Learning Better **Lossless Compression** Using Lossy Compression

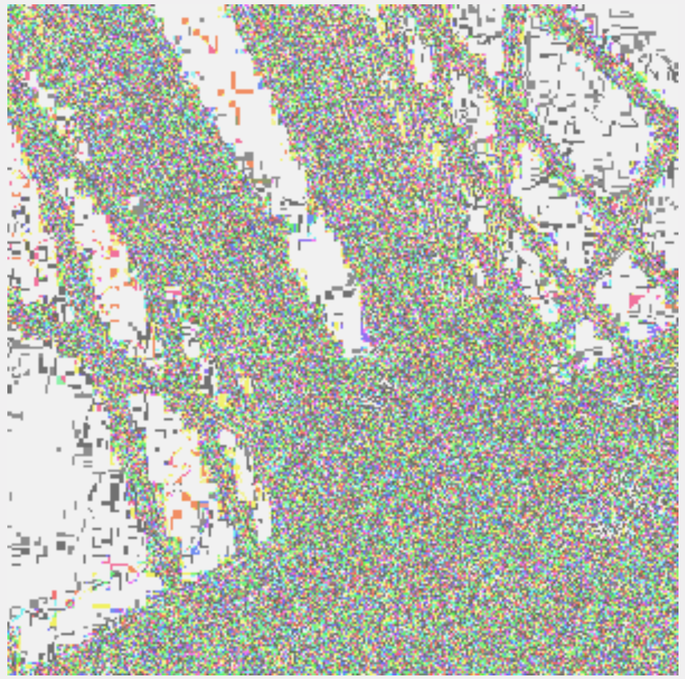
Fabian Mentzer
Luc Van Gool
Michael Tschannen

What and Why

Encode residual of lossy image compression algorithm BPG to do lossless compression.

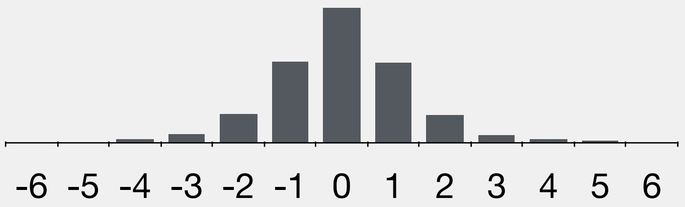
BPG x

BPG x_l



Residual $x - x_l$

Residual is centered closely around 0 →



Background: Arithmetic Coding

Set of symbols:

$\sim p$

Setup: *Finite set of symbols, the probability of a symbol occurring is given by the probability distribution* p.

We can encode a stream of symbols losslessly to a bistream with ***Arithmetic Coding***.

Background: Arithmetic Coding

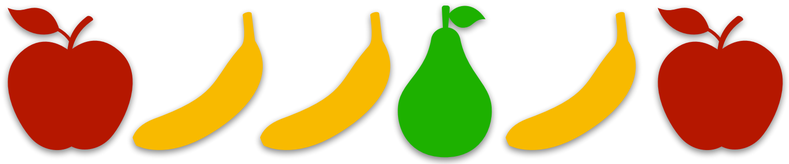
Set of symbols:

$\sim p$

Setup: *Finite set of symbols, the probability of a symbol occurring is given by the probability distribution* p.

We can encode a stream of symbols losslessly to a bistream with ***Arithmetic Coding***.

Stream of symbols:



i.i.d. $\sim p$

This is an example stream. $p($ 🍌 $)$ *is higher, so we want to use fewer bits when encoding a banana symbol! That's what arithmetic coding does.*

Background: Arithmetic Coding

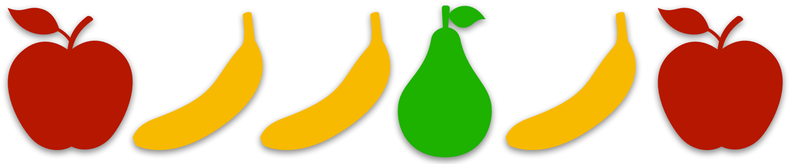
Set of symbols:

$\sim p$

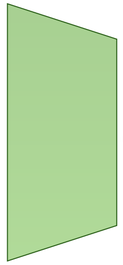
Setup: *Finite set of symbols, the probability of a symbol occurring is given by the probability distribution* p.

We can encode a stream of symbols losslessly to a bistream with ***Arithmetic Coding***.

Stream of symbols:



i.i.d. $\sim p$



→ bitstream

arithmetic encoder

Arithmetic coding assigns a bit-sequence to each symbol, such that more likely symbols get shorter sequences. We encode the stream of symbols into one bitstream.

Background: Arithmetic Coding

Set of symbols:

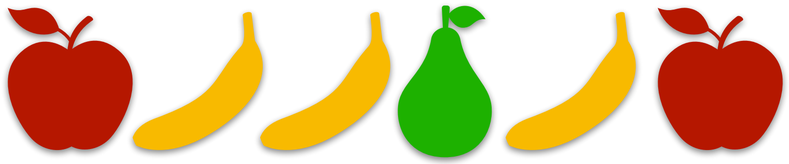
$\sim p$

***Setup**: Finite set of symbols, the probability of a symbol occurring is given by the probability distribution p.*

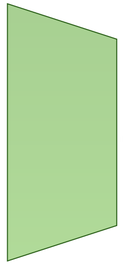
*We can encode a stream of symbols losslessly to a bistream with **Arithmetic Coding**.*

Decoding recovers the stream

Stream of symbols:

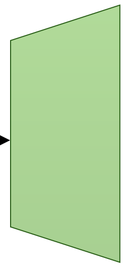


i.i.d. $\sim p$

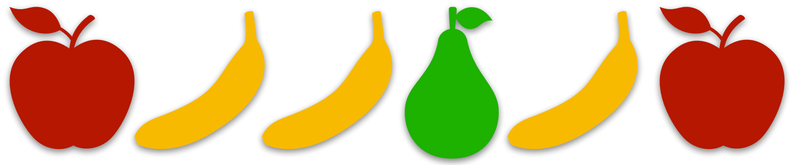


arithmetic encoder

bitstream



arithmetic decoder



Background: Arithmetic Coding

Set of symbols:

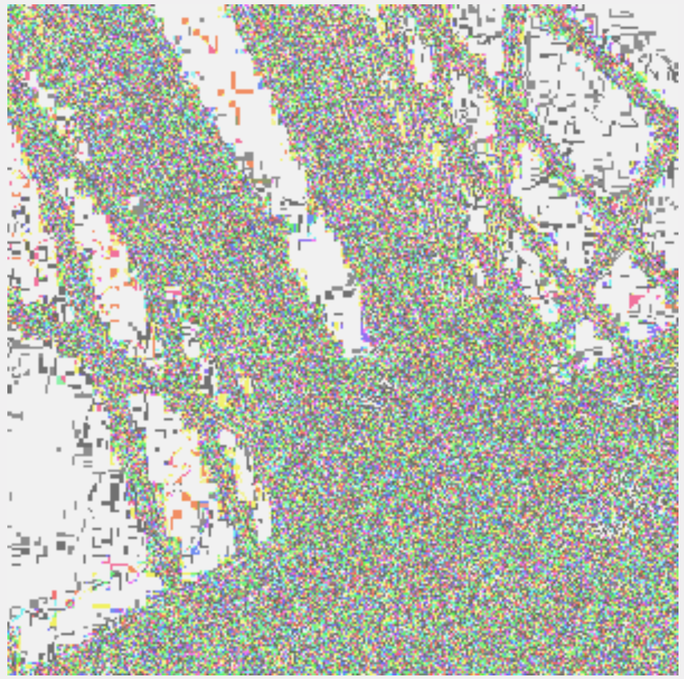
$\sim p$

***Problem**: In general, we don't know p. So we learn a model p' for it by minimizing the cross-entropy $H(p, p')$. This is the same as minimizing the negative log likelihood of p' over mini-batches.*

Lossless Compression

BPG x

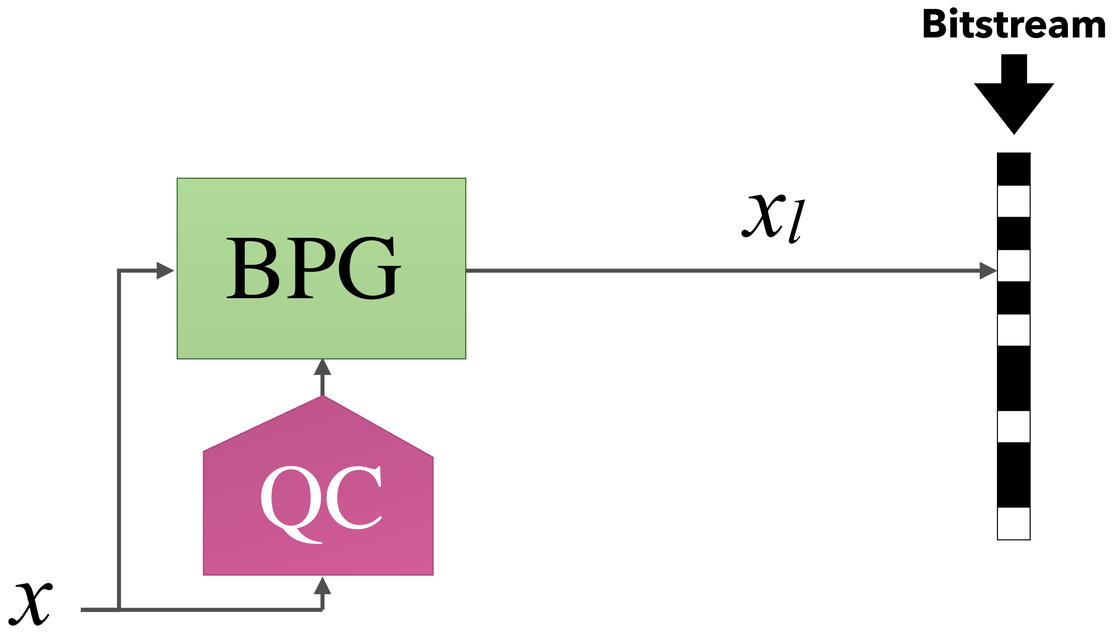
BPG x_l



Residual $x - x_l$

So, to encode this residual, all we need is a model $p(r)$**.**

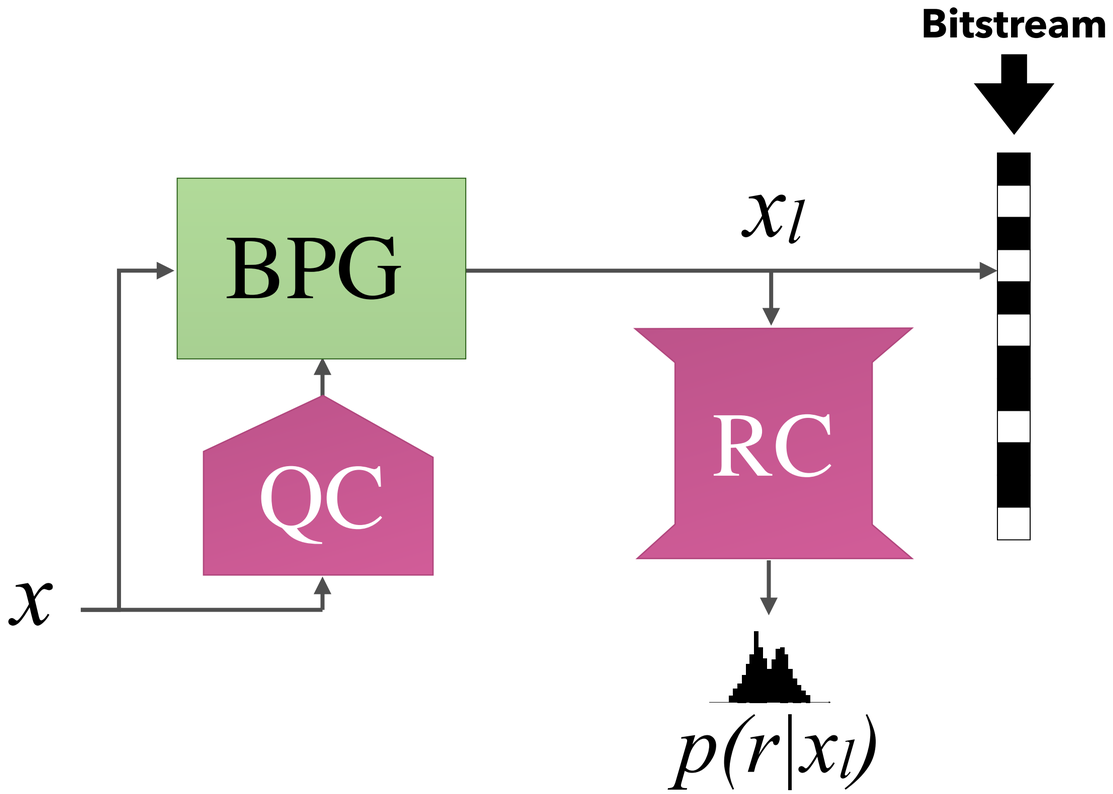
How



Encoding

1. Get quality factor Q for BPG from the **learned Q-classifier (QC)**.
2. Encode input x with BPG using that Q, get x_l and save to bitstream.

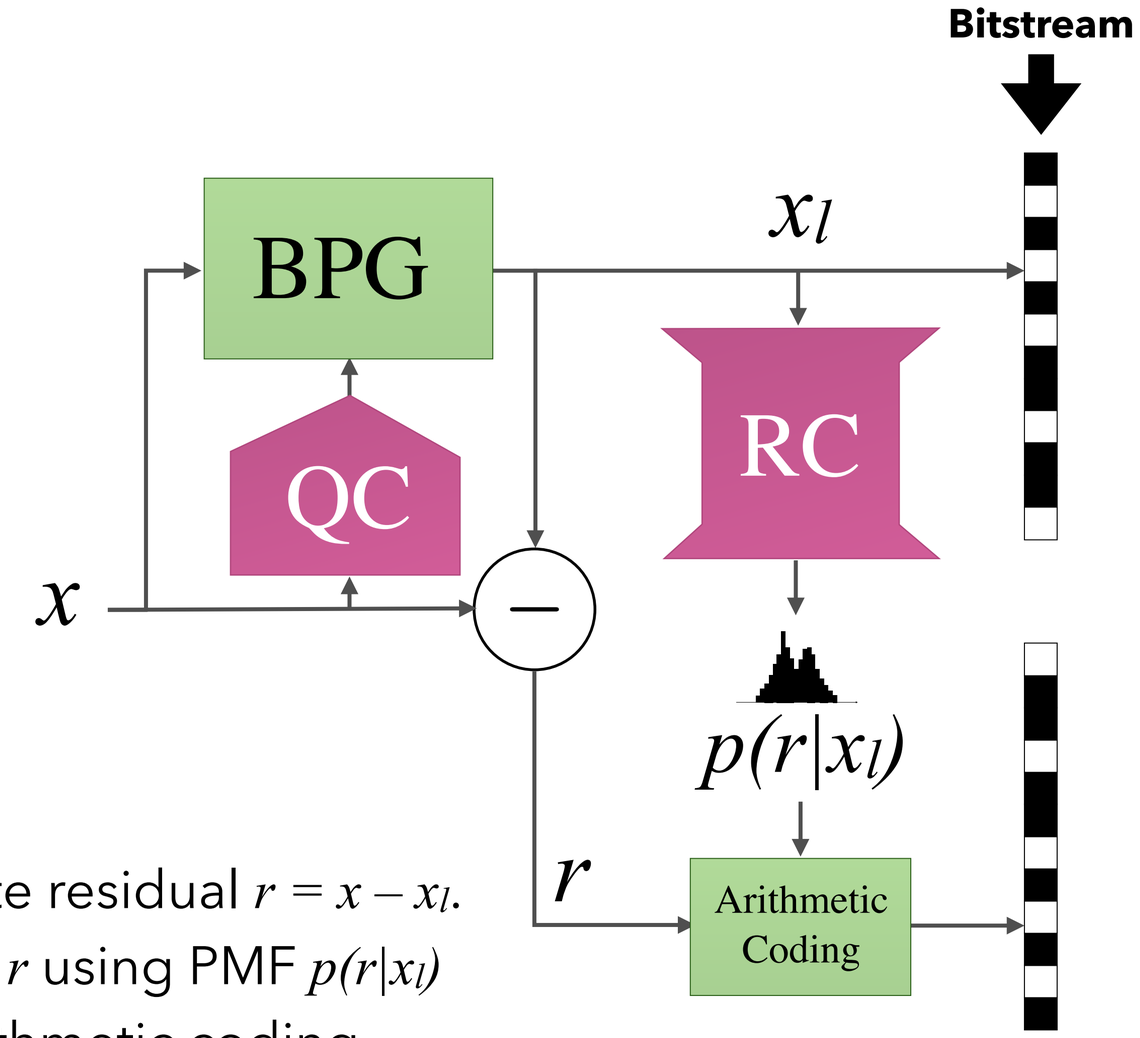
How



Encoding

3. Feed x_l to the **learned Residual Compressor (RC)**, which predicts the conditional PMF $p(r|x_l)$.

How



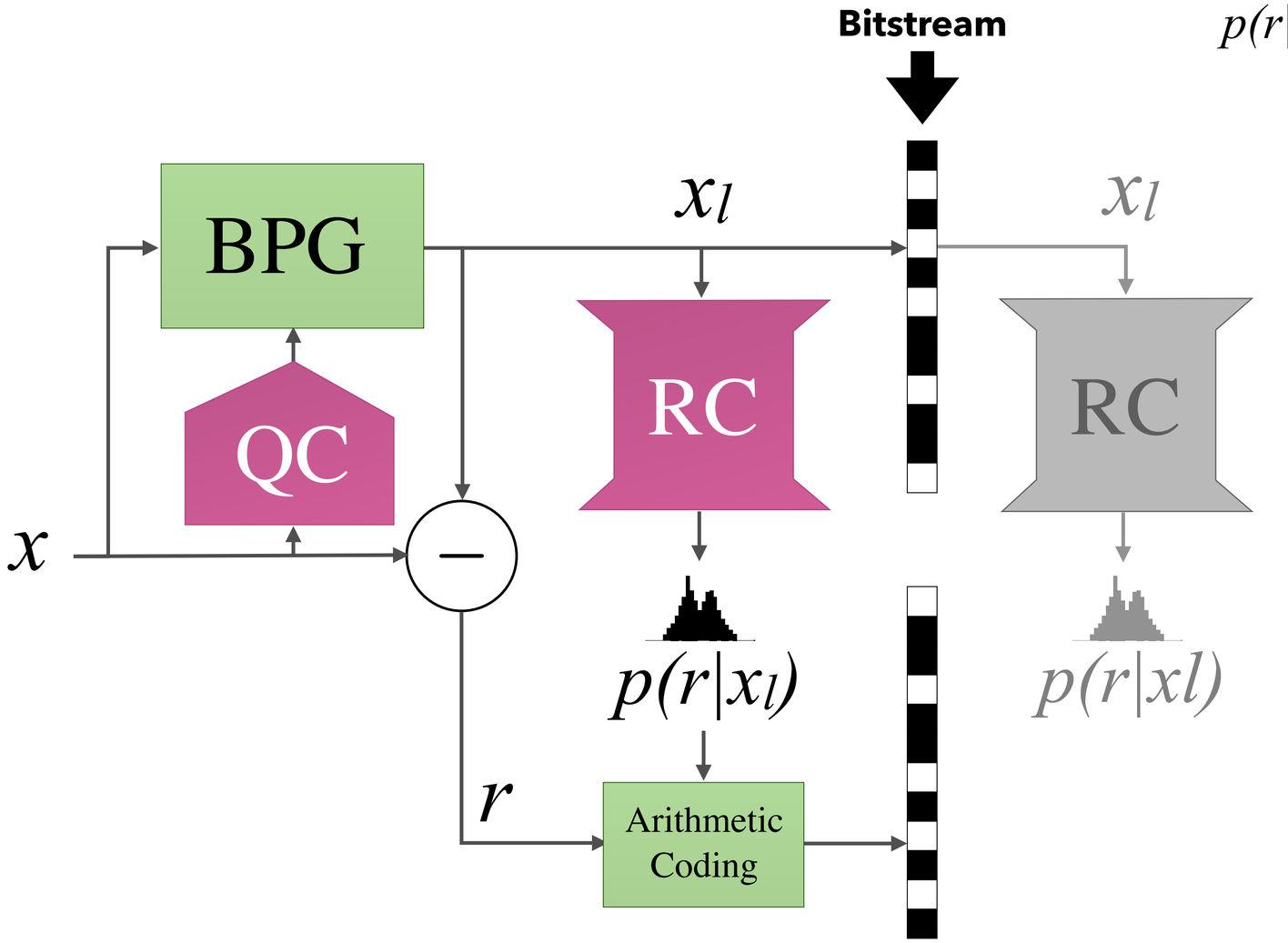
Encoding

4. Calculate residual $r = x - x_l$.
5. Encode r using PMF $p(r|x_l)$ with Arithmetic coding.

How

Decoding

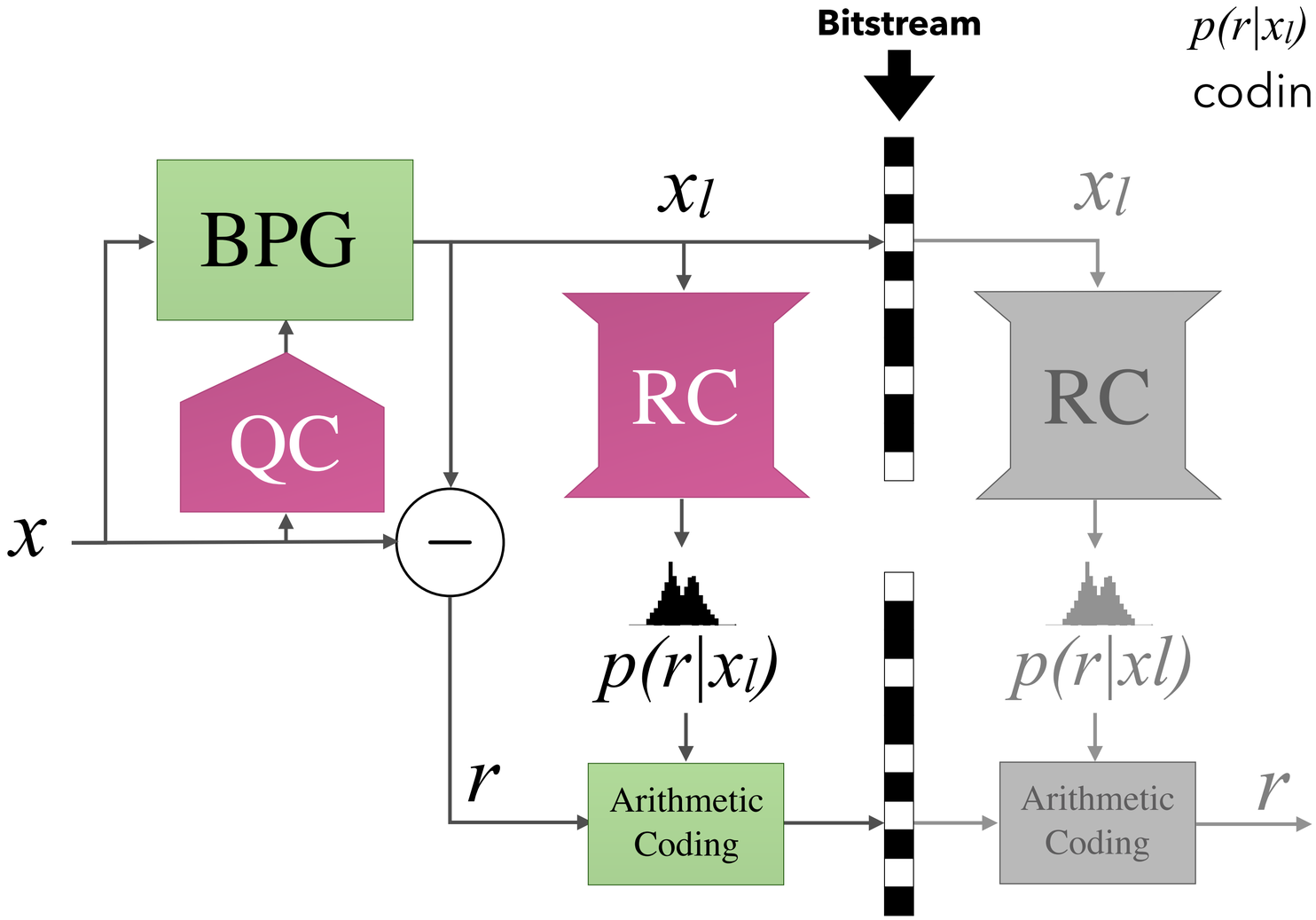
1. Obtain x_l from the bitstream.
2. Feed x_l into RC to get PMF $p(r|x_l)$.



How

Decoding

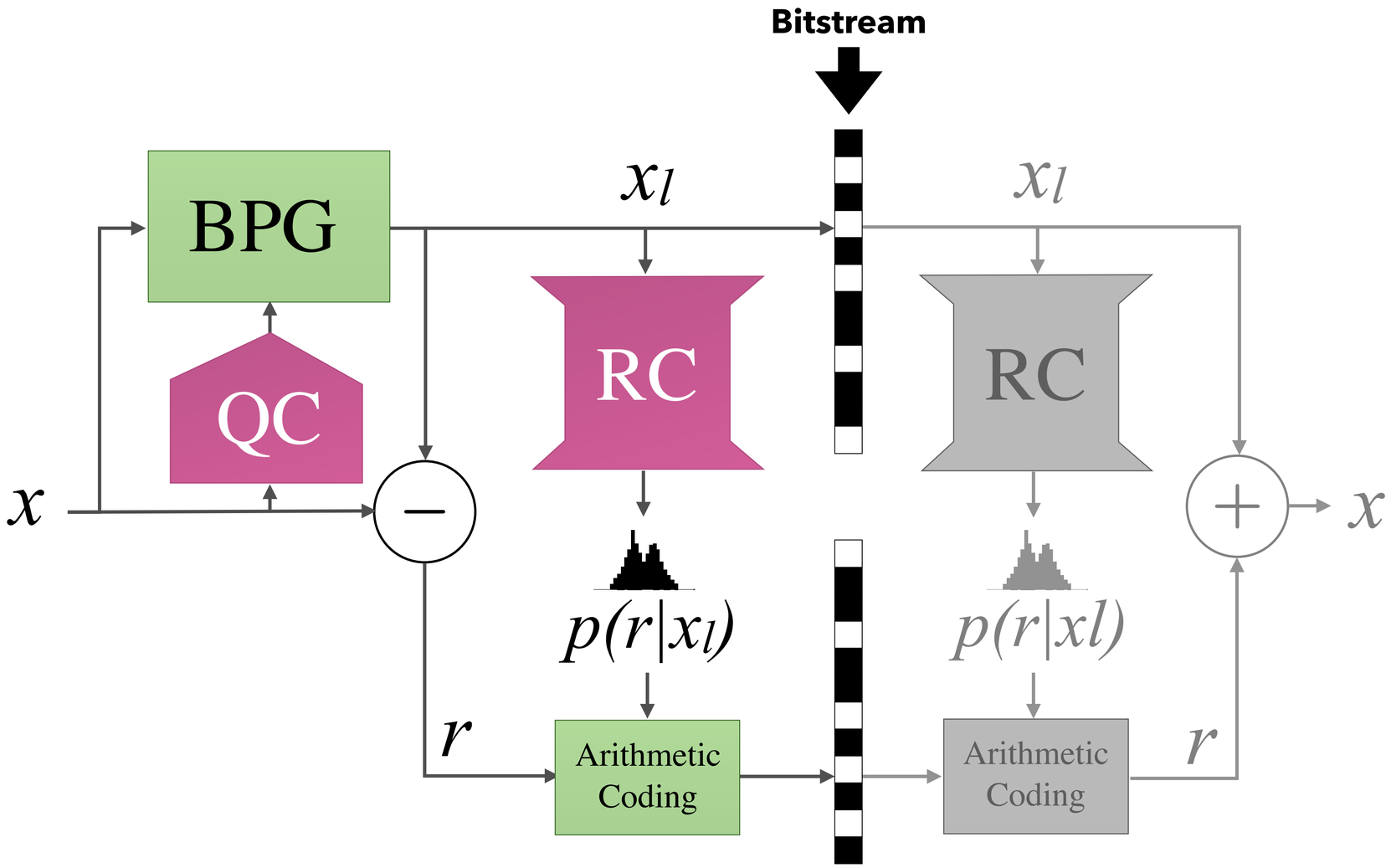
3. Decode r from the bitstream, using the PMF $p(r|x_l)$ and arithmetic coding.



How

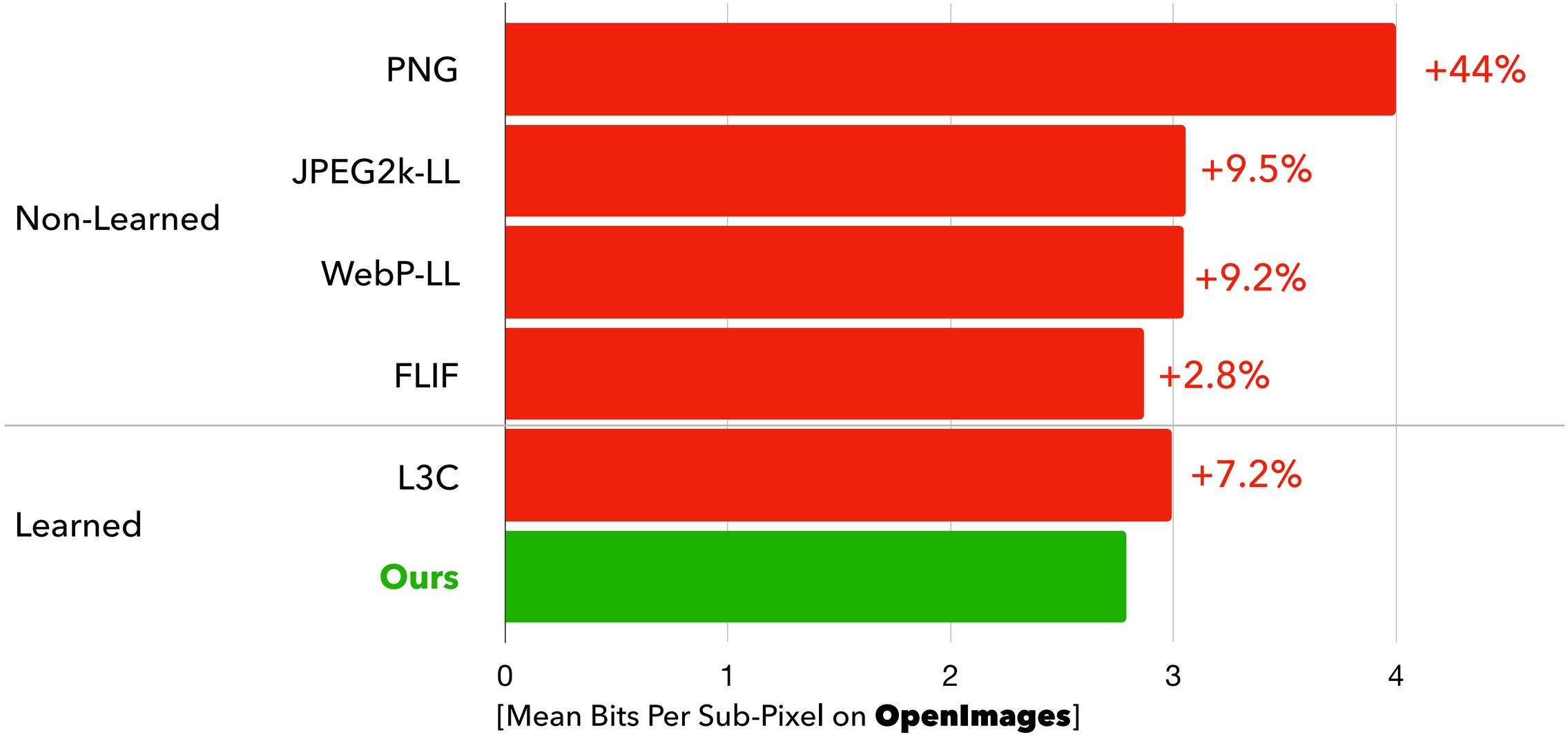
Decoding

4. Obtain $x = r + x_l$.



Results

State of the Art on Open Images, where we train.



Results

Better than previous learned method (L3C) on other datasets