Code structure

- *langevin_sampling_gm.py*: Demonstrates Langevin sampling from a 1D Gaussian mixture distribution with known score.
- *score_matching_gm.py*: The score function of a 1D Gaussian mixture is learned by a simple MLP. In this case, we have access to the ground-truth score during training. We then sample from the trained model using Langevin sampling.
- *sliced_score_matching_gm.py*: Demonstrates Langevin sampling from a 2D Gaussian mixture with score learned by a simple MLP. Implements the basic score matching objective function, as well as sliced score matching.

Sliced Score Matching

The score matching objective is

$$J(\theta) = \frac{1}{2} E_{p_{\text{data}}} [\|\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x)\|_2^2]$$

Since $\nabla_x \log p_{\text{data}}(x)$ is unknown, we construct an equivalent formulation

$$J(\theta) = E_{p_{\text{data}}} \left[\operatorname{tr}(\nabla_x^2 \log p_\theta(x)) + \frac{1}{2} \|\nabla_x \log p_\theta(x)\|_2^2 \right] + \text{const}$$

This avoids the dependence on $\nabla_x \log p_{\text{data}}(x)$, but still requires us to compute the Hessian ∇_x^2, which is impractical with high dimensionality. *Sliced score matching* proposes to project the scores onto random directions, turning the vector fields of scores into scalar fields. Doing this yields the following objective

$$J(\theta) = E_{p_{\text{data}}} \left[v^T \nabla_x^2 \log p_\theta(x) v + \frac{1}{2} \left(v^T \nabla_x \log p_\theta(x) \right)^2 \right] + \text{const}$$