THRML-Powered RBM Demonstration

Training an RBM on binary bars and stripes data

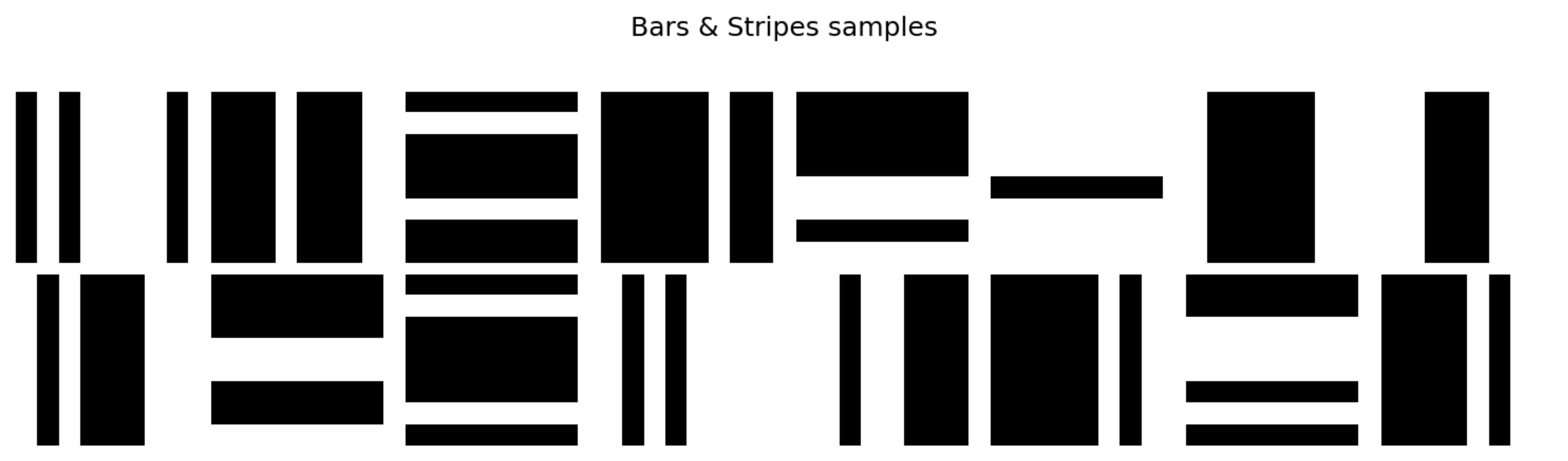
Alen Ribić & Alessio Toniolo, 2025

Overview

- Build a **Restricted Boltzmann Machine (RBM)** as an **Ising Energy-Based Model**
- Train it using **THRML's KL Gradient Estimator**
- Evaluate using:
  - 1-step **clamped reconstructions**
  - **THRML free-running sampling**
  - **Naive Python Gibbs sampling**
- Show how **THRML acts as a general Gibbs engine**
- Visualize learned filters and sampling results

Dataset: 8×8 Bars & Stripes

- Synthetic binary dataset used in EBM research
- Contains:
    - Horizontal bars
    - Vertical bars
    - Multiple bars per image
- Structured, multimodal, ideal for generative testing


Bars & Stripes samples

Training with THRML

THRML provides:

- **Block sampling** engine
- **KL divergence gradient estimator**
- **Factor graph representation**
- **Sampling schedules**
- Multi-chain sampling via `batch_shape`

We train RBM parameters:

- weights (W)
- visible biases (b_v), hidden biases (b_h)
- temperature parameter (\beta)

Our RBM/Ising energy function

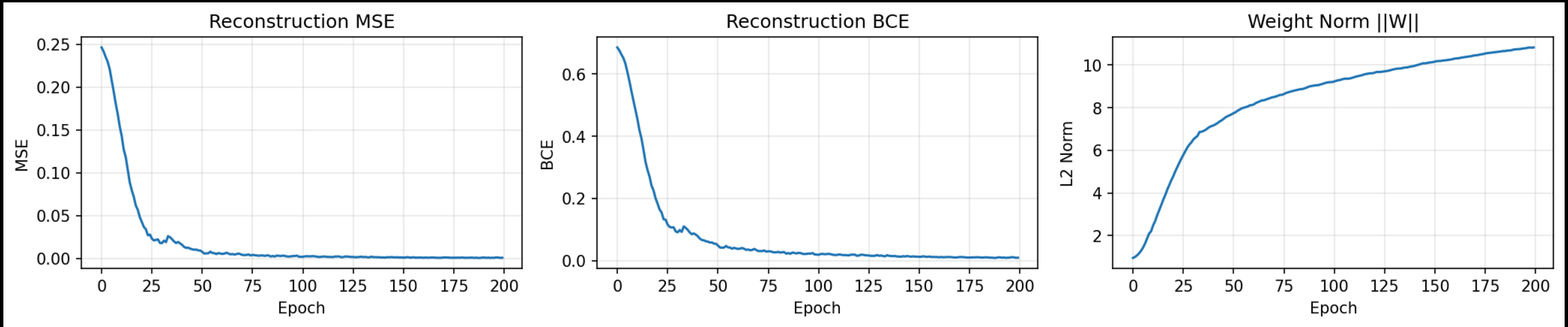
$$E(v,h) = -\beta \left[\sum_i b_i^{(v)} v_i + \sum_j b_j^{(h)} h_j + \sum_{i,j} W_{ij} v_i h_j \right]$$

directly taken from the Gibbs/Ising Hamiltonian, and each sum corresponds to a physical energy term

- 1st term, external field on visible spins
- 2nd term, external field on hidden spins
- Pairwise interaction energy

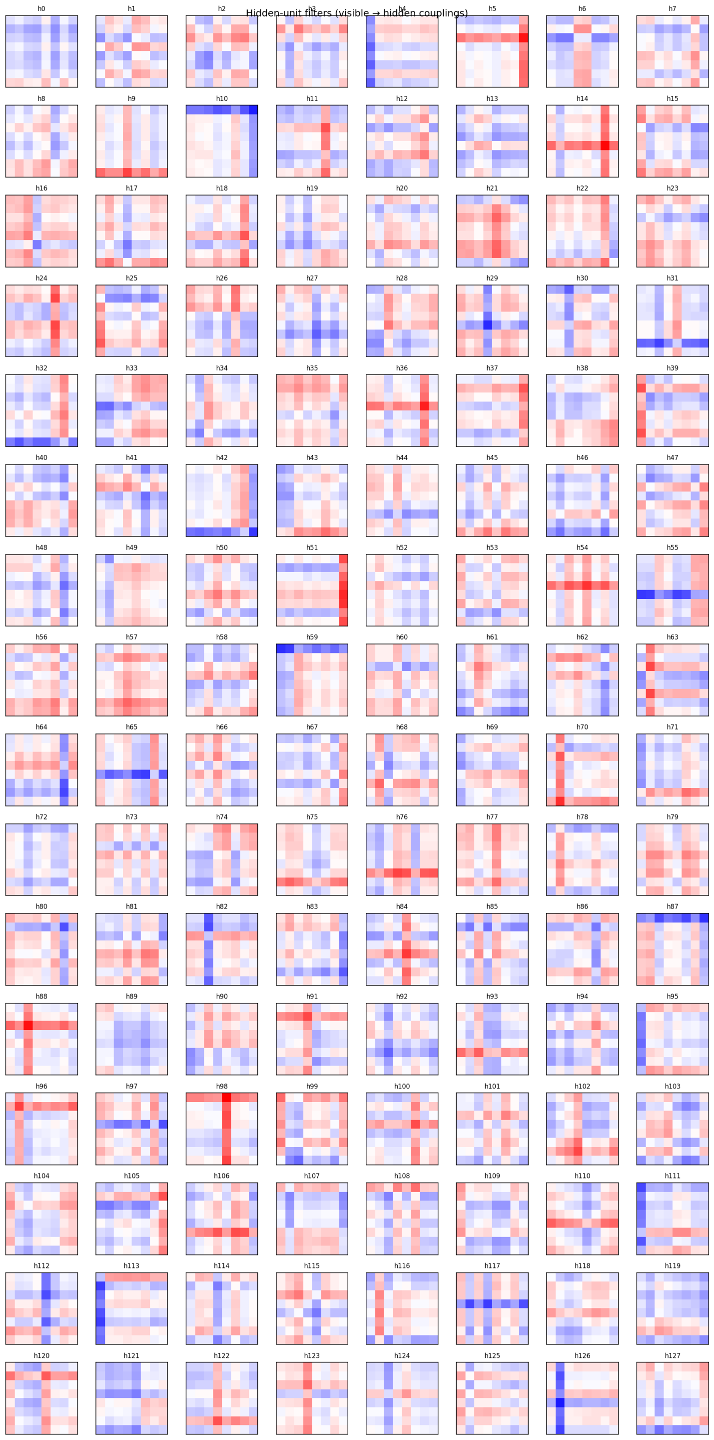
Training Curves

- **MSE** drops sharply → good reconstructions
- **BCE** approaches ~0 → near-perfect predictions
- Weight norm increases steadily → filters specialize


Reconstruction MSE
MSE
0.25
0.20
0.15
0.10
0.05
0.00
0
25
50
75
100
125
150
175
200
Epoch
Reconstruction BCE
BCE
0.6
0.4
0.2
0.0
Weight Norm ||W||
L2 Norm
10
8
6
4
2

Learned Hidden Unit Filters

- Each hidden unit learns a **feature detector**
- Filters resemble:
    - Vertical bars
    - Horizontal bars
    - Crossings
    - Composite stripe patterns
- RBM learns a **distributed encoding** of the dataset


Hidden-unit filters (visible → hidden couplings)
h0
h1
h2
h3
h4
h5
h6
h7
h8
h9
h10
h11
h12
h13
h14
h15
h16
h17
h18
h19
h20
h21
h22
h23
h24
h25
h26
h27
h28
h29
h30
h31
h32
h33
h34
h35
h36
h37
h38
h39
h40
h41
h42
h43
h44
h45
h46
h47
h48
h49
h50
h51
h52
h53
h54
h55
h56
h57
h58
h59
h60
h61
h62
h63
h64
h65
h66
h67
h68
h69
h70
h71
h72
h73
h74
h75
h76
h77
h78
h79
h80
h81
h82
h83
h84
h85
h86
h87
h88
h89
h90
h91
h92
h93
h94
h95
h96
h97
h98
h99
h100
h101
h102
h103
h104
h105
h106
h107
h108
h109
h110
h111
h112
h113
h114
h115
h116
h117
h118
h119
h120
h121
h122
h123
h124
h125
h126
h127

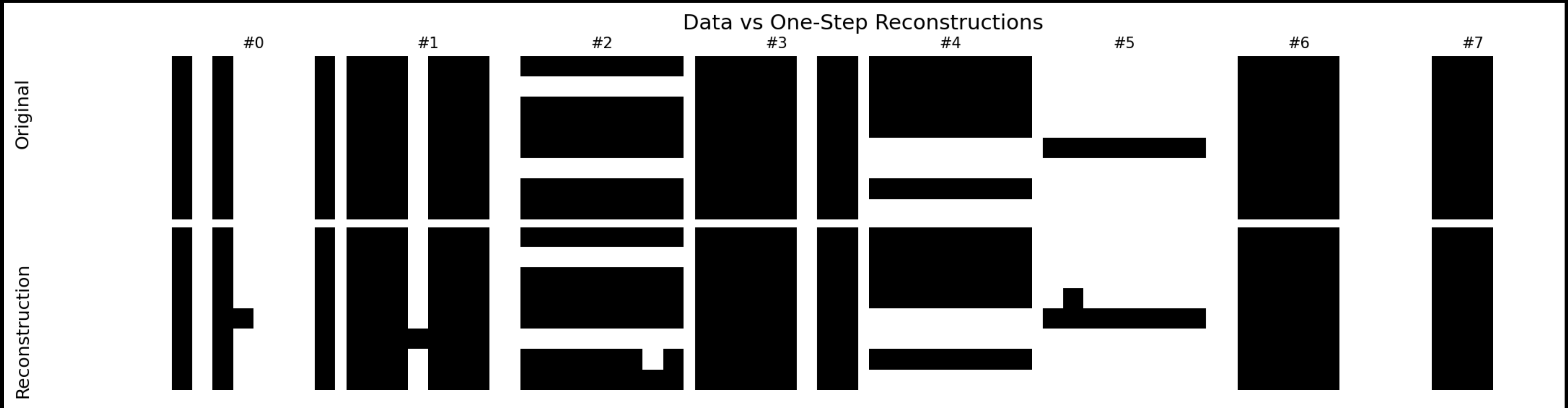
Clamped 1-Step Reconstructions

Process:

1. Clamp visibles to data
2. Sample (h \sim p(h|v))
3. Sample (v' \sim p(v|h))

Purpose:

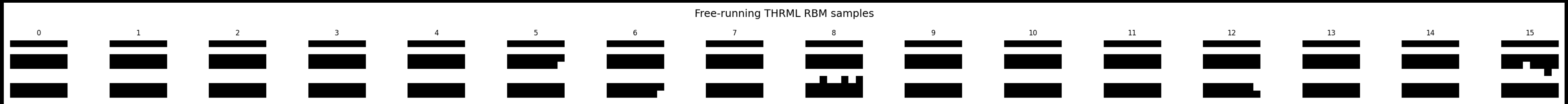
- Validate training
- Check encoder–decoder consistency
- Ensure stable attractor basins


Data vs One-Step Reconstructions
#0
#1
#2
#3
#4
#5
#6
#7
Original
Reconstruction

THRML Free-Running Sampling

- Start from random spins
- Run full Gibbs chain (no clamping)
- THRML explores the learned energy landscape
- Produces pure model samples

Shows true generative behavior of the RBM.


Free-running THRML RBM samples
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

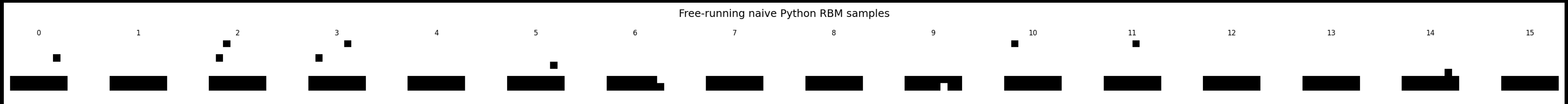
Naive Python Gibbs Sampling (Baseline)

Why?

- Validate correctness of THRML sampler
- Check if a hand-written sampler produces similar samples

Results:

- Python sampler produces Bars & Stripes
- Confirms correctness of learned energy landscape
- THRML behaves consistently with standard RBM sampling


Free-running naive Python RBM samples
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Why Three Sampling Modes?

1. Clamped Reconstructions

- ✓ Measure training quality
- ✓ Checks encoder/decoder mapping
- ✓ Local consistency

2. THRML Free-Running Sampling

- ✓ Explore full Boltzmann distribution
- ✓ Checks generative behavior
- ✓ Shows learned modes

3. Naive Gibbs Sampling

- ✓ Baseline sanity check
- ✓ Confirms THRML correctness
- ✓ Ensures consistent energy landscape

What We Learned

RBM Learning

- RBM trained via THRML achieves extremely low recon error
- Filters reflect dataset structure

Energy Landscape

- THRML free-running samples match Bars & Stripes
- Energy minima match dataset modes

Sampler Validation

- Python Gibbs sampler matches THRML outputs
- Confirms correctness of THRML's block sampler

THRML as a Gibbs Engine

Benefits

- Eliminates hand-written Gibbs loops
- Works for **arbitrary** factor graphs
- Supports multi-chain sampling
- Hardware-friendly (GPU/TPU acceleration)
- Clean integration with EBMs

Takeaway

THRML can serve as a **general-purpose thermodynamic sampling engine** for probabilistic models.

Conclusion

We successfully:

- Built a THRML-powered RBM
- Learned structured representations
- Generated high-quality samples
- Validated sampler correctness
- Demonstrated the feasibility of using THRML as a Gibbs engine

A strong foundation for larger EBMs and future thermodynamic computing systems.

Thank You

Questions?

Contact: Alen Ribić / @alenribic, Alessio Toniolo / @_alessiotoniolo