1 Experiment Details

In this section, we provide more details about the experimental configurations and the training procedure used for our Latent-Variable Energy-Based Model (LV-EBM). We evaluate our method on both synthetic datasets (LCR-2D, LCS-3D, HMR-2D) and real-world UCI benchmarks (POWER, MINIBOONE).

Table 1 summarizes the hyperparameters used in each dataset. We maintain a consistent configuration for most parameters, while tuning the number of training steps and Langevin steps to accommodate the varying complexity of the data distributions.

Algorithm 1 gives the pseudocode for our method. The algorithm begins by initializing a joint buffer and a latent bank. Inside the main training loop, it first updates the latent variables associated with the current data batch using conditional Langevin updates to approximate the posterior. Second, it updates the joint buffer using joint Langevin dynamics to generate samples from the model distribution. These updated particles are then used to compute the positive-phase energy from the latent bank and the negative-phase energy from the joint buffer samples, and the neural network parameters are finally updated using this loss.

Hyperparameter	**Notation**	**LCR-2D**	**LCS-3D**	**HMR-2D**	**UCI**
Joint buffer size	N	4096	4096	4096	4096
Latent per data point	M	16	16	16	16
Batch size	B	512	512	512	512
Training steps	T	4000	5000	7000	6000
Langevin steps	K	30	25	20	25
Step sizes	η_x, η_z	10^{-3}	8×10^{-4}	10^{-3}	10^{-3}
Refresh prob. (joint)	p_{joint}	0.02	0.02	0.02	0.02
Refresh prob. (bank)	p_{bank}	0.10	0.10	0.10	0.10
Noise scale (for x, z)	ξ_x, ξ_z	1.0	1.0	1.0	1.0

Table 1: Hyperparameter settings for LV-EBM experiments. The table lists the configurations used for LCR-2D, LCS-3D, HMR-2D, and UCI datasets.

Algorithm 1: LV-EBM Algorithm

Input: Dataset $\{x_i\}_{i=1}^n$; energy $E_\theta(x,z)$; joint buffer size N; number of latent particles per data point M; Langevin steps K; step sizes η_x, η_z; temperature τ; refresh probabilities $p_{\text{joint}}, p_{\text{bank}}$.

Initialize joint buffer $\mathcal{J} = \{(x^{(j)}, z^{(j)})\}_{j=1}^N$ from any distribution;

Initialize latent bank $\mathcal{B} = \{z_i^{(m)}\}_{i=1..n,\, m=1..M}$ from any distribution;

for $t \leftarrow 1$ **to** T **do**

    Sample a batch index set $I = \{i_1, \ldots, i_B\}$;
    $x_b \leftarrow \{x_i\}_{i \in I}$;
    $z_{I,1:M} \leftarrow \{z_i^{(m)} \mid i \in I, m = 1..M\}$ from latent bank;
    `// Latent bank Langevin updates`
    Flatten $z_{I,1:M}$ into $Z \in \mathbb{R}^{B \times M \times 2}$;
    for $k \leftarrow 1$ **to** K **do**
        $g_z \leftarrow \nabla_Z E_\theta(X, Z)$;
        Sample $\xi_z \sim \mathcal{N}(0, I)$;
        $Z \leftarrow Z - \eta_z g_z + \sqrt{2\eta_z}\, \xi_z$;
    Reshape Z to $\{z_i^{(m)}\}_{i \in I, m=1..M}$ and write back into $\mathcal{B}$;
    Reinitialize individual latent-bank particle $z_i^{(m)}$ periodically with probability p_{bank};
    `// Joint buffer Langevin updates`
    $(x^{\text{joint}}, z^{\text{joint}}) \leftarrow \mathcal{J}$
    for $k \leftarrow 1$ **to** K **do**
        $g_x, g_z \leftarrow \nabla_{x^{\text{joint}}, z^{\text{joint}}} E_\theta(x^{\text{joint}}, z^{\text{joint}})$;
        Sample $\xi_x, \xi_z \sim \mathcal{N}(0, I)$;
        $x^{\text{joint}} \leftarrow x^{\text{joint}} - \eta_x g_x + \sqrt{2\eta_x}\, \xi_x$;
        $z^{\text{joint}} \leftarrow z^{\text{joint}} - \eta_z g_z + \sqrt{2\eta_z}\, \xi_z$;
    Reinitialize individual joint particle $z_i^{(m)}$ periodically with probability p_{joint};
    `// Positive phase:  latent particles`
    $E_{b,m}^{\text{pos}} \; = \; E_\theta\big(x_b,\; z_{i_b}^{(m)}\big),\; b = 1, \ldots, B,\;\; m = 1, \ldots, M.$
    $e_{\text{pos}} \leftarrow \frac{1}{B} \sum_{b=1}^{B} \left[-\tau \log \left(\frac{1}{M} \sum_{m=1}^{M} \exp\big(- E_{b,m}^{\text{pos}} / \tau \big) \right) \right]$;
    `// Negative phase:  joint particles`
    Sample B pairs $\{(x_b^-, z_b^-)\}_{b=1}^B$ from joint buffer $\mathcal{J}$;
    $e_{\text{neg}} \leftarrow \frac{1}{B} \sum_{b=1}^{B} E_\theta(x_b^-, z_b^-)$;
    `// Loss and parameter update`
    loss $= e_{\text{pos}} - e_{\text{neg}}$;
    Update θ by one step of Adam/SGD on loss;