLT by using a more precise limit distribution under an additional assumption. Let μ_i denote the i^{th} moment, and b_i denote the i^{th} absolute moment of X . If Cramer's condition holds:

$$\lim_{t \rightarrow \infty} \sup |E(\exp(itX))| < 1, \quad (34)$$

then Equation 2.2 in Bentkus et al.'s work on Edgeworth expansions [Bentkus et al., 1997] provides the following:

Lemma 5. *Let X_1, \dots, X_n be a set of zero-mean, independent, identically-distributed random variables with sample mean \hat{X} and with t defined as in Lemma 3. If X satisfies Cramer's condition, then*

$$\sup_x \left| \Pr(t < x) - G\left(x, \frac{\mu_3}{b_2^{3/2}}\right) \right| \leq \frac{c(\epsilon, b_2, b_3, b_4, b_{4+\epsilon})}{n} \quad (35)$$

where

$$G_n(x, y) = \Phi(x) + \frac{y(2x^2 + 1)}{6\sqrt{n}} \Phi'(x). \quad (36)$$

Lemma 5 shows that the average of the X_i has a more precise, skewed CDF limit $G_n(x, y)$ where the skew term has weight proportional to a certain measure of skew derived from the moments: $\frac{\mu_3}{b_2^{3/2}}$. Note that if the X_i are symmetric, the weight of the correction term is zero, and the CDF of the average of the X_i converges to $\Phi(x)$ at a rate of $O(n^{-1})$.

Here the limit $G_n(x, y)$ is a normal CDF plus a correction term that decays as $n^{-1/2}$. Importantly, since

$\phi''(x) = x^2\phi(x) - \phi(x)$ where $\phi(x) = \Phi'(x)$, the correction term can be rewritten giving:

$$G_n(x, y) = \Phi(x) + \frac{y}{6\sqrt{n}}(2\phi''(x) + 3\phi(x)) \quad (37)$$

From which we see that $G_n(x, y)$ is a linear combination of $\Phi(x)$, $\phi(x)$ and $\phi''(x)$. In Algorithm 1, we correct for the difference in σ between Δ^* and the variance needed by X_{corr} using X_{nc} . This same method works when we wish to estimate the error in Δ^* vs $G_n(x, y)$. Since all of the component functions of $G_n(x, y)$ are derivatives of a (unit variance) $\Phi(x)$, adding a normal variable with variance σ' increases the variance of all three functions to $1 + \sigma'$. Thus we add X_{nc} as per Algorithm 1 preserving the limit in Equation (37).

The deconvolution approach can be used to construct a correction variable X_{corr} between $G_n(x, y)$ and $S(x)$ the standard logistic function. An additional complexity is that $G_n(x, y)$ has additional parameters y and n . Since these act as a single multiplier $\frac{y}{6\sqrt{n}}$ in Equation (37), its enough to consider a function $g(x, y')$ parametrized by $y' = \frac{y}{6\sqrt{n}}$. This function can be computed and saved offline. As we have shown earlier, errors in the “limit” function propagate directly through as errors in the acceptance test. To achieve a test error of 10^{-6} (close to single floating point precision), we need a y' spacing of 10^{-6} . It should not be necessary to tabulate values all the way to $y' = 1$, since y' is scaled inversely by the square root of minibatch size. Assuming a max y' of 0.1 requires us to tabulate about 100,000. Since our x resolution is 10,000, this leads to a table with about 1 billion values, which can comfortably be stored in memory. However, if $g(x, y)$ is moderately smooth in y , it should be possible to achieve similar accuracy with a much smaller table. We leave further analysis and experiments with $g(x, y)$ as future work.