

1 We thank all of the reviewers for their valuable comments and suggestions.

2 **Runtime Comparisons.** As suggested by all reviewers, we add wall-clock time comparisons in order to demonstrate
 3 empirically the computational speed-up of Poisson-Gibbs. We have replaced "Iterations" with "Time" in Figure 1
 4 (using the same setup) and report the corresponding results in Figure 3. The number of factors being evaluated of
 5 Poisson-Gibbs varies each iteration, so in Figure 3d, we report the average number of factors being evaluated per
 6 iteration. Overall, the empirical results presented in Figure 3 align with our theoretical analysis: Poisson-Gibbs is
 7 significantly faster than plain Gibbs on both the Potts model and the continuous spin model, and faster than other
 8 minibatch Gibbs methods on the Potts (they cannot be applied to continuous domain). Compared to plain Gibbs,
 9 Poisson-Gibbs speed up the computation by evaluating only a subset of factors in each iteration (Figure 3d). Compared
 10 to DoubleMIN-Gibbs, Poisson-Gibbs is faster because it removes the need of an additional M-H correction step.

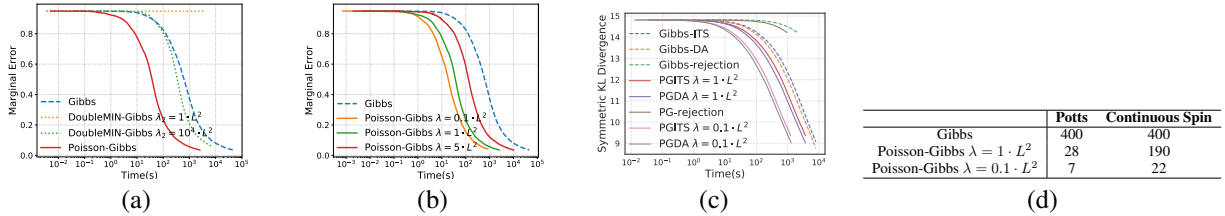


Figure 3: Runtime performance of Poisson-Gibbs, plain Gibbs and previous minibatch Gibbs methods on (a)-(b) a Potts model and (c) a continuous spin model. (d) Number of factors being evaluated per iteration.

11 **R2: Wall-clock time comparison and # of potential functions being evaluated.** Please refer to Figure 3. Additionally,
 12 the average number of factors being evaluated at each iteration in truncated Gaussian mixture is 1802 (10^6 in total).

13 **R2: Are there problems on which vanilla Gibbs would be prohibitively expensive?** Yes, there are many applications
 14 which use very large graphical models such as social and biological networks [12], which would make plain Gibbs
 15 infeasible. However, in these cases our methods would be hard to compare empirically without the ground truth and the
 16 results of plain Gibbs. This is why we focus on relatively large graphical models where it is still possible to run Gibbs
 17 sampling, though with a fairly long runtime. We would be happy to investigate applying our methods to larger models
 18 in the future work.

19 **R2: Are there problems on which Poisson-Gibbs might fail?** Our theorems suggest that, as long as plain Gibbs can
 20 converge well on this problem, our methods are guaranteed to perform similarly well by setting λ as suggested in the
 21 paper. The hyperparameters of the degree of Chebyshev polynomials in PGITS and PGDA need tuning for different
 22 tasks (e.g. grid search). Empirically, we did not find a poor initialization issue for Poisson-Gibbs. One weakness of
 23 Poisson-Gibbs is that it requires the domain to be bounded and the bound L cannot be too large. If the problem has
 24 a very large or even unbounded L , Poisson-Gibbs may not necessarily work efficiently. We would like to study the
 25 problem of extending Poisson-Gibbs to unbounded domains in the future.

26 **R2: Why is the supposition that s_ϕ are iid ok?** There is no additional supposition since the equation after line 379 is
 27 just another way to write the equation after line 378. The equivalence is because $\sum_x x * p(x)$ (Eq 378) = $\mathbf{E}[x]$ (Eq 379).
 28 Also, s_ϕ are independent but not identical since their parameters include different M_ϕ . (We will clarify this.)

29 **R2: "local maximum energy" seems a misnomer.** This name is from prior work [3], so for consistency we keep it.

30 **R3: Why not set $\lambda = 4L^2$ instead of $\lambda = L^2$ in simulation?** When setting λ larger (e.g. $4L^2$), the theoretical
 31 convergence rate of Poisson-Gibbs becomes closer to plain Gibbs but the expected minibatch size upper bound $\lambda + L$
 32 will become larger. So there is a trade-off. Also, setting $\lambda = 4L^2$ will result in $\exp(-1)$ in the bound, not canceling the
 33 exponential. Empirically, we find that setting $\lambda = L^2$ gives reasonably good performance (see Figure 1b).

34 **R3: $\lambda = 500$ not $O(L^2)$ in truncated Gaussian mixture.** We have considered higher values of λ and find that setting
 35 $\lambda = 500$ already provides good performance on this task. The theory guarantees that setting $\lambda = L^2$ can produce
 36 similar performance as plain Gibbs. But in practice we can sometimes set λ to be smaller to gain more speed-up.

37 **R1: δ_k should be proportional to $\exp(L + \delta_m)$ instead of $\exp(L)$ in Theorem 4.** Yes, we have fixed this.

38 **R1: Clarification.** For Table 1, we follow [3] to report the computational cost. For line 112, we want to say that our
 39 disjoint distribution will only use a subset of factors which achieves the same effect as minibatching. For line 129, we
 40 explain it in the later sentences (line 130-132) that we can do the sampling in a more efficient way and it is statistically
 41 equivalent. We will clarify the above in the final version.

42 **Presentation.** Thanks for all suggestions about the presentation. We will improve the presentation accordingly and
 43 check proofs and typos carefully. We will present Poisson-MH in a better way and add a simple experiment for it.