Accelerating Relative Entropy Coding with Space Partitioning

Jiajun He, Gergely Flamich, José Miguel Hernández-Lobato

Accepted In *NeurIPS 2024*

Check out our paper!

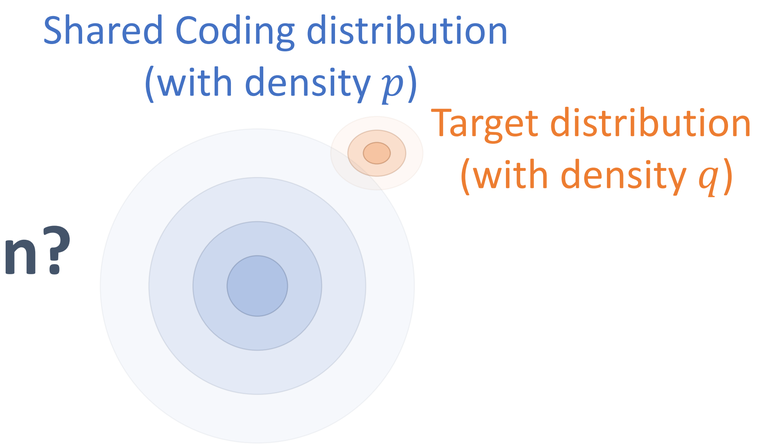
What is Relative Entropy Coding?

- How can you encode a *given* sample losslessly?

- Entropy Coding

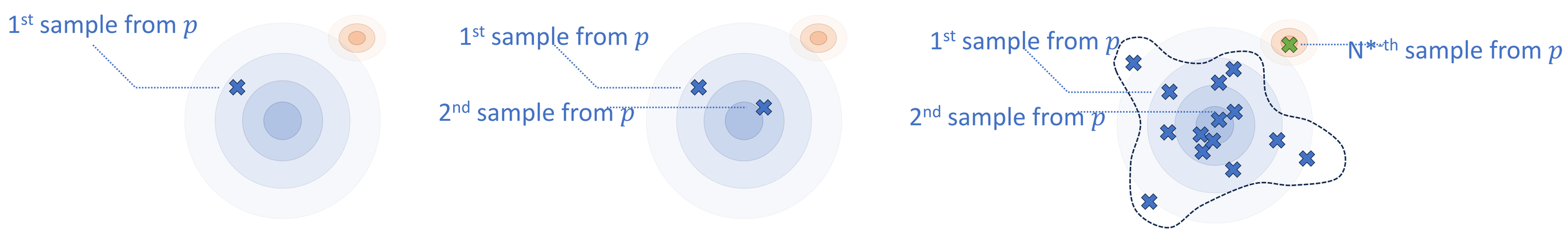
- How can you encode a *random* sample from a distribution?

- Relative Entropy Coding (REC) / Channel Simulation



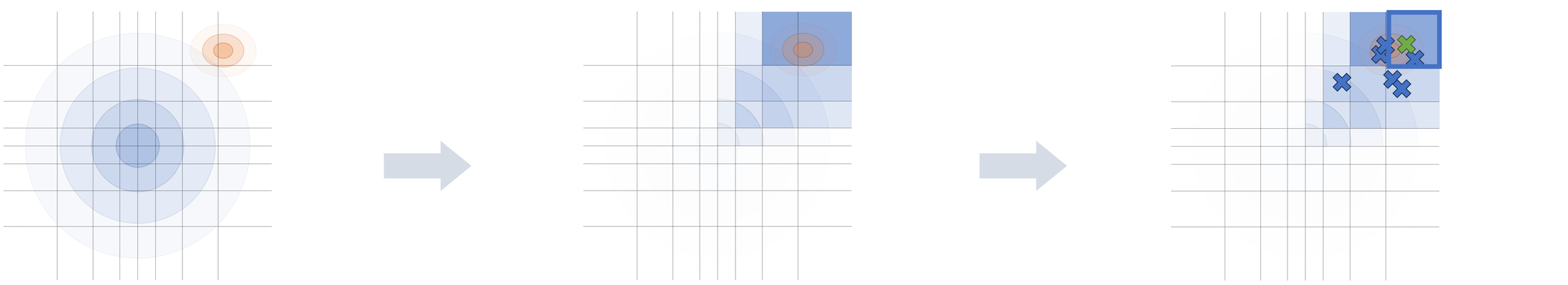
How does it work?

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



Can we make it faster?

SLOW!



Partition space into bins → Reweight each bin → Sample from reweighted prior!

- Does this mean we also need to encode the reweighting?
- **No!** We only need to encode
  (1) the bin index where the sample is located; (2) the sample index in this bin

Codelength?

Standard REC:

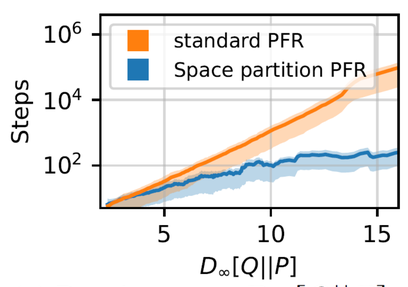
$$\leq \mathbf{I}[X,Y] + \log(\mathbf{I}[X,Y] + 1) + O(1)$$

Our Space-partitioning REC:

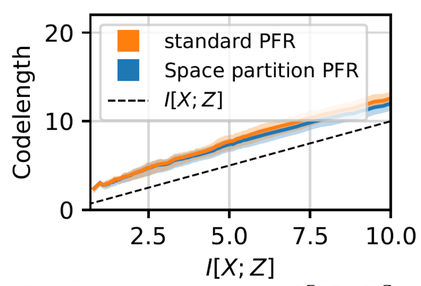
$$\leq \mathbf{I}[X,Y] + \mathbf{E}_Y[\epsilon] + \log(\mathbf{I}[X,Y] - \log J + \mathbf{E}_Y[\epsilon] + 1) + O(1)$$

Overhead introduced by the two-part codes

$$\epsilon = \mathbf{E}_{X\sim P_{X|Y}}[\log J - \log \frac{p_{X|Y}(X|Y)}{p_X(X)}]$$



(a) Runtime w.r.t $D_\infty[Q||P]$.



(b) Codelength w.r.t $I[\mathbf{X};\mathbf{Z}]$.