You can build 200 trees per second on v5-all with my cuda extra tree booster (on an A100,) and it can fit a dataset several times larger.

https://drive.google.com/file/d/1EU0a4mMLyUBCBLdjIS3_qEM85bHERtKi/view?usp=sharing

It uses some obvious optimizations like bit-packing and quantized gradients, but the more interesting one is that it builds layers (of different trees) at each depth simultaneously allowing for efficient packing of node index bits and full utilization of shared memory to accomplish more per global memory read.

This strategy requires layer-wise feature sampling which tends to prefer fewer features per layer than tree-wise sampling as trees see a greater variety of features overall than they do in one layer. My guess is that layer-wise sampling should perform better as it enables greater variety in trees.

This is the validation diagnostic when I attempt to reproduce the procedure of Numerai's example preds with an ensemble of Teager and Cyrus and retraining at eras 470, 626, 783, 939, and 1078 (no neutralization).

[

image

1053×620 43.2 KB

](https://forum.numer.ai/uploads/default/original/2X/f/f77425fdd8e2163ac99e77ad296a8095d9094961.png)

You can verify the result with this script that runs in about an hour:

https://drive.google.com/file/d/1xG0oEprBKI4hikY1xGdze2HVNI4sicWi/view?usp=sharing

This is still a fairly rough draft; it doesn't support sample weight and won't support depths greater than 7 until I re-write it in pycuda or find a way to adjust the shared memory carve-out in numba.