**Problem**

Sample from distribution \( p(x) \propto e^{-f(x)}, x \in \mathbb{R}^d \) given access to \( f(x), \nabla f(x) \) (e.g., sampling posteriors).

**Background**

The great divide of optimization

<table>
<thead>
<tr>
<th>Convex optimization</th>
<th>Non-convex optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local minima = global minima</td>
<td>Possibly bad local minima</td>
</tr>
<tr>
<td>Gradient descent finds global min.</td>
<td>Gradient descent can be bad</td>
</tr>
<tr>
<td>Provably algorithms, beautiful math</td>
<td>NP-hard in the worst-case (messy?)</td>
</tr>
<tr>
<td>ML problems often non-convex</td>
<td>Works remarkably well in practice</td>
</tr>
</tbody>
</table>

The great divide of sampling

<table>
<thead>
<tr>
<th>Log-concave distribution</th>
<th>Non-log-concave distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unimodal</td>
<td>Potentially multimodal</td>
</tr>
<tr>
<td>Natural algorithm: Langevin diffusion</td>
<td>Langevin can mix exponentially slowly</td>
</tr>
<tr>
<td>Provably algorithms, beautiful math</td>
<td>#P-hard in the worst-case (messy?)</td>
</tr>
<tr>
<td>ML problems often non log-concave</td>
<td>Works well in practice</td>
</tr>
</tbody>
</table>

**Fixing Langevin?**

A Markov chain with local moves such as Langevin diffusion gets stuck in a local mode. Creating a meta-Markov chain which changes the temperature (simulated tempering) can exponentially speed up mixing.

**Our question:** Can we give provable guarantees for such an algorithm in natural "non-log-concave" settings?

**Main Theorem**

Let \( p(x) \propto e^{-f(x)} \) on \( \mathbb{R}^d \) be s.t. \( f(x) = -\log \left( \sum_{j=1}^m w_j e^{-\frac{|x-y_j|^2}{2\sigma_j^2}} \right) \)

and we can query \( f(x), \nabla f(x) \). There is an algorithm (based on Langevin diffusion + simulated tempering) running in time poly \( \left( \frac{1}{\epsilon^{2}}, \frac{1}{\sigma^2}, \frac{d}{\max\|\mu\|} \right) \) that samples from a distribution \( q \)

with \( \|p - q\|_2 \leq \epsilon \). A \( \epsilon \) perturbation of \( \Delta \) multiplies time by a factor poly \( (\epsilon^3) \).

**Algorithmic tools**

1. **Langevin diffusion** (gradient flow + Brownian motion or in discrete form gradient descent + gaussian noise)
2. **Simulated tempering:** heuristic for speeding up MCs on multimodal distributions

**Simulated tempering + Langevin diffusion**

At point \((i,x)\),

- Evolve according to Langevin with inverse temperature \( \beta_i \):
  \[
  dx_i = -\beta_i \nabla f(x_i) dt + \sqrt{2\beta_i} dW_i.
  \]
- Propose swaps with rate \( \lambda \).
- When a swap is proposed, pick \( i' = i \pm 1 \) with probability \( \frac{1}{2} \). Set next point to \((i',x)\) with probability \( \min \left( \frac{p(x')}{p(x)}, 1 \right) \).

**Proof outline**

1. Markov chain decomposition theorem
2. Mixing for each component
3. Mixing for "projected" chain
4. Mixing for "actual" heating \( p_i \propto \sum_{j=1}^m w_j \exp \left( -\beta_i |x-y_j|^2 \right) \)

**Main theorem**

Decomposing using distributions

Inspiration: MC decomposition theorem (Madras, Randall 2002)

If MC mixes rapidly when restricted to each set of a partition, and "projected" MC mixes rapidly => MC mixes rapidly.

(Transition in projected chain: avg. prob. flow between sets.)

**Soft partition**

We prove a new decomposition theorem for distributions instead of sets.

**Soft decomposition theorem**

Let tempering chain be made up of Markov chains \( M_i \). Suppose there is a decomposition \( M(x,y) = \sum M_i(x,y)M_j(y) \) where \( M_j \) has stationary distribution \( p_j \), if each \( M_j \) mixes in time \( C \) and projected chain mixes in time \( \tilde{C} \) => simulated tempering mixes in time \( \tilde{C}C \).

Intuition:
1. (i) mixing time is equal to Poincare constant \( \max[\text{Var}_{\mu}(p)/\text{Var}_{\mu}(g)] \) where \( \text{Var}(g, h) = -\langle g, \nabla h \rangle \) is Dirichlet (bilinear) form and \( L \) is the generator of MC.
2. (i) Dirichlet form "decomposes" into Langevin chains for components, and variance decomposes as

   \[
   \text{Var}_{\mu}(p) = \sum_{i=1}^m \sum_{j=1}^m W_{ij} \left[ \text{Var}_{\mu_i}(p_{ij}) + \left( E_{p_{ij}}[C] - \text{Var}_{\mu_i}(g) \right)^2 \right]
   \]

   Use Poincare inequality for \( p_{ij} \) get factor of \( C \).

   Use Poincare inequality for \( \beta_i \) get factor of \( C \).

Again, for intuition, 2 extreme cases.
1. If all expectations \( E_{p_{ij}}[\beta_i] \) are equal => factor \( C \) from the component chains.
2. If \( \beta_i \)’s constant on each \( p_{ij} \), only vary between \( p_{ij} \)'s => factor \( C \) from projected chain.

**Project chain has large probability flow between \((i,j)\) in the same or adjacent levels with similar distributions:\n
\[
L((i,j),(i',j')) = \sum_{\Omega_{ij}} P_{ij}(\nu_{ij}/\nu_{ij'})
\]

where \( \sum_{\Omega_{ij}} \nu_{ij}(p,q) = \max[p(x)^{\Omega_{ij}}(p), q(x)^{\Omega_{ij}}(q)] \).

Using the decomposition theorem:
1. Apply Langevin for "approximately" heated distributions (Langevin on individual components \( p_{ij} \propto \exp(\frac{-|x-y_j|^2}{2\sigma_j^2}) \) mixes rapidly).
2. Compare to "actually" heated distributions, losing factors of \( W_{ij} \).