Accelerating Relative Entropy Coding with Space Partitioning

Jiajun He[1], Gergely Flamich[1], José Miguel Hernández-Lobato[1] [1]*University of Cambridge*

Check out our paper!

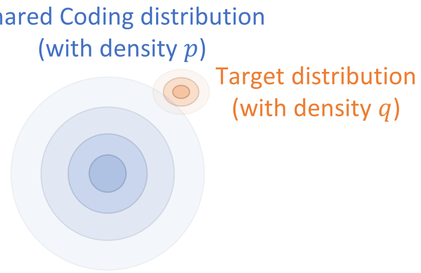
What is Relative Entropy Coding/Channel Simulation?

Encode a *random* sample from a distribution:

If $q := q(X|Y), p := \mathrm{E}_Y[q(X|Y)]$:

$$|M| \leq \mathbf{I}[X,Y] + \log(\mathbf{I}[X,Y] + 1) + O(1)$$

Overhead. If $\mathbf{I}[X,Y]$ is large, this terms becomes relatively small

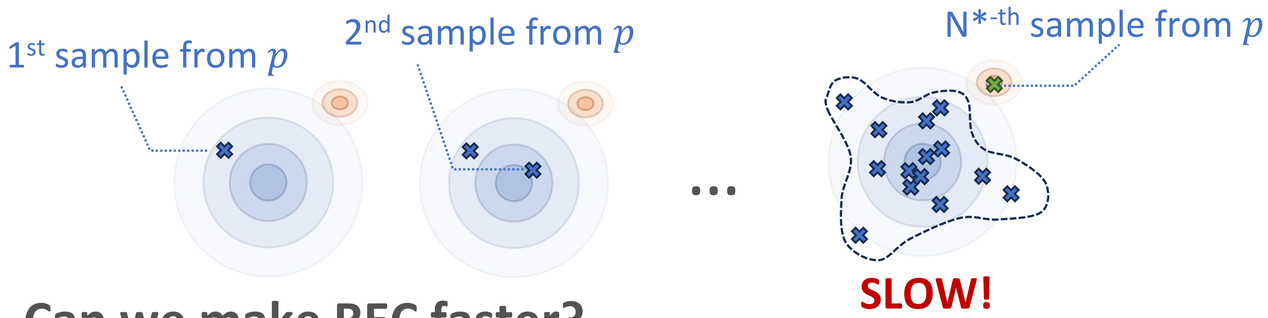


How does REC work?

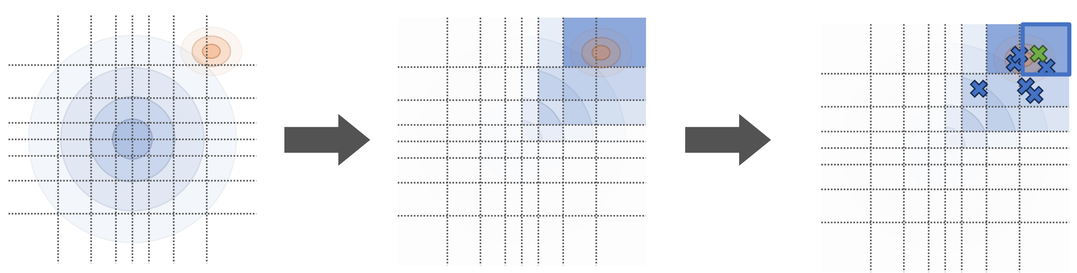
Poisson Functional Representation (PFR) algorithm

Ordered Random Coding (ORC) algorithm:

- draw samples from share p using a shared random seed;
- encode the index for the sample that matches q the best.



Can we make REC faster?



1. Partition space into J bins 2. Reweight each bin 3. Sample from reweighted prior!

- Does this mean we need to encode and transmit the reweighting?
- **No!** We only need to encode

(1) the bin index where the sample is located;

(2) the sample index in this bin

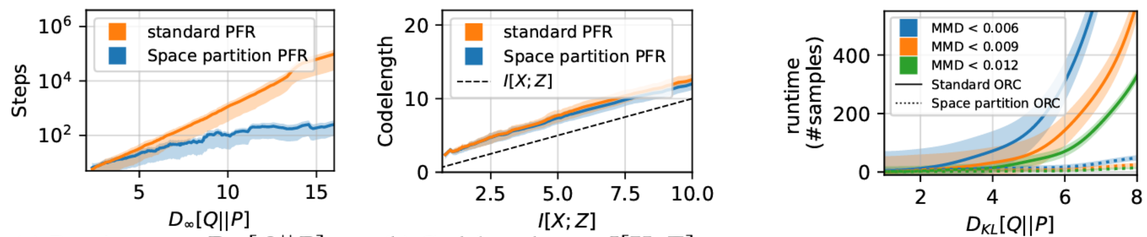
$$|M| \leq \mathbf{I}[X,Y] + \mathbf{E}_Y[\epsilon] + \log(\mathbf{I}[X,Y] - \log J + \mathbf{E}_Y[\epsilon] + 1) + O(1)$$

Overhead introduced by the two-part codes: $\epsilon = \mathbf{E}_{X\sim q}[\log J - \log \frac{q(X)}{p(X)}]$

Well-bounded for Gaussian $\mathbf{E}[\epsilon] \leq 0.9\sqrt{\mathbf{I}[X,Y]}$, or Uniform $\epsilon = 0$

Experimental Results

Toy example:



(a) Runtime w.r.t $D_\infty[Q||P]$. (b) Codelength w.r.t $I[\mathbf{X};\mathbf{Z}]$.

Figure 2: Comparing standard PFR and PFR with our proposed space partitioning algorithm on toy examples. Solid lines and the shadow areas represent the mean and IQR.

Figure 3: Comparing standard ORC and ORC with our proposed space partitioning algorithm on toy examples.

Apply proposed algorithm for data compression:

Compress CIFAR-10 Images with Bayesian Implicit Neural Representations:
Our proposed algorithm can handle $\mathbf{I}[X,Y]$ upto 48 bits, which requires 2^{48} samples by baselines