Supplementary Materials

A. XOR-Sampling for the Weighted Case

The text here provides a synopsis for the approach in (Ermon et al., 2013b). We still encourage the readers to read the original text for a better explanation. Let $w(x)$ as defined before, $Z = \sum_{x\in\mathcal{X}} w(x)$ and $P(x) = w(x)/Z$, the high-level idea of XOR-Sampling is to first dicretize $w(x)$ to $w'(x)$ as in Definition 1, followed by embedding the weighted $w'(x)$ to the unweighted space Δ_w. Finally, XOR-sampling uses hashing and randomization to sample uniformly from Δ_w.

Definition 1. *Assume $w(x)$ has both upper and lower bound, namely, $M = \max_x w(x)$ and $m = \min_x w(x)$. Let $b \geq 1, \epsilon > 0, r = 2^b/(2^b-1)$ and $l = \lceil \log_r(2^n/\epsilon)\rceil$. Partition the configurations into the following weight based disjoint buckets: $\mathcal{B}_i = \{x|w(x) \in (\frac{M}{r^{i+1}}, \frac{M}{r^i}]\}, i = 0, \ldots, l-1$ and $\mathcal{B}_l = \{x|w(x) \in (0, \frac{M}{r^l}]\}$. The discretized weight function $w' : \{0,1\}^n \to \mathbb{R}^+$ is defined as follows: $w'(x) = \frac{M}{r^{i+1}}$ if $x \in \mathcal{B}_i, i = 0, \ldots, l-1$ and $w'(x) = 0$ if $x \in \mathcal{B}_l$. This leads to the corresponding discretized probability distribution $p'(x) = w'(x)/Z'$ where Z' is the normalization constant of $w'(x)$.*

For the weighted case, the goal of XOR-sampling is to guarantee that the probability of sampling one x is proportional to the unnormalized density (up to a multiplicative constant). Using the discretization in Definition 1, we obtain a distribution $p'(x)$ which satisfying $\frac{1}{\rho}p(x) \leq p'(x) \leq \rho p(x)$ where $\rho = \frac{r^2}{1-\epsilon}$. Then, XOR-sampling implements a horizontal slice technique to transform a weighted problem into an unweighted one. For the easiness of illustration, we denote $M' = \max_x w'(x)$ and m' as the smallest nonzero value of $w'(x)$. Then consider a simple case where $b = 1$ and $r = 2$. In this case we have $M' = 2^{l-1}m'$. Let $y = (y_0, \ldots, y_{l-2})^T \in \{0,1\}^{l-1}$ be a binary vector of length $l-1$, XOR-sampling samples (x, y) uniformly at random from the following set Δ_w using the unweighted version of XOR-sampling based on hashing and randomization:

$$\Delta_w = \{(x, y) : w'(x) \leq 2^{i+1}m' \Rightarrow y_i = 0\}. \tag{13}$$

Upon obtaining one sample (x, y) uniformly at random from Δ_w, we only return x. It can be proved that the probability of sampling x from $w'(x)$ is proportional to $m'2^{i-1}$ when $w'(x)$ is sandwiched between $m'2^{i-1}$ and $m'2^i$. Therefore, this technique leads to the constant approximation guarantee of XOR-Sampling, which states formally as Theorem 5:

Theorem 5. *(Ermon et al., 2013b) Let $\epsilon > 0, b > 1, P \geq 2$, $0 < \delta_0 < 1$, and $\gamma_0 = \log((P+2\sqrt{P+1}+2)/P)$. For any $\alpha \in \mathbb{Z}$, $\alpha > \gamma_0$, let $c(\alpha, P) = 1-2^{\gamma_0-\alpha}/(1-\frac{1}{P}-2^{\gamma_0-\alpha})^2$. Let $r = 2^b/(2^b-1), l = \lceil \log_r(2^n/\epsilon)\rceil, \rho = r^2/(1-\epsilon), \kappa = 1/c(\alpha, P)$ and bucket $\mathcal{B}_l$ as in Definition 1 in the supplementary materials. Denote $Pr'_s(x)$ as distribution of the samples generated by XOR-Sampling$(w, l, b, \delta, P, \alpha)$ and let $\phi : \{0,1\}^n \to \mathbb{R}^+$ be one non-negative function. Then, with probability at least $(1-\delta_0)c(\alpha, P)2^{-(\gamma_0+\alpha+1)}\frac{P}{P-1}$, XOR-Sampling succeeds and outputs a sample x_0. Upon success, each x_0 is output with probability $P'(x_0)$, which is within a constant factor of the true $P(x_0)$. Furthermore, the expectation of a non-negative function $\phi(x)$, $\mathbb{E}_{P(x)}[\phi(x)]$ can be bounded by:*

$$\begin{aligned}\frac{1}{\rho\kappa}\mathbb{E}_{P'(x)}[\phi(x)] - \epsilon\eta_\phi \leq \mathbb{E}_{P(\theta)}[\phi(x)] \\ \leq \rho\kappa\mathbb{E}_{P'(x)}[\phi(x)] + \epsilon\eta_\phi.\end{aligned} \tag{14}$$

Theorem 1 is a simplified representation of the previous Theorem. Our new theoretic results can be stated simpler and clearer building upon the statement in Theorem 1. Assuming all the parameters of Theorem 5, we set $\delta = \rho\kappa$, and $\gamma = 1-(1-\delta_0)c(\alpha, P)2^{-(\gamma_0+\alpha+1)}\frac{P}{P-1}$ and can obtain the corresponding guarantee stated in Theorem 1.

B. Proofs

Theorem 2 states that the function value of the output of XOR-CD, in expectation converges to the true optimum within a small constant distance at a linear speed w.r.t. the number of iterations T. To prove Theorem 2, we first prove two lemmas.

Lemma 1. *If the total variation $\max_\theta Var_{P_\theta}(\phi(X)) \leq \sigma_2^2$, then $l(\theta)$ is σ_2^2-smooth w.r.t. θ.*

B.1. Proof of Lemma 1

Proof. Since $l(\theta) = -\frac{1}{N}\sum_{i=1}^N \log P_\theta(x_i)$, L-smoothness requires that

$$||\nabla l(\theta_1) - \nabla l(\theta_2)||_2 \leq L||\theta_1 - \theta_2||_2, \ \ \forall\theta_1, \theta_2 \in \ dom\ f,$$

where L is a constant. Because of the mean value theorem, there exists a point $\tilde{\theta} \in (\theta_1, \theta_2)$ such that

$$\nabla l(\theta_1) - \nabla l(\theta_2) = \nabla(\nabla l(\tilde{\theta}))(\theta_1 - \theta_2).$$

Taking the L_2 norm for both sides, we have

$$\begin{aligned}||\nabla l(\theta_1) - \nabla l(\theta_2)||_2 =& ||\nabla(\nabla l(\tilde{\theta}))(\theta_1 - \theta_2)||_2 \\ \leq& ||\nabla(\nabla l(\tilde{\theta}))||_2\ ||\theta_1 - \theta_2||_2\end{aligned} \tag{15}$$

Then, the problem is to bound the matrix 2-norm $||\nabla(\nabla l(\tilde{\theta}))||_2$. Since we know the explicit form of $l(\theta)$, we know

$$\begin{aligned}\nabla l(\theta) &= \nabla\Lambda(\theta) - \frac{1}{N}\sum_{i=1}^N \phi(x_i), \\ \nabla(\nabla l(\theta)) &= \sum_{x\in\mathcal{X}}[\phi(x) - \nabla\Lambda(\theta)][\phi(x) - \nabla\Lambda(\theta)]^T P_\theta(x),\end{aligned} \tag{16}$$

where $\nabla(\nabla l(\theta))$ is the co-variance matrix. Denote $\text{Cov}_\theta[\phi(X)] = \nabla(\nabla l(\theta))$, which is both symmetric and positive semi-definite. We have

$$||\nabla(\nabla l(\tilde{\theta}))||_2 = ||\text{Cov}_\theta[\phi(X)]||_2 = \lambda_{max},$$

where λ_{max} is the maximum eigenvalue of the matrix $\text{Cov}_\theta[\phi(X)]$. Then, because of the positive semi-definiteness of the co-variance matrix, all the eigenvalues are non-negative, and we can bound λ_{max} as

$$\lambda_{max} \leq \sum_i \lambda_i = Tr(\text{Cov}_\theta[\phi(X)]),$$

where $Tr(\text{Cov}_\theta[\phi(X)])$ is the trace of matrix $\text{Cov}_\theta[\phi(X)]$. Using the definition in Equation 16, $Tr(\text{Cov}_\theta[\phi(X)])$ can be further derived as:

$$Tr(\text{Cov}_\theta[\phi(X)]) = \mathbb{E}_{P_\theta}[||\phi(X)||_2^2] - ||\mathbb{E}_{P_\theta}[\phi(X)]||_2^2,$$

which is equal to the total variation $Var_{P_\theta}(\phi(X))$. Therefore, we have

$$||\nabla(\nabla l(\tilde{\theta}))||_2 \leq Var_{P_\theta}(\phi(X)) \leq \sigma_2^2.$$

Combining this with Equation 15, we know

$$||\nabla l(\theta_1) - \nabla l(\theta_2)||_2 \leq \sigma_2^2 \, ||\theta_1 - \theta_2||_2.$$

This completes the proof.

□

In addition, based on the constant approximation of $\mathbb{E}_{D,P'_\theta}[\overline{g_t}]$, we can bound another two important terms shown in Lemma 2.

Lemma 2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function and $\theta^* = argmin_\theta f(\theta)$. In iteration t, g_t is the estimated gradient. If there exists a constant $c \geq 1$ s.t. $\frac{1}{c}[\nabla f(\theta_t)]^+ \leq \mathbb{E}[g_t^+] \leq c[\nabla f(\theta_t)]^+$ and $c[\nabla f(\theta_t)]^- \leq \mathbb{E}[g_t^-] \leq \frac{1}{c}[\nabla f(\theta_t)]^-$, then we have*

$$\frac{1}{c}||\mathbb{E}[g_t]||_2^2 \leq \langle \nabla f(\theta_t), \mathbb{E}[g_t] \rangle \leq c||\mathbb{E}[g_t]||_2^2.$$

$$\frac{1}{c}\langle \mathbb{E}[g_t], \theta_t - \theta^* \rangle \leq \langle \nabla f(\theta_t), \theta_t - \theta^* \rangle \leq c\langle \mathbb{E}[g_t], \theta_t - \theta^* \rangle.$$

B.2. Proof of Lemma 2

Proof. (Lemma 2) Since we have the constant bound that

$$\frac{1}{c}\nabla f(\theta_t)^+ \leq \mathbb{E}[g_t^+] \leq c\nabla f(\theta_t)^+. \tag{17}$$

$$c\nabla f(\theta_t)^- \leq \mathbb{E}[g_t^-] \leq \frac{1}{c}\nabla f(\theta_t)^-. \tag{18}$$

and because of $g_t^+ \geq \mathbf{0}$ and $g_t^- \leq \mathbf{0}$ we can obtain

$$\begin{aligned}
\frac{1}{c}||\mathbb{E}[g_t^+]||_2^2 = \frac{1}{c}\langle \mathbb{E}[g_t^+], \mathbb{E}[g_t^+] \rangle &\leq \langle \nabla f(\theta_t)^+, \mathbb{E}[g_t^+] \rangle \\
&\leq c\langle \mathbb{E}[g_t^+], \mathbb{E}[g_t^+] \rangle = c||\mathbb{E}[g_t^+]||_2^2. \\
\frac{1}{c}||\mathbb{E}[g_t^-]||_2^2 = \frac{1}{c}\langle \mathbb{E}[g_t^-], \mathbb{E}[g_t^-] \rangle &\leq \langle \nabla f(\theta_t)^-, \mathbb{E}[g_t^-] \rangle \\
&\leq c\langle \mathbb{E}[g_t^-], \mathbb{E}[g_t^-] \rangle = c||\mathbb{E}[g_t^-]||_2^2.
\end{aligned}$$

which exactly means

$$\frac{1}{c}||\mathbb{E}[g_t]||_2^2 \leq \langle \nabla f(\theta_t), \mathbb{E}[g_t] \rangle \leq c||\mathbb{E}[g_t]||_2^2.$$

To prove the second inequality, we need to take advantage of the convexity of f. Denote $[\theta_t - \theta^*]^+ = \max\{\theta_t - \theta^*, \mathbf{0}\}$ and $[\theta_t - \theta^*]^- = \min\{\theta_t - \theta^*, \mathbf{0}\}$, we know $\theta_t - \theta^* = [\theta_t - \theta^*]^+ + [\theta_t - \theta^*]^-$. In addition, because f is convex, the index set of non-zero entries of $[\theta_t - \theta^*]^+$ and $\nabla f(\theta_t)^+$ is the same. The index set of non-zero entries of $[\theta_t - \theta^*]^-$ and $\nabla f(\theta_t)^-$ is also the same. In addition, because of Equation 17 and 18, the index set of non-zero entries of $\mathbb{E}[g_t^+]$ ($\mathbb{E}[g_t^-]$) is the same with $\nabla f(\theta_t)^+$ ($\nabla f(\theta_t)^-$). Combining these facts with Equations 17 and 18, we have

$$\begin{aligned}
\frac{1}{c}\langle \mathbb{E}[g_t^+], [\theta_t - \theta^*]^+ \rangle &\leq \langle \nabla f(\theta_t)^+, [\theta_t - \theta^*]^+ \rangle \\
&\leq c\langle \mathbb{E}[g_t^+], [\theta_t - \theta^*]^+ \rangle. \\
\frac{1}{c}\langle \mathbb{E}[g_t^-], [\theta_t - \theta^*]^- \rangle &\leq \langle \nabla f(\theta_t)^-, [\theta_t - \theta^*]^- \rangle \\
&\leq c\langle \mathbb{E}[g_t^-], [\theta_t - \theta^*]^- \rangle.
\end{aligned}$$

Combining these two equations, we have

$$\frac{1}{c}\langle \mathbb{E}[g_t], \theta_t - \theta^* \rangle \leq \langle \nabla f(\theta_t), \theta_t - \theta^* \rangle \leq c\langle \mathbb{E}[g_t], \theta_t - \theta^* \rangle.$$

This completes the proof. □

Lemma 2 gives the new bounds of two terms assuming the constant bound on the gradient, which are essential to the proof of convergence rate. Based on Lemma 2, we can prove Theorem 3, which bounds the error of Stochastic Gradient Descent (SGD) on a convex optimization problem when the estimated gradient g_t in the t-th step resides in a constant bound of $\nabla f(\theta_t)$.

B.3. Proof of Theorem 3

Theorem 3 indicates that even the expectation of gradients in each iteration only have a constant approximation, SGD is still able to converge to the optimal solution within a small constant gap at linear speed. The complete proof of Theorem 3 is as follows:

Proof. (Theorem 3) By L-smooth of f, for the t-th iteration,

$$\begin{aligned}
f(\theta_{t+1}) &\leq f(\theta_t) + \langle \nabla f(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2}||\theta_{t+1} - \theta_t||_2^2, \\
&= f(\theta_t) - \eta\langle \nabla f(\theta_t), g_t \rangle + \frac{Lt^2}{2}||g_t||^2.
\end{aligned}$$

Because of the constant bound on gradient and $||\mathbb{E}[g_t]||_2^2 = \mathbb{E}[||g_t||_2^2] - Var(g_t)$, by taking expectation on both sides w.r.t g_t we get from Lemma 2 that

$$\begin{aligned}
&\mathbb{E}[f(\theta_{t+1})] \leq f(\theta_t) - \frac{\eta}{c}||\mathbb{E}[g_t]||_2^2 + \frac{L\eta^2}{2}\mathbb{E}[||g_t||_2^2], \\
&= f(\theta_t) - \frac{\eta}{c}(\mathbb{E}[||g_t||_2^2] - Var(g_t)) + \frac{L\eta^2}{2}\mathbb{E}[||g_t||_2^2], \\
&\leq f(\theta_t) - \frac{\eta(2-L\eta c)}{2c}\mathbb{E}[||g_t||_2^2] + \frac{\eta}{c}\sigma^2, \\
&\leq f(\theta_t) - \frac{\eta c}{2}\mathbb{E}[||g_t||_2^2] + \frac{\eta}{c}\sigma^2,
\end{aligned}$$

where the last inequality follows as $L\eta c \leq 2 - c^2$. Because f is convex, still from Lemma 2 we get

$$\begin{aligned}
&\mathbb{E}[f(\theta_{t+1})] \\
&\leq f(\theta^*) + \langle \nabla f(\theta_t), \theta_t - \theta^* \rangle - \frac{\eta c}{2}\mathbb{E}[||g_t||_2^2] + \frac{\eta}{c}\sigma^2, \\
&\leq f(\theta^*) + c\langle \mathbb{E}[g_t], \theta_t - \theta^* \rangle - \frac{\eta c}{2}\mathbb{E}[||g_t||_2^2] + \frac{\eta}{c}\sigma^2, \\
&= f(\theta^*) + c\mathbb{E}[\langle g_t, \theta_t - \theta^* \rangle - \frac{\eta}{2}||g_t||_2^2] + \frac{\eta}{c}\sigma^2.
\end{aligned}$$

We now repeat the calculations by completing the square for the middle two terms to get

$$\begin{aligned}
&\mathbb{E}[f(\theta_{t+1})] \\
&\leq f(\theta^*) + \frac{c}{2\eta}\mathbb{E}[2\eta\langle g_t, \theta_t - \theta^* \rangle - \eta^2||g_t||_2^2] + \frac{\eta}{c}\sigma^2, \\
&\leq f(\theta^*) + \frac{c}{2\eta}\mathbb{E}[||\theta_t - \theta^*||_2^2 - ||\theta_t - \theta^* - \eta g_t||_2^2] + \frac{\eta}{c}\sigma^2, \\
&= f(\theta^*) + \frac{c}{2\eta}\mathbb{E}[(||\theta_t - \theta^*||_2^2 - ||\theta_{t+1} - \theta^*||_2^2)] + \frac{\eta}{c}\sigma^2.
\end{aligned}$$

Summing the above equations for $t = 0, \ldots, T-1$, we get

$$\begin{aligned}
&\sum_{t=0}^{T-1}\mathbb{E}[f(\theta_{t+1}) - f(\theta^*)] \\
&\leq \frac{c}{2\eta}(||\theta_0 - \theta^*||_2^2 - \mathbb{E}[||\theta_T - \theta^*||_2^2]) + \frac{T\eta}{c}\sigma^2 \\
&\leq \frac{c||\theta_0 - \theta^*||_2^2}{2\eta} + \frac{T\eta}{c}\sigma^2.
\end{aligned}$$

Finally, by Jensen's inequality, $tf(\overline{\theta_T}) \leq \sum_{t=1}^{T} f(\theta_t)$,

$$\begin{aligned}
\sum_{t=0}^{T-1}\mathbb{E}[f(\theta_{t+1}) - f(\theta^*)] &= \mathbb{E}[\sum_{t=1}^{T} f(\theta_t)] - Tf(\theta^*) \\
&\geq T\mathbb{E}[f(\overline{\theta_T})] - Tf(\theta^*).
\end{aligned}$$

Combining the above equations we get

$$\mathbb{E}[f(\overline{\theta_T})] \leq f(\theta^*) + \frac{c||\theta_0 - \theta^*||_2^2}{2\eta T} + \frac{\eta}{c}\sigma^2.$$

This completes the proof. □

B.4. Proof of Theorem 2

Finally, we give the full proof of Theorem 2 as follows:

Proof. (Theorem 2) Since we use M samples from the training set $\{x_i\}_{i=1}^N$ and K samples $x'_1, \ldots, x'_K$ from $P_{\theta_t}(X)$ using XOR-Sampling at each iteration, we have

$$\overline{g_t} = \frac{1}{M}\sum_{j=1}^{M}\phi(x_j) - \frac{1}{K}\sum_{i=1}^{K}\phi(x'_i).$$

Denote $g_t^i = \frac{1}{M}\sum_{j=1}^{M}\phi(x_j) - \phi(x'_i)$, we have the expectation of $\overline{g_t}$ as

$$\mathbb{E}_{D,P'_\theta}[\overline{g_t}] = \mathbb{E}_{P'_\theta}[\mathbb{E}_D[\phi(x)] - \phi(x')] = \mathbb{E}_{D,P'_\theta}[g_t^i].$$

In each iteration t we can adjust the parameters in XOR-Sampling to make the tail $\epsilon\eta_\phi$ zero, then for each g_t^i we can obtain from Theorem 1 that

$$\frac{1}{\delta}[g(\theta_t)]^+ \leq \mathbb{E}_{D,P'_\theta}[g_t^{i+}] \leq \delta[g(\theta_t)]^+. \tag{19}$$

$$\delta[g(\theta_t)]^- \leq \mathbb{E}_{D,P'_\theta}[g_t^{i-}] \leq \frac{1}{\delta}[g(\theta_t)]^-. \tag{20}$$

where $g(\theta_t)$ is the true gradient at t-th iteration. Denote $\overline{g_t}^+ = \max\{\overline{g_t}, \mathbf{0}\}$ and $\overline{g_t}^- = \min\{\overline{g_t}, \mathbf{0}\}$. Clearly, $g_t^{i+} \geq 0$ and $g_t^{i-} \leq 0$. Moreover, for a given dimension, either $g_t^{i+} = 0$ for that dimension or $g_t^{i-} = 0$. Evaluating $\overline{g_t}$ dimension by dimension, we can see that $\overline{g_t}^+ = \frac{1}{K}\sum_{i=1}^{K} g_t^{i+}$ and $\overline{g_t}^- = \frac{1}{K}\sum_{i=1}^{K} g_t^{i-}$. Combined with Equation 19 and 20, we know

$$\frac{1}{\delta}[g(\theta_t)]^+ \leq \mathbb{E}_{D,P'_\theta}[\overline{g_t}^+] \leq \delta[g(\theta_t)]^+.$$

$$\delta[g(\theta_t)]^- \leq \mathbb{E}_{D,P'_\theta}[\overline{g_t}^-] \leq \frac{1}{\delta}[g(\theta_t)]^-.$$

In terms of variance, the variance of each $\phi(x'_i)$ can be bounded by

$$\begin{aligned}
Var_{P'_\theta}(\phi(x'_i)) &= \mathbb{E}_{P'_\theta}[||\phi(x_i)||_2^2] - ||\mathbb{E}_{P'_\theta}[\phi(x_i)]||_2^2, \\
&\leq \delta\mathbb{E}_{P_\theta}[||\phi(x_i)||_2^2], \\
&= \delta(Var_{P_\theta}(\phi(x_i)) + ||\mathbb{E}_{P_\theta}[\phi(x_i)]||_2^2), \\
&\leq \delta(\sigma_2^2 + \varepsilon^2).
\end{aligned}$$

Because $\mathbb{E}_{D,P'_\theta}[\overline{g_t}] = \mathbb{E}_{D,P'_\theta}[g_t^i]$, the variance of $\overline{g_t}$, denoted as $Var_{D,P'_\theta}(\overline{g_t})$, can then be bounded as

$$\begin{aligned}
&Var_{D,P'_\theta}(\overline{g_t}) \\
&= Var_D(\frac{1}{M}\sum_{j=1}^{M}\phi(X_j)) + Var_{P'_\theta}(\frac{1}{K}\sum_{i=1}^{K}\phi(x'_i)) \\
&= \frac{1}{M}Var_D(\phi(X_j)) + \frac{1}{K}Var_{P'_\theta}(\phi(x'_i))
\end{aligned}$$

$$\leq \frac{1}{M}\sigma_1^2 + \frac{\delta}{K}(\sigma_2^2 + \varepsilon^2)$$

Therefore, since $l(\theta)$ is convex and σ_2^2-smooth from Lemma 1, we can then apply Theorem 3 to get the result in Theorem 2.

$$\begin{aligned} &OPT - \mathbb{E}[l(\overline{\theta^T})] \\ &\leq \frac{\delta||\theta_0 - \theta^*||_2^2}{2\eta T} + \frac{\eta \max_{\theta_t}\{Var_{D,P'_\theta}(\overline{g_t})\}}{\delta} \\ &\leq \frac{\delta||\theta_0 - \theta^*||_2^2}{2\eta T} + \frac{\eta(\sigma_2^2 + \varepsilon^2)}{K} + \frac{\eta\sigma_1^2}{\delta M}. \end{aligned}$$

This completes the proof. □

B.5. Proof of Theorem 4

Proof. (Theorem 4) From Theorem 1 we know that in each iteration of XOR-CD, we need to access $O(n \ln \frac{n}{\gamma})$ queries of NP oracles in order to generate one sample. However, as specified also in Ermon et al. (2013b), only the first sample needs those many queries. Once we have the first sample, the number of XOR constraints to add (depends on the sizes of the set Δ_w stated in supplementary materials section A) can be known in generating future samples for this SGD iteration. Therefore, we fix the number of XOR constraints added starting the generation of the second sample. As a result, we only need one NP oracle query in generating each of the following $K-1$ samples. Therefore, total queries in each iteration will be $O(n \ln \frac{n}{\gamma} + K)$. To complete all T SGD iterations, XOR-CD needs $O(Tn \ln \frac{n}{\gamma} + TK)$ NP oracle queries in total. □

C. Additional Experimental Details

Here we show some additional experiments we have done for this paper and additional details of the experiments discussed in the main text.

C.1. Maximum Likelihood Learning

Here we show XOR-CD is able to learn exponential family models with higher likelihood compared to competing approaches. We consider a discrete exponential family model with n binary variables $x = (x_1, \ldots, x_n)^T$, where each $x_i \in \{0, 1\}$ for $i \in \{1, \ldots, n\}$. The exponential family model we consider is in the form: $Pr(x) \propto \exp(\sum_{k=1}^{K} \theta_k^T \phi(x^k))$. Here, each x^k is a subset of all n variables, and is often referred to as a *clique*. Suppose x^k is of size l_k, ϕ is the Cartetian product, i.e., it maps a vector of l_k binary variables to a vector of size 2^{l_k}, where each entry in the vector evaluates to 1 if and only if x^k takes a particular assignment (There are 2^{l_k} different value assignments to l_k binary variables).

We synthetically generate a few exponential family models and test if learning algorithms can rediscover these models. In generating one model, we first draw the number of cliques uniformly from $[n, 2n]$. The size of each clique is chosen from the range of $[1, 6]$ at random. Then, to generate $\theta_k = (\theta_{k,1}, \ldots, \theta_{k,2^{l_k}})^T$, each $\theta_{k,i}$ is generated in the form of $\theta_{k,i} = v_{ki1} + v_{ki2}v_{ki3}$, where v_{ki1} is uniformly drawn from $(0, 1)$, v_{ki3} uniformly from $(10, 1000)$ and binary variable v_{ki2} uniformly randomly drawn from $\{0, 1\}$. In the experiment, we vary n from 10 to 31 in intervals of 3, and generate 10 models for each n. For each model, we generate 1000 training data points from the ground-truth probability distribution (possibly overlapping) and see if learning algorithms can rediscover the exponential family models.

In learning exponential family models, we keep the structure of the exponential family model to be learned the same as the one that generates training data, and initialize all θ parameters to be the absolute values of samples drawn from a Gaussian distribution $\mathcal{N}(10, 10)$. Learning rate is fixed as 0.1 and parameters in XOR-Sampling are the same as in (Ermon et al., 2013b). For comparison, in addition to Gibbs-CD, we also compare with Belief Propagation equipped Contrastive Divergence (Ping & Ihler, 2017), denoted as BP-CD, and BPChain-CD (Fan & Xue, 2020). We allow the competing methods to draw 10000 samples from the model distribution while XOR-CD only draws 100 samples for a fair comparison (because it takes less time to draw samples using e.g., MCMC). When testing, we use ACE (Barton et al., 2016) to sample exactly from a target distribution. We implement XOR-CD using IBM ILOG CPLEX Optimizer 12.63 for queries to NP oracles. Experiments are carried out on a cluster, where each node has 24 cores and 96GB memory.

Likelihood Comparison Figure 3 shows the results of the four algorithms. The x-axis is exponential family model with different numbers of variables, and y-axis is the average log-likelihood of 1000 randomly generated samples from models learned by each of the learning algorithm. Here we use ACE (Barton et al., 2016) to compute the exact log-likelihood. We can see XOR-CD learns models that generate samples with higher average log-likelihood compared to competing approaches.

Time complexity We test the time complexity of different methods and find XOR-CD runs faster than Gibbs-CD. In particular, XOR-CD with 100 samples takes 1 minute 50 seconds per XOR iteration, while Gibbs-CD with 10,000 MCMC samples needs 2.5 minutes when learning models with dimension $n = 31$. Notice that XOR-CD outperforms Gibbs-CD in likelihood values also.

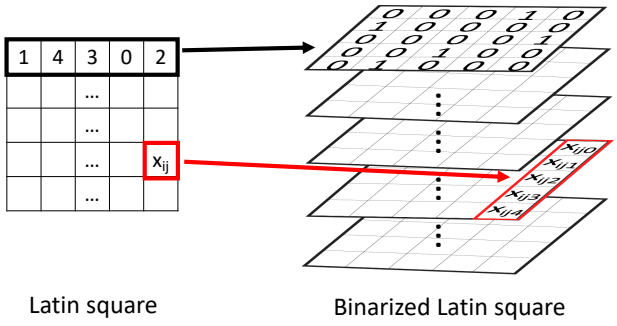

Figure 6. This figure shows how to binarize a Latin square. For each entry $x_{i,j} \in \{1, 2, ..., n\}$ in the $i-$th row and $j-$th column of a Latin square, we use n binary variables $x_{i,j,k} \in \{0, 1\}$ to represent its value where $k \in \{1, 2, ..., n\}$. If the value of $x_{i,j}$ is k, then only $x_{i,j,k}$ is equal to 1 and the other $n-1$ variables are equal to 0.

C.2. Dispatching Route Generation

Here is additional information regarding the dispatching route generation experiment.

Models and experimental settings We learn an exponential family model to capture the likelihood of different permutations. Specifically, the exponential family model is $Pr(x) \propto \exp(\theta_0 + \sum_{i,j} \theta_{i,j} x_{i,j} + \sum_{i,j,k} \theta_{i,j,k} x_{i,j} x_{i+1,k})$ and the θ's are the parameters to learn. In generating training data, we assume the n locations are fully-connected, i.e., there is an edge between every two locations. Assuming the locations are labeled from 1 to n, we assume the distance between two locations is the difference between its two indices. For example, the distance between 1 and 3 is 2 (=3-1). The total travel distance d of a delivery route is the sum of the distances of each edge traveled in a trip. We randomly sample delivery routes according to the travel distance distribution shown in the rightmost column of the middle figure of Figure 4 to form the training dataset.

In learning the exponential family model, we initialize each model parameter (e.g., θ_0, $\theta_{i,j}$, $\theta_{i,j,k}$) to be the absolute value of samples drawn from a Gaussian distribution $\mathcal{N}(10, 10)$. For XOR-CD, we set $T = 500$ and a time limit of 10 hours. $M = K = 100$, and learning rate is 0.1. Parameters of XOR-Sampling are set the same as in Ermon et al. (2013b) in order to ensure $\delta = \sqrt{2}$. For Gibbs-CD, we let $K = 10000$ and the others are set the same as XOR-CD. As for the structure of GAN, the generator takes the input of random noise vector of dimension n (n is the number of locations), and has the structure $fc\{2n\} - fc\{5n\} - fc\{n^2\}$. Here, $fc\{.\}$ denotes a fully connected layer of output dimension $\{.\}$. For example, the first fully connected layer $fc\{2n\}$ maps an input of dimension n to an output of dimension $2n$. The output of the generator is of the shape n^2 which represent variables x_{ij}. The discriminator has a structure $fc\{n^2\} - fc\{5n\} - fc\{2n\} - fc\{2\} - softmax$. We use ReLU as the activation function between fc layers. Number of training epochs is 1000, and the wall-time limit is also 10 hours.

Hamilton Cycle Constraints In sampling from current model distribution, XOR-CD has to sample the assignments to variables $x_{i,j}$ which form valid Hamilton cycles; ie, the locations to visit in a route form a permutation. We include the following two constraints in the mixed integer program of XOR-sampling to enforce this contraint:

$$\sum_{j=1}^{n} x_{i,j} = 1, \quad \forall i \in \{1, ..., n\}; \tag{21}$$

$$\sum_{i=1}^{n} x_{i,j} = 1, \quad \forall j \in \{1, ..., n\}. \tag{22}$$

C.3. Optimal Experiment Design

Models and experimental settings Let $x_{i,j,k}$ be an indicator variable which is 1 if and only if crop k is planted at the i, j-th entry. The exponential family model is: $Pr(x) \propto \exp(\theta_0 + \sum_{i,j,k} \theta_{i,j,k} x_{i,j,k} + \sum_{i,j,m,l} \theta^1_{i,j,m,l} \phi(x_{i,j,m}, x_{i+1,j,l}) + \theta^2_{i,j,m,l} \phi(x_{i,j,m}, x_{i,j+1,l})$, in which $\theta_{i,j,k}, \theta^1_{i,j,m,l}, \theta^2_{i,j,m,l}$ are the parameters to learn. We define $d_i(j, k)$ as the distance between symbols j and k in row i. This distance is calculated as the absolute difference of the column indices of where symbols j and k appear in row i. The total distance $d(j, k)$ is defined as $d(j, k) = \sum_{i=1}^{n} d_i(j, k)$. The spatial variance of a Latin square is defined as the variance of all the total distances $d(i, j), \forall i \neq j$. This spatial variance is an important metric determining whether an experiment design is good (Gomes et al., 2004; Smith et al.; Le Bras et al., 2012). Notice that we can prove there are only three possible variances for a 5x5 Latin square. In generating training data, we randomly sample 5-by-5 Latin squares to form a training set, the distribution of the spatial variance of which is shown in the rightmost column of the right figure of Figure 4.

We set hyper-parameters of both XOR-CD and Gibbs-CD the same as in the dispatching route generation task. As for the structure of GAN, the generator takes the input of random noise vector of dimension n, and has the structure $fc\{2n\} - fc\{n^2\} - fc\{n^3/2\} - fc\{n^3\}$. The output of the generator is of the shape n^3 which represent the assignments to variables x_{ijk}. The discriminator has a structure $fc\{n^3/2\} - fc\{n^2\} - fc\{n\} - fc\{2\} - softmax$. Here fc denotes fully connected layer and the number denotes the dimension of output of this layer. We use ReLU as the activation function between each two fc layers. Number of epochs is 1000, and the wall-time limit is also 10 hours.

Latin Square Constraints We can enforce the following

set of constraints to ensure the output forms a Latin square in XOR-CD:

$$\sum_{k=1}^{n} x_{i,j,k} = 1, \quad \forall i, j \in \{1, ..., n\}; \tag{23}$$

$$\sum_{j=1}^{n} x_{i,j,k} = 1, \quad \forall i, k \in \{1, ..., n\}; \tag{24}$$

$$\sum_{i=1}^{n} x_{i,j,k} = 1, \quad \forall j, k \in \{1, ..., n\}. \tag{25}$$

Constraints (23) indicate each cell in the Latin square must be a valid integer between 1 and n. Constraints (24) indicate cells in each row must be different, and constraints (25) indicate cells in each column must be different.

C.4. Sequence-based Protein Homology Detection

As shown in Figure 1, the alignment matrix of sequence $\mathcal{S}_1$ of size N_1 and sequence $\mathcal{S}_2$ of size N_2 is of size $(N_1 + 1) \times (N_2 + 1)$. In the matrix, the rows represent the amino acids in $\mathcal{S}_1$ and the columns represent the ones in $\mathcal{S}_2$. Each alignment forms a path from the upper-left node to the bottom-right node as shown in the right panel of Figure 1, where each transition in the path is either horizontal, representing an insertion in $\mathcal{S}_2$, vertical, representing an insertion in $\mathcal{S}_1$, or diagonal, representing a match. We use symbol M, I_1 and I_2 to represent a match, an insertion in $\mathcal{S}_1$, and an insertion in $\mathcal{S}_2$, respectively. The (i, j)-th node in the alignment matrix is associated with three binary variables: $z^u_{i,j}$, where $u \in \{M, I_1, I_2\}$. $z^u_{i,j}$ is 1 if and only if the path passes the (i, j)-th node with type u. Let $A^p = \{z^u_{i,j} = z^{u(p)}_{i,j}, 1 \le i \le N_1 + 1, 1 \le j \le N_2 + 1, u \in \{M, I_1, I_2\}\}$ be one value assignment to all $z^u_{i,j}$ variables that form path p. Notice A^p also represents one alignment between sequence $\mathcal{S}_1$ and $\mathcal{S}_2$. Hence we will refer a path and an alignment interchangeably. Let $A = \{A^p \mid p \text{ is a valid path}\}$ be the set of all alignments. Our exponential family model formulation estimates the probability of having the alignment A^p to be:

$$P_\theta(A^p|\mathcal{S}_1, \mathcal{S}_2) = \frac{e^{\sum_{i,j,u} \theta^u_{i,j} z^{u(p)}_{i,j} + \sum_{i,j,u,k,l,v} \theta^{uv}_{i,j,k,l} z^{u(p)}_{i,j} z^{v(p)}_{k,l}}}{\mathcal{Z}(\mathcal{S}_1, \mathcal{S}_2)},$$

where $\mathcal{Z}(\mathcal{S}_1, \mathcal{S}_2) = \sum_{A^p \in A} e^{\sum \theta^u_{i,j} z^{u(p)}_{i,j} + \sum \theta^{uv}_{i,j,k,l} z^{u(p)}_{i,j} z^{v(p)}_{k,l}}$ is the normalization factor. Here, $\theta = \{\theta^u_{i,j}, \theta^{uv}_{i,j,k,l}\}$ are the set of variables to learn. In practice, we further parameterize $\theta^u_{i,j}$ to be $\theta^u_{i,j} = r^T \Phi_{i,j}(\mathcal{S}_{1i}, \mathcal{S}_{2j}, u)$, and parameterize $\theta^{uv}_{i,j,k,l}$ to be $\theta^{uv}_{i,j,k,l} = s^T \Xi_{i,j}(\mathcal{S}_{1i}, \mathcal{S}_{2j}, u, \mathcal{S}_{1k}, \mathcal{S}_{2l}, v)$, where both Φ and Ξ are features extracted from data, and we instead focus on learning parameters r and s. Features are extracted based on profile conservation, secondary structure, and solvent accessibility of each protein.

Learning. Given a training set of $(A^p_k, \mathcal{S}^k_{1,k}, \mathcal{S}_{2,k})_{k=1}^N$, where $\mathcal{S}_{1,k}, \mathcal{S}_{2,k}$ are a pair of sequences, and A^p_k is the observed alignment between the two sequences. We want to learn the exponential family model via maximizing the likelihood, which translates to the following problem:

$$\max_{r,s} \prod_{k=1}^{N} P_{r,s}(A^p_k|\mathcal{S}_{1,k}, \mathcal{S}_{2,k}).$$

Inference. After learning $P(A^p|\mathcal{S}_1, \mathcal{S}_2)$, we can use the model to find the best alignment between two new sequences by solving the following linear programming problem:

$$\begin{aligned} A^{p*} &= \arg\max_{A^p \in A} P(A^p|\mathcal{S}_1, \mathcal{S}_2) \\ &= \arg\max_{A^p \in A} \sum_{i,j,u} \theta^u_{i,j} z^{u(p)}_{i,j} + \sum_{i,j,k,l,u,v} \theta^{uv}_{i,j,k,l} z^{u(p)}_{i,j} z^{v(p)}_{k,l}. \end{aligned}$$

Sequence Alignment Constraints In the task of sequence alignment, each alignment must be a valid path in the alignment matrix as shown in Figure 1. Consider the alignment matrix of size $(N_1 + 1) \times (N_2 + 1)$, and $3N_1N_2$ binary variables $z^u_{i,j}$, where $1 \le i \le N_1 + 1, 1 \le j \le N_2 + 1$, and $u \in \{M, I_1, I_2\}$. If we consider the alignment matrix as a directed graph and each node (i, j) has three income edges, i.e., $z^{I_2}_{ij}$ denotes the edge from node $(i, j-1)$, $z^{I_1}_{ij}$ denotes the edge from node $(i-1, j)$ and z^M_{ij} denotes the edge from node $(i-1, j-1)$. $z^u_{i,j} = 1$ means the corresponding edge exists and $z^u_{i,j} = 0$ otherwise. Then, to form a valid path (alignment), the following set of constraints $\mathcal{C}$ must be satisfied:

$$z^M_{1,1} = 1, z^{I_1}_{1,1} = 0, z^{I_2}_{1,1} = 0; \tag{26}$$

$$\sum_u z^u_{N_1+1,N_2+1} = 1; \tag{27}$$

$$for \quad \forall i \in \{1, ..., N_1\}, j \in \{1, ..., N_2\}:$$
$$\sum_u z^u_{i,j} - z^{I_1}_{i+1,j} - z^{I_2}_{i,j+1} - z^M_{i+1,j+1} = 0; \tag{28}$$

$$for \quad \forall i = N_1 + 1, j \in \{1, ..., N_2\}:$$
$$\sum_u z^u_{i,j} - z^{I_2}_{i,j+1} = 0; \tag{29}$$

$$for \quad \forall i \in \{1, ..., N_1\}, j = N_2 + 1, :$$
$$\sum_u z^u_{i,j} - z^{I_1}_{i+1,j} = 0; \tag{30}$$

$$for \quad \forall i = 1, j \in \{2, ..., N_2 + 1\}:$$
$$z^M_{i,j} + z^{I_1}_{i,j} = 0; \tag{31}$$

$$for \quad \forall i \in \{2, ..., N_1 + 1\}, j = 1, :$$
$$z^M_{i,j} + z^{I_2}_{i,j} = 0. \tag{32}$$

Experiment Setup. To run XOR-CD in this experiment, we set $T = 500$ and force each query of NP oracle in XOR-CD to stop in 15 minutes. As a result, not all the queries to

NP oracles are solved up to optimality. We also enforce a timeout of 10 hours for all algorithms. The learning rate is 0.1 for the first 100 epochs and 0.01 for the next 400 epochs for both XOR-CD and Gibbs-CD. Both M and K are set to 100 in XOR-CD and parameters in XOR-Sampling are set the same as in (Ermon et al., 2013b). For Gibbs-CD, we change K to 10000 for a fair comparison, which takes approximately the same amount of time compared with XOR-CD for each SGD iteration.