AstroCompress:
A benchmark dataset for multi-purpose compression of astronomical data

Tuan Truong*, Rithwik Sudharsan*
Yibo Yang, Peter Ma, Ruihan Yang
Stephan Mandt, Joshua Bloom

*Equal contribution first authors. Direct correspondence to:
<tuannt2@uci.edu>, <rithwik@berkeley.edu>

01: The Problem

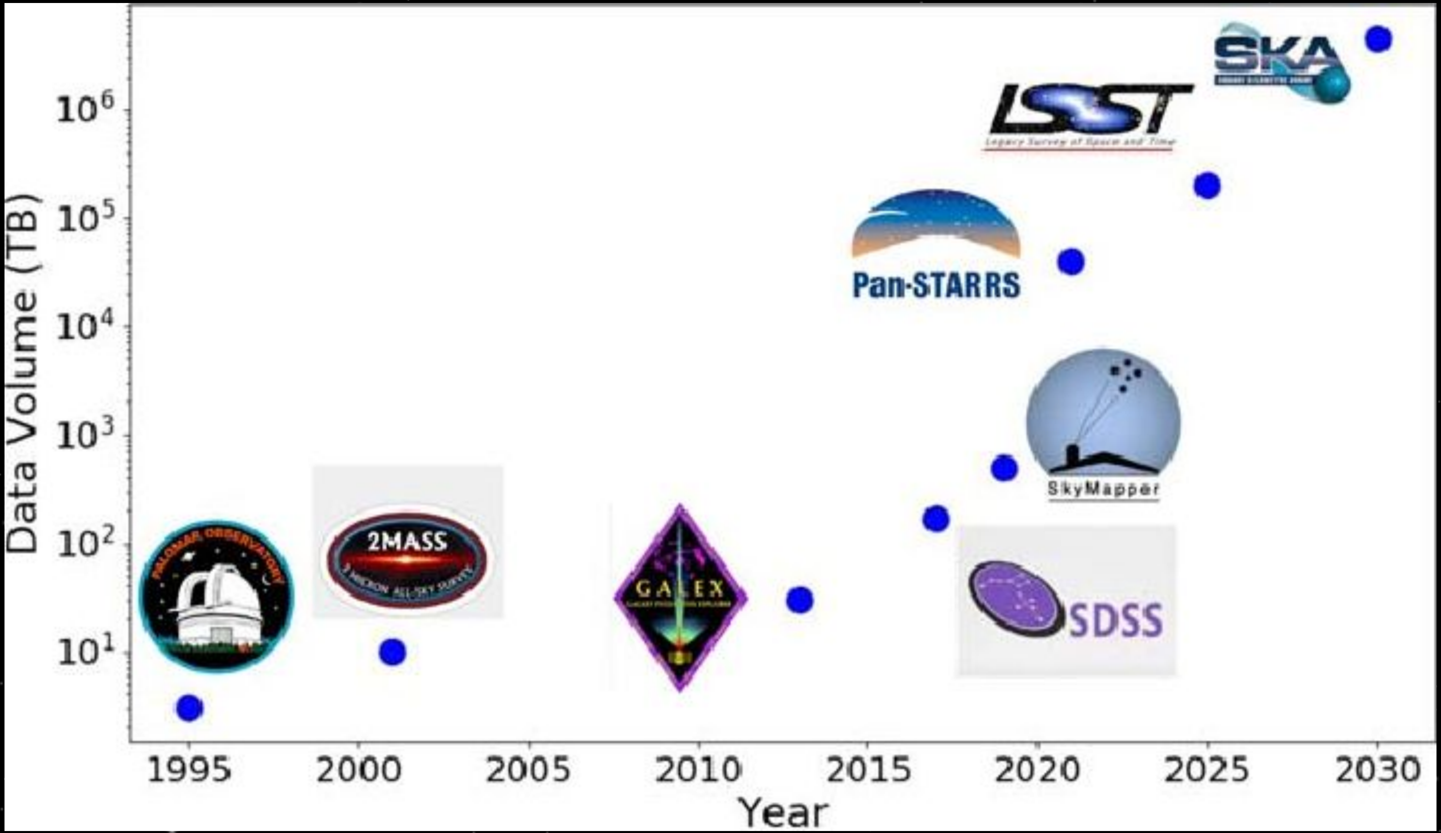
The state-of-the-art James Webb Space Telescope (JWST) has:

- ~65 GB storage capacity
- ~60 GB is downlinked per day at 28 Mbps
- Contact with Earth exists only 8 hours per day → lots of downtime

JWST is at L2, but even satellites in LEO (like CuRIOS-ED) struggle with bandwidth.

There certainly is a desire to collect more data: at the extreme, ground-based telescopes like the Square Kilometer Array will collect ~1 exabyte per year / 3 Tbps, which is 40,000x JWST.

- Transporting this data even on the ground is a challenge, so compression is needed!



Data Volume (TB)
10^6
10^5
10^4
10^3
10^2
10^1
1995
2000
2005
2010
2015
2020
2025
2030
Year
PALOMAR OBSERVATORY
2MASS
2 MICRON ALL-SKY SURVEY
GALEX
Pan-STARRS
LSST
Legacy Survey of Space and Time
SKA
SkyMapper
SDSS

SDP Challenges:

Input
- 400 Gbyte/s INGEST

Distribute
- 400 million tasks in the graph
- Half an Exaflop/s total peak

Generate (and destroy)
- 1.3 ZettaBytes intermediate data products

Preserve and ship
- 1 PetaByte per day of Science Data Products
- Each cube up to a few PBytes

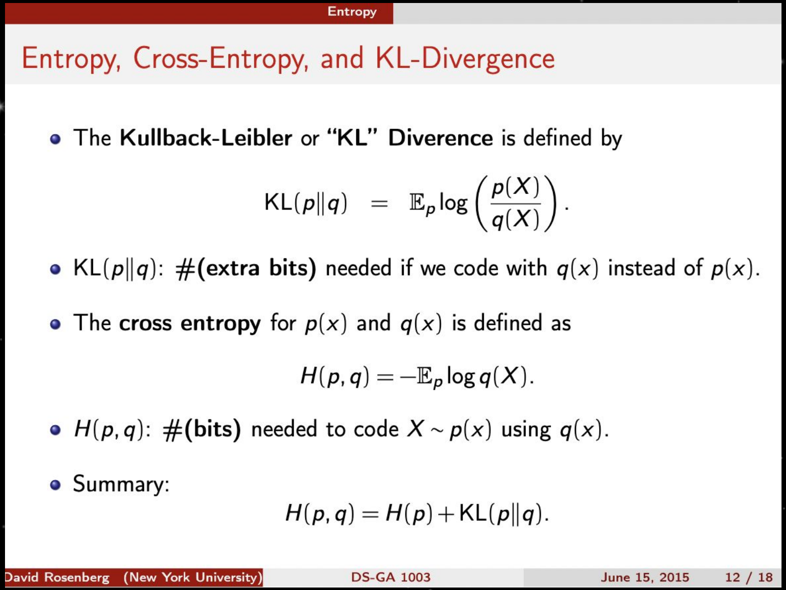
SKA SQUARE KILOMETRE ARRAY SDP SCIENCE DATA PROCESSOR

UNIVERSITY OF CAMBRIDGE

02: Shannon's Theorem

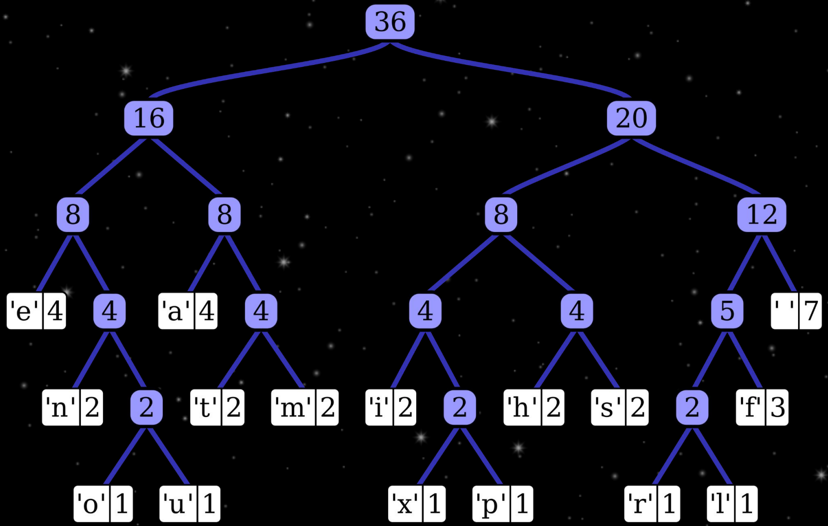
Min compression bitrate = entropy of X.

Density estimation = lossless compression.

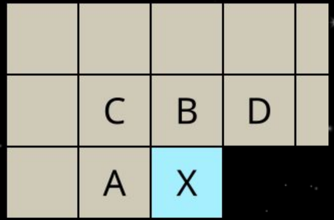


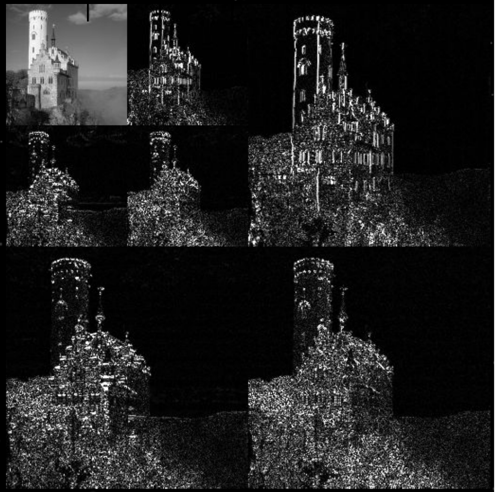
Entropy, Cross-Entropy, and KL-Divergence

- The **Kullback-Leibler** or **"KL" Diverence** is defined by
$$\mathrm{KL}(p\|q) \quad = \quad \mathbb{E}_p \log\left(\frac{p(X)}{q(X)}\right).$$
- $\mathrm{KL}(p\|q)$: **#(extra bits)** needed if we code with $q(x)$ instead of $p(x)$.
- The **cross entropy** for $p(x)$ and $q(x)$ is defined as
$$H(p,q) = -\mathbb{E}_p \log q(X).$$
- $H(p,q)$: **#(bits)** needed to code $X \sim p(x)$ using $q(x)$.
- Summary:
$$H(p,q) = H(p) + \mathrm{KL}(p\|q).$$



03: Existing Codecs





RICE (used in astronomy today)

- Lightning fast, almost no spatial information used, entropy coding assuming Geometric distribution

HCOMPRESS, JPEG-2000

- Wavelet transforms + entropy coding
- Sometimes use other things like run-length encoding and LZ77

JPEG-LS

- Autoregressive prediction w/ context modeling + entropy coding of residual

JPEG-XL

- All of the above on steroids

04: Our Dataset

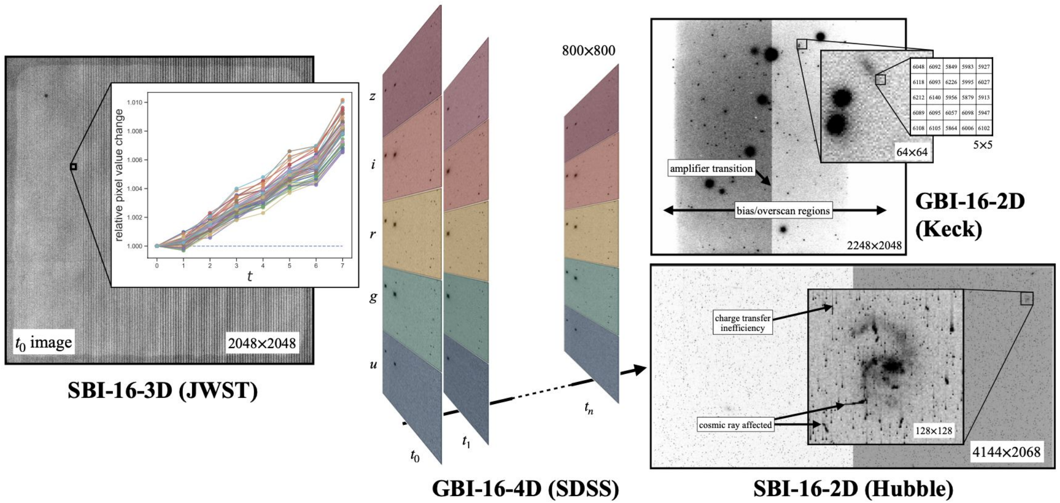


Figure 1: Depiction of salient features in the AstroCompress corpus using representative images from each dataset. Inset to the JWST t_0 (first) image are the value changes in time for a small sample of pixels. In SDSS there are 5 filtered images per observation epoch, up to a variable number n observations in the same portion of the sky. The inset of Hubble zooms in on a spiral galaxy, showing cosmic ray hits (black) and charge transfer inefficiency, causing vertical flux smearing. The actual pixel values in Keck are shown for a zoomed-in $5{\times}5$ pix^2 region.

05: ML Methods: IDF

- Normalizing flows
- Integer operations only, to avoid lossy quantization at the end
- Factored logistic mixture distribution enforced on z
- During decoding, we only have to decode the latent z based on the logistic mixture
- Incentive to decorrelate z_i is implicit in the "factored" distribution.
  That's why we don't just compress x instead of z
- Jacobian = 1
- Exact density p(z)=p(x) is known

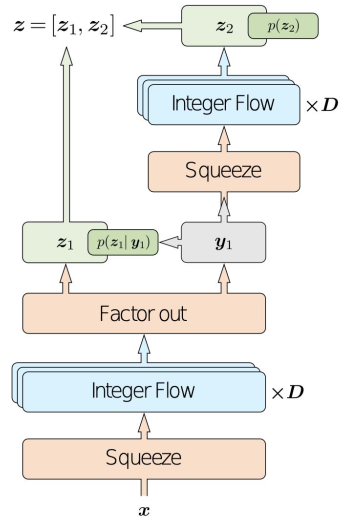
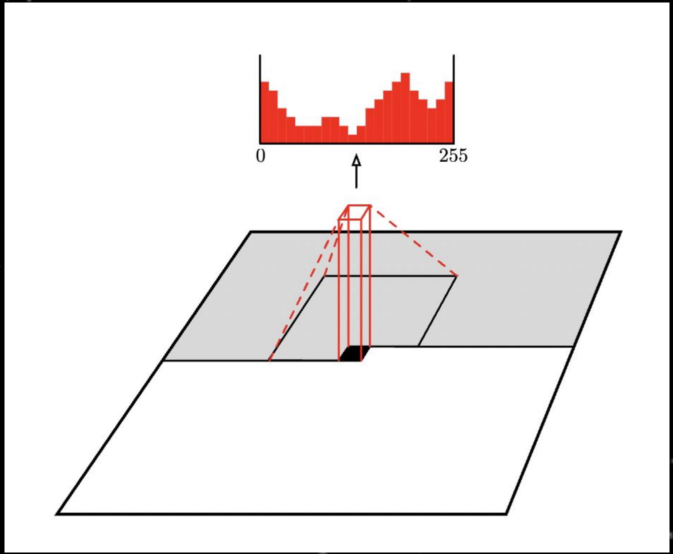


Figure 4: Example of a 2-level flow architecture. The squeeze layer reduces the spatial dimensions by two, and increases the number of channels by four. A single integer flow layer consists of a channel permutation and an integer discrete coupling layer. Each level consists of D flow layers.

05: ML Methods: PixelCNN++

- Autoregressive
- Predict distribution over possible values
    - Again, logistic mixture (discretized)
- Simple, similar to JPEG-LS
- Parallelizable encoding, slow decoding
- Exact density p(x) is known



05: ML Methods: L3C

- Basically an autoregressive model that models **features at different scales** autoregressively rather than every single pixel
- Thus, it's super fast
- Parallels to U-Net: D(k) passes skip connections to D(k-1)
- It's cool and creative but not that effective

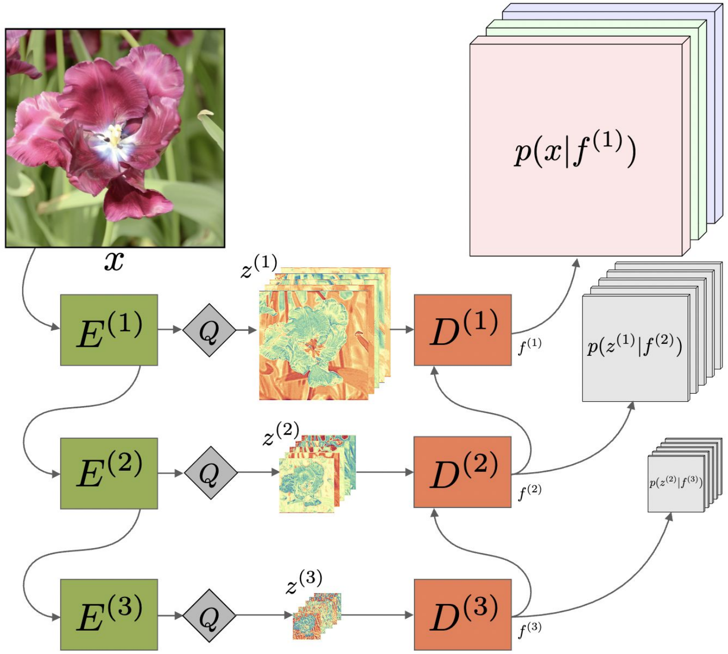
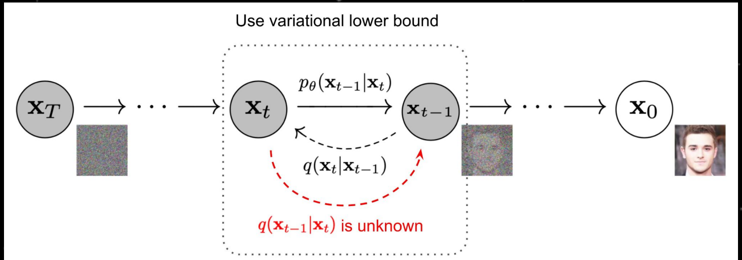


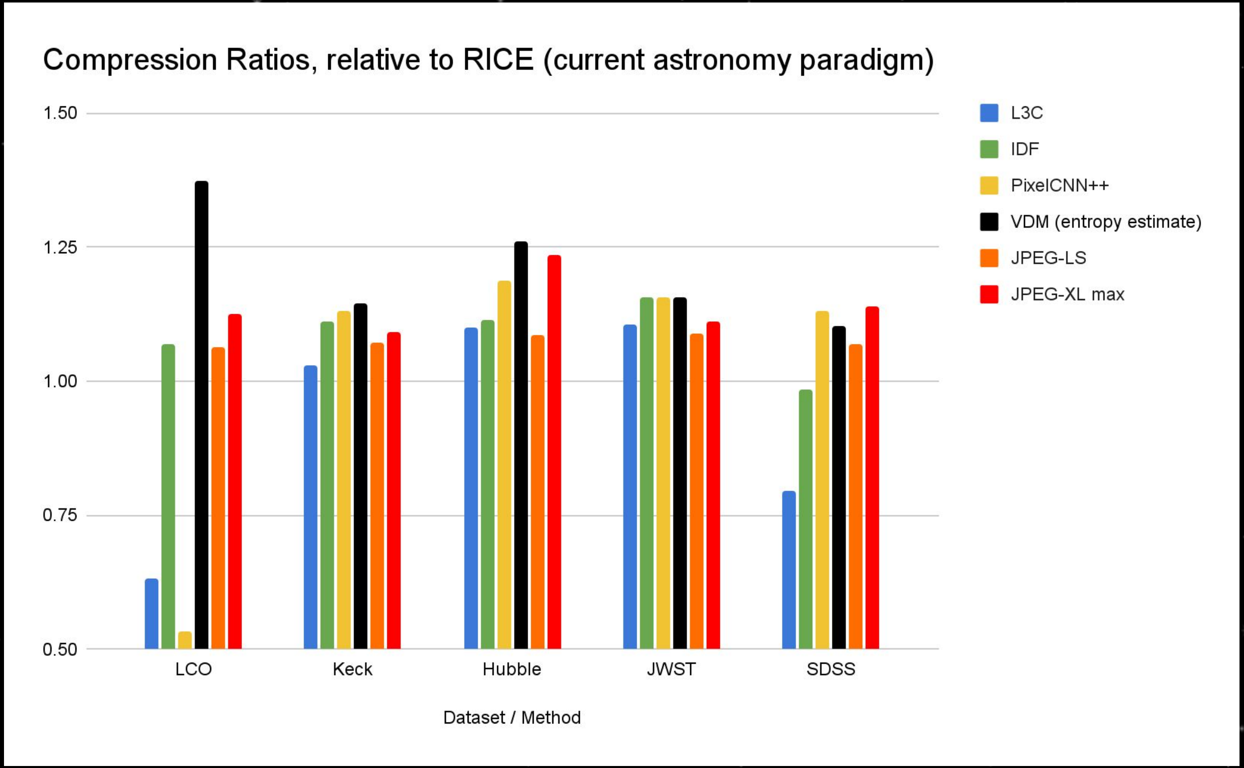
Figure 1: Overview of the architecture of L3C. The feature extractors $E^{(s)}$ compute quantized (by Q) auxiliary hierarchical feature representation $z^{(1)}, \ldots, z^{(S)}$ whose joint distribution with the image x, $p(x, z^{(1)}, \ldots, z^{(S)})$, is modeled using the non-autoregressive predictors $D^{(s)}$. The features $f^{(s)}$ summarize the information up to scale s and are used to predict p for the next scale.

05: ML Methods: VDM

- Imagine 1 VAE… a VDM is ~a bunch of stacked VAE's where each one is predicting the slightly denoised image instead of reconstructing the whole image
- Issue: latent is always a sample of a random distribution. Need "bits-back" coding to actually do compression
- Transmit the noisiest latent z_1 with prior p(z_1)
- Encode residuals at each stage of VDM with p(current z | previous z)
- It's like a diffusion model where we always have the exact correct image at each stage
- Extremely slow (but optimizations exist)



06: Results


Compression Ratios, relative to RICE (current astronomy paradigm)
1.50
1.25
1.00
0.75
0.50
LCO
Keck
Hubble
JWST
SDSS
Dataset / Method
L3C
IDF
PixelCNN++
VDM (entropy estimate)
JPEG-LS
JPEG-XL max

06: Results

Codec	SDSS-2D (800x800)	Hubble (4144x2068)
IDF	0.42 ± 0.01	6.03 ± 0.24
L3C	5.18 ± 1.04	73.04 ± 2.36
PixelCNN++	1.48 ± 1.05	20.49 ± 0.18
JPEG-XL max	3.14 ± 0.14	87.76 ± 13.30
JPEG-XL default	0.06 ± 0.002	0.91 ± 0.07
JPEG-LS	0.02 ± 0.0002	0.316 ± 0.04
JPEG-2000	0.09 ± 0.003	1.76 ± 0.11
RICE	0.008 ± 0.0002	0.12 ± 0.02
VDM	2301 ± 229.2	33072 ± 19.8

Table 3: Compression (encoding) runtime (in seconds/image) on the SDSS-2D and Hubble datasets. For neural methods, we measure the time for evaluating the likelihood under the model without entropy coding.

		Evaluation Set			
Training Set	LCO	Keck	Hubble	JWST-2D	SDSS-2D
LCO	**2.83**	1.01	1.09	0.84	2.31
Keck	2.70	**2.05**	2.20	1.19	3.02
Hubble	0.67	0.94	2.94	1.22	0.69
JWST-2D	1.46	1.45	1.47	**1.44**	1.50
SDSS-2D	2.27	1.24	1.75	1.02	2.91
All data	2.82	1.87	**2.98**	1.38	**3.18**

Table 2: IDF generalized performance across single-frame datasets. Rows indicate train set; columns indicate test set. Bold indicates best in test set; underline indicates second-best.

07: Analysis

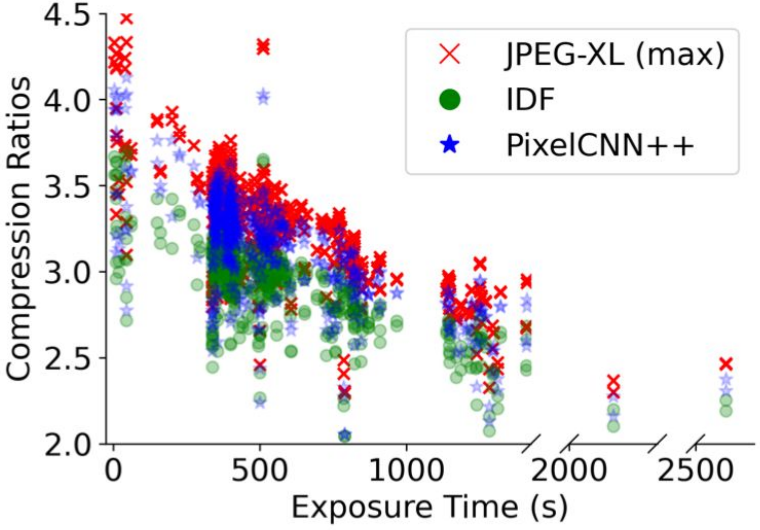


Figure 2: Hubble exposure times plotted against compression ratios using various algorithms. Longer exposure times tend to induce more incompressible noise and, hence, reduce compression ratios.

07: Analysis

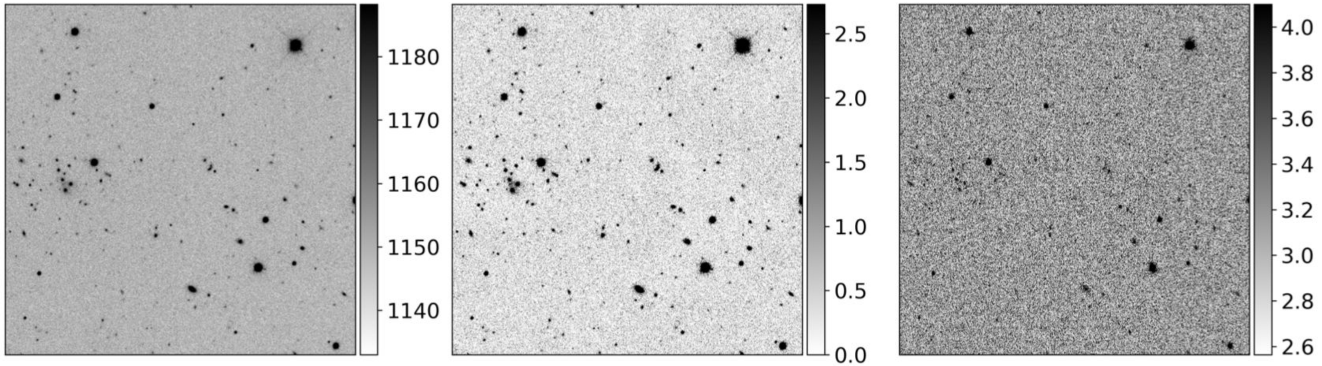
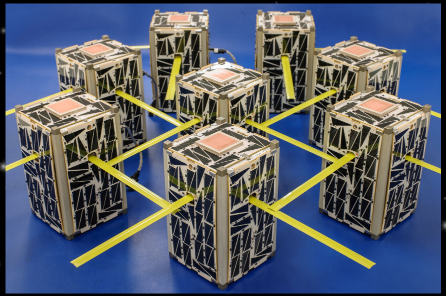


Figure 3: From left to right, an example SDSS-2D image: raw image, SNR heatmap, and PixelCNN++ bitrate heatmap. Colors are z-score normalized for visualization; colorbars indicate true values.

08: Future Directions

- Near-lossless compression
- CuRIOS satellite
- Autoregressive models' encode-decode time tradeoff
- More work on 3D / 4D data cube compression

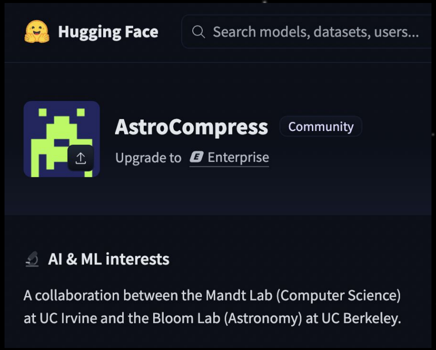


Published! (to an ML Audience)

Under review as a conference paper at ICLR 2025

ASTROCOMPRESS: A BENCHMARK DATASET FOR MULTI-PURPOSE COMPRESSION OF ASTRONOMICAL IMAGERY

Anonymous authors
Paper under double-blind review



External sources can be found by clicking any image in this slides file, which can be found at: https://github.com/tuatruog/AstroCompress