A Proof of Theorem 1

Proof. Assume that $\mathcal{P}$ is piecewise smooth under analytic partition. Thus,

$$\mathcal{P}(x) = \sum_{i=1}^{N} \prod_{j=1}^{M_i} \mathbb{1}[p_{i,j}(x) \geq 0] \cdot \prod_{l=1}^{O_i} \mathbb{1}[q_{i,l}(x) < 0] \cdot h_i(x) \tag{2}$$

for some N, M_i, O_i and $p_{i,j}, q_{i,l}, h_i$ that satisfy the properties in Definition 1.

We use one well-known fact: the zero set $\{x \in \mathbb{R}^n \mid p(x) = 0\}$ of an analytic function p is the entire $\mathbb{R}^n$ or has zero Lebesgue measure [30]. We apply the fact to each $p_{i,j}$ and deduce that the zero set of $p_{i,j}$ is $\mathbb{R}^n$ or has measure zero. Note that if the zero set of $p_{i,j}$ is the entire $\mathbb{R}^n$, the indicator function $\mathbb{1}[p_{i,j} \geq 0]$ becomes the constant-1 function, so that it can be omitted from the RHS of equation (2). In the rest of the proof, we assume that this simplification is already done so that the zero set of $p_{i,j}$ has measure zero for every i, j.

For every $1 \leq i \leq N$, we decompose the i-th region

$$R_i = \{x \mid p_{i,j} \geq 0 \text{ and } q_{i,l}(x) < 0 \text{ for all } j, l\} \tag{3}$$

to

$$\begin{aligned} R_i' &= \{x \mid p_{i,j} > 0 \text{ and } q_{i,l}(x) < 0 \text{ for all } j, l\} \\ R_i'' &= R_i \setminus R_i'. \end{aligned} \tag{4}$$

Note that R_i' is open because the $p_{i,j}$ and $q_{i,l}$ are analytic and so continuous, both $\{r \in \mathbb{R} \mid r > 0\}$ and $\{r \in \mathbb{R} \mid r < 0\}$ are open, and the inverse images of open sets by continuous functions are open. This means that for each $x \in R_i'$, we can find an open ball at x inside R_i' so that $\mathcal{P}(x') = h_i(x')$ for all x' in the ball. Since h_i is smooth, this implies that $\mathcal{P}$ is differentiable at every $x \in R_i'$.

For the other part R_i'', we notice that

$$R_i'' \subseteq \bigcup_{j=1}^{M_i} \{x \mid p_{i,j}(x) = 0\}.$$

The RHS of this equation is a finite union of measure-zero sets, so it has measure zero. Thus, R_i'' also has measure zero as well.

Since $\{R_i\}_{1 \leq i \leq N}$ is a partition of $\mathbb{R}^n$, we have that

$$\mathbb{R}^n = \bigcup_{i=1}^{N} R_i' \cup \bigcup_{i=1}^{N} R_i''.$$

The density $\mathcal{P}$ is differentiable on the union of R_i''s. Also, since the union of finitely or countably many measure-zero sets has measure zero, the union of R_i'''s has measure zero. Thus, we can set the set A required in the theorem to be this second union. □

B Proof of Theorem 2

Proof. As shown in Equation 1,

$$\mathcal{P} := \Big(\sum_{i=1}^{N_D} \eta_i \cdot k_i\Big) \cdot \Big(\sum_{j=1}^{N_F} \zeta_j \cdot l_j\Big)$$

it suffices to show that both factors are non-negative and piecewise smooth under analytic partition, because such functions are closed under multiplication.

We prove a more general result. For any expression e, let $\mathrm{Free}(e)$ be the set of its free variables. Also, if a function $\mathcal{G}$ in Definition 1 satisfies additionally that its h_i's are analytic, we say that this function $\mathcal{G}$ is *piecewise analytic* under analytic partition. We claim that for all expressions e (which may contain free variables), if $e \rightsquigarrow (\Delta, \Gamma, D, F)$, where $D = \{(\eta_i, k_i) \mid 1 \leq i \leq N_D\}$ and $F = \{(\zeta_j, l_j, v_j) \mid 1 \leq j \leq N_F\}$, then $\Big(\sum_{i=1}^{N_D} \eta_i \cdot k_i\Big)$ and $\Big(\sum_{j=1}^{N_F} \zeta_j \cdot l_j\Big)$ are non-negative functions on variables in $\mathrm{Free}(e) \cup \Delta$ and they are piecewise analytic under analytic partition, as k and l' in the sum are analytic. These two properties in turn imply that $\Big(\sum_{i=1}^{N_D} \eta_i \cdot k_i\Big) \cdot \Big(\sum_{j=1}^{N_F} \zeta_j \cdot l_j\Big)$ is a function on variables in $\mathrm{Free}(e) \cup \Delta$ and it is also piecewise analytic (and thus piecewise smooth) under analytic partition. Thus, the desired conclusion follows. Regarding our claim, we can prove it by induction on the structure of the expression e. □

C Discontinuous Hamiltonian Monte Carlo

The discontinuous HMC (DHMC) algorithm was proposed by [8]. It uses a coordinate-wise integrator, Algorithm 1, coupled with a Laplacian momentum to perform inference in models with non-differentiable densities. The algorithm works because the Laplacian momentum ensures that all discontinuous parameters move in steps of $\pm m_b \epsilon$ for fixed constants m_b and step size ϵ, where the index b is associated to each discontinuous coordinate. These properties are advantages because they remove the need to know where the discontinuity boundaries between each region are; the change of the potential energy in the state before and after the $\pm m_b \epsilon$ move provides us with information of whether we have enough kinetic energy to move into this new region. If we do not have enough energy we reflect backwards $\mathbf{p}_b = -\mathbf{p}_b$. Otherwise, we move to this new region with a proposed coordinate update $\mathbf{x}_b^*$ and momentum $\mathbf{p}_b - m_b \cdot sign(\mathbf{p}_b) \cdot \Delta U$. This is in contrast to algorithms such as Reflect, Refract HMC [7], that explictly need to know where the discontinuities boundaries are. Hence, it is important to have a compilation scheme that enables one to do that.

The addition of the random permutation ϕ of indices b is to ensure that the coordinate-wise integrator satisfies the criterion of reversibility in the Hamiltonian. Although the integrator does not reproduce the exact solution, it nonetheless preserves the Hamiltonian exactly, even if the density is discontinuous. See Lemma 1 and Theorems 2-3 in [8]. This yields a rejection-free proposal.

Algorithm 1 Coordinate-wise Integrator. A random permutation ϕ on $\{1, \dots, B\}$ is appropriate if the induced random sequences $(\phi(1), \dots, \phi(|B|))$ and $(\phi(|B|), \dots, \phi(1))$ have the same distribution

function COORDINATEWISE($\mathbf{x}, \mathbf{p}, \epsilon, U$)
    pick an appropriate random permutation ϕ on B
    for $i = 1, \dots, B$ **do**
        $b \leftarrow \phi(i)$
        $\mathbf{x}^* \leftarrow \mathbf{x}$
        $\mathbf{x}_b^* \leftarrow \mathbf{x}_b^* + \epsilon m_b \cdot sign(\mathbf{p}_b)$
        $\Delta U \leftarrow U(\mathbf{x}^*) - U(\mathbf{x})$
        if $K(\mathbf{p}_b) = m_b|\mathbf{p}_b| > \Delta U$ **then**
            $\mathbf{x}_b \leftarrow \mathbf{x}_b^*$
            $\mathbf{p}_b \leftarrow \mathbf{p}_b - m_b \cdot sign(\mathbf{p}_b) \cdot \Delta U$
        else
            $\mathbf{p}_b \leftarrow -\mathbf{p}_b$
        end if
    end for
    return $\mathbf{x}_b, \mathbf{p}_b$
end function

Then DHMC algorithm [8] adpated for LF-PPL and our compilation scheme is as follows:

Algorithm 2 Discontinuous HMC Integrator for the LF-PPL.
χ is a map from random-variable names n in Δ to their values $\mathbf{x}_n$, H is the total Hamiltonian, $\epsilon > 0$ is the step size, and L is the trajectory length.

function DHMC-LFPPL$(\Delta, \Gamma, D, F, \mathbf{x}, \mathbf{p}, H, \epsilon, L)$
    $B = \Gamma; \quad A = \Delta \setminus \Gamma$
    for $a \in A$ **do** ▷ a represents the set of continuous variables
        $\mathbf{x}_a^0 \leftarrow \mathbf{x}_a; \quad \mathbf{p}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
    end for
    for $b \in B$ **do**
        $\mathbf{x}_b^0 \leftarrow \mathbf{x}_b; \quad \mathbf{p}_b \sim \mathit{Laplace}(\mathbf{0}, \mathbf{1})$ ▷ b represents the set of discontinuous variables
    end for
    $\forall a \in A,\ \mathbf{x}_a^0 \leftarrow \mathbf{x}_a; \quad \mathbf{p}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ ▷ A represents the set of continuous variables
    $\forall b \in B,\ \mathbf{x}_b^0 \leftarrow \mathbf{x}_b; \quad \mathbf{p}_b \sim \mathit{Laplace}(\mathbf{0}, \mathbf{1})$ ▷ B represents the set of discontinuous variables
    $U \leftarrow -\text{LogJointDensity}(D, F)$
    for $i = 1$ to L **do**
        $U_A \leftarrow U$ with names in B replaced by their values in $\mathbf{x}_B^i$
        $(\mathbf{x}_A^i, \mathbf{p}_A^i) \leftarrow \text{Halfstep1}(\mathbf{x}_A^{i-1}, \mathbf{p}_A^{i-1}, \epsilon, U_A)$
        $U_B \leftarrow U$ with names in A replaced by their values in $\mathbf{x}_A^i$
        $(\mathbf{x}_B^i, \mathbf{p}_B^i) \leftarrow \text{Coordinate-wise}(\mathbf{x}_B^{i-1}, \mathbf{p}_B^{i-1}, \epsilon, U_B)$
        $U_A \leftarrow U$ with names in B replaced by their values in $\mathbf{x}_B^i$
        $(\mathbf{x}_A^i, \mathbf{p}_A^i) \leftarrow \text{Halfstep2}(\mathbf{x}_A^i, \mathbf{p}_A^i, \epsilon, U_A)$
    end for
    $\mathbf{x}^L \leftarrow \mathbf{x}_A^L \cup \mathbf{x}_B^L,\ \mathbf{p}^L \leftarrow \mathbf{p}_A^L \cup \mathbf{p}_B^L;$
    $\mathbf{x}^*, \mathbf{p}^* \leftarrow \text{Evaluate}(F,\ \mathbf{x}^L, \mathbf{p}^L)$
    $\alpha \sim Uniform(0, 1)$
    if $\alpha > \min\{1, \exp(H(\mathbf{x}, \mathbf{p}) - H(\mathbf{x}^*, \mathbf{p}^*))\}$ **then**
        return $\mathbf{x}^*, \mathbf{p}^*$
    else
        return $\mathbf{x}, \mathbf{p}$
    end if
end function
function HALFSTEP1$(\mathbf{x}, \mathbf{p}, \epsilon, U)$
    $\mathbf{p}' \leftarrow \mathbf{p} - \frac{\epsilon}{2} \nabla_{\mathbf{x}} U(\mathbf{x})$
    $\mathbf{x}' \leftarrow \mathbf{x} + \frac{\epsilon}{2} \nabla_{\mathbf{p}'} K(\mathbf{p}')$
    return $(\mathbf{x}', \mathbf{p}')$
end function
function HALFSTEP2$(\mathbf{x}, \mathbf{p}, \epsilon, U)$
    $\mathbf{x}' \leftarrow \mathbf{x} + \frac{\epsilon}{2} \nabla_{\mathbf{p}} K(\mathbf{p})$
    $\mathbf{p}' \leftarrow \mathbf{p} - \frac{\epsilon}{2} \nabla_{\mathbf{x}'} U(\mathbf{x}')$
    return $(\mathbf{x}', \mathbf{p}')$
end function

D Program code

```
(let [y (vector -2.0 -2.5 ... 2.8)
     pi [0.5 0.5]
     z1 (sample (categorical pi))
     ...
     z10(sample (categorical pi))
     mu1 (sample (normal 0 2))
     mu2 (sample (normal 0 2))
     mus (vector mu1 mu2)]
  (if (< (- z1) 0)
      (observe (normal mu1 1) (nth y 0))
      (observe (normal mu2 1) (nth y 0)))
  ...
  (if (< (- z10) 0)
     (observe (normal mu1 1) (nth y 9))
     (observe (normal mu2 1) (nth y 9)))
  (mu1 mu2 z1 ... z10))
```

Figure 4: The LF-PPL version of the Gaussian mixture model detailed in Section 6.

```
(let [x (sample (uniform -6 6))
      abs-x (max x (- x))
      z (- (sqrt (* x (* A x))))]
   (if (< (- abs-x 3) 0)
      (observe (factor z) 0)
      (observe (factor (- z 1)) 0))
  x)
```

Figure 5: The LF-PPL version of the heavy-tailed model detailed in Section 6.