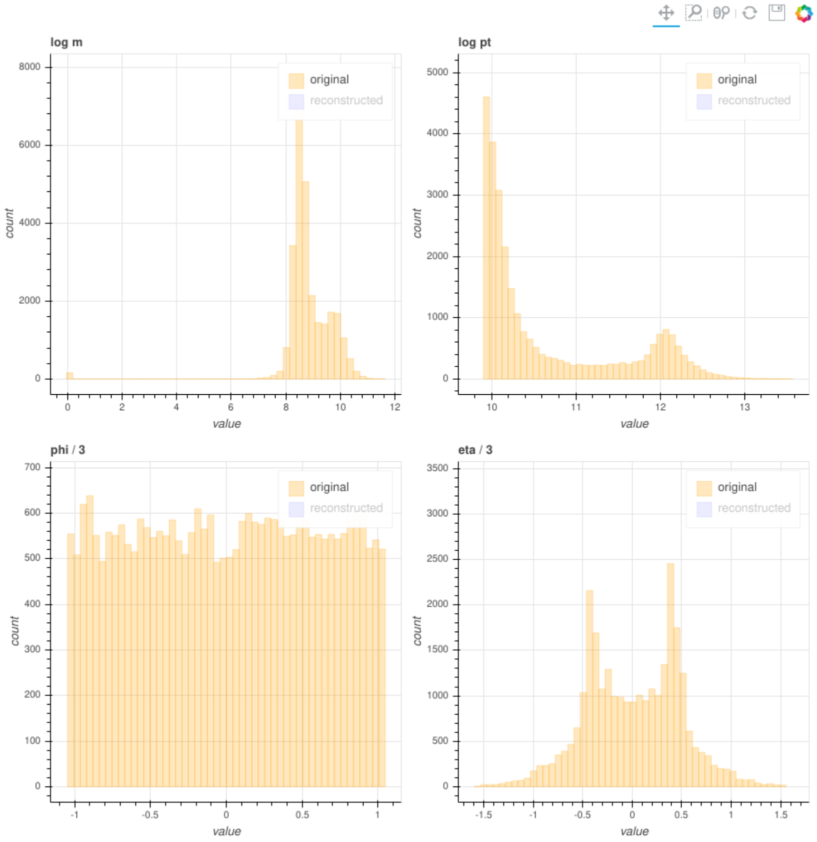
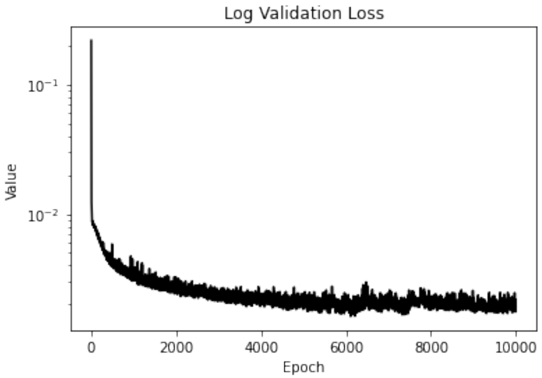
Original data

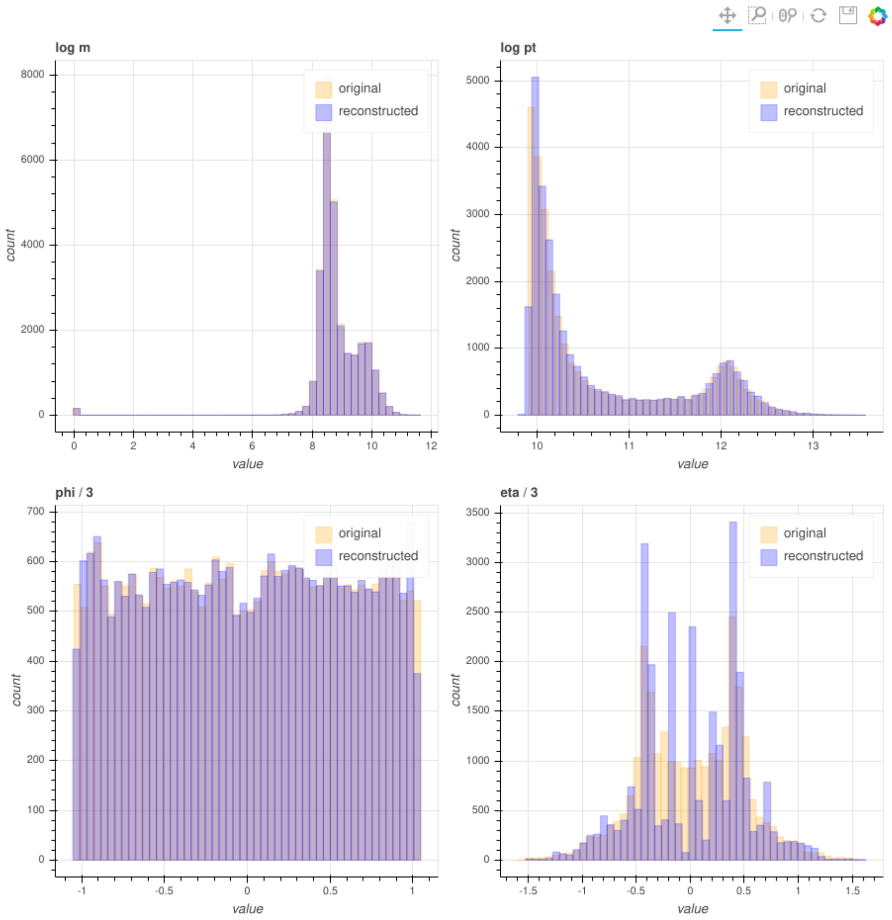
- 111778 data points for training, 27945 for validation
- Each point representing 4 variables
- Data was processed so that all variables are approximately with the same ranges
- Goal: use neural networks (autoencoders) to compress the 4 variables into 3 latent ones



Reconstructed data

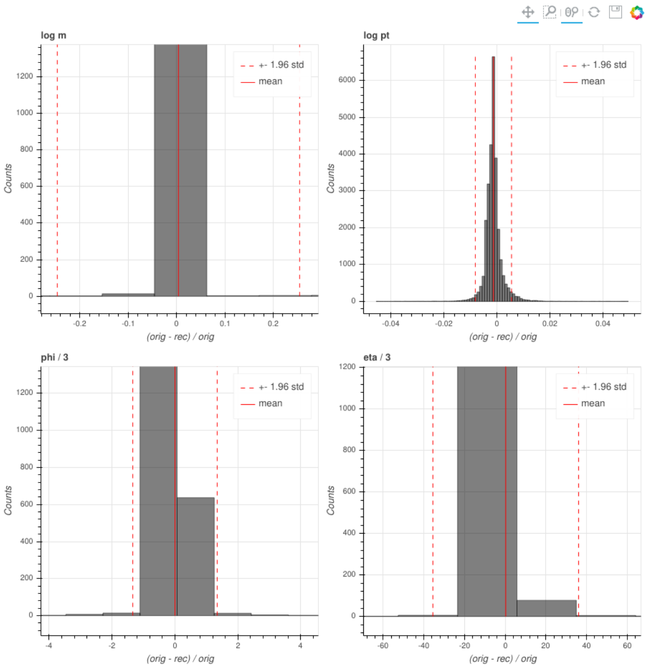
- Captured well the bimodality of `pt` and the uniformity of `phi`
- Failed to capture the distribution of `eta`
- Not/very slightly overfitting after 10000 epochs
  - Larger network could be used
  - Final training MSE loss: 0.001898
  - Final validation MSE loss: 0.001919

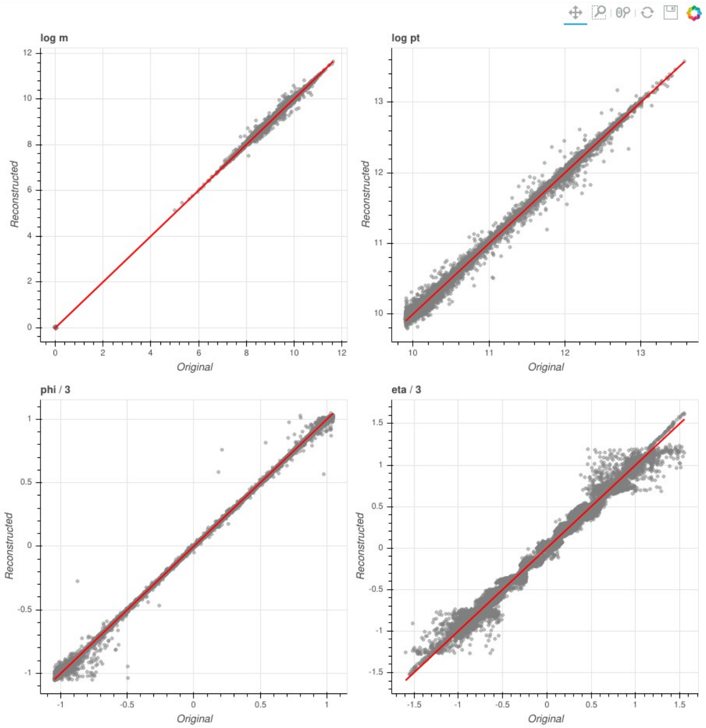




Reconstructed vs original

- The red lines represents a perfect reconstruction
- Most noticeable reconstruction failures are at the extremal values, see `phi` or `eta`





Improvements

- Try out different architectures using Neural Architecture Search, hyperparameter search
- Transfer learning – use the network from previous project
- Using our current knowledge of being unable to properly reconstruct variable `eta`, we could:

  Used weighted random sampling to resample points with high reconstruction error

  Switch Mean Squared Error loss function to a weighted equivalent (or use different loss)

  – assign higher weights to badly reconstructed variables

  Instead of compression 4 → 3, try compressing 3 → 2

  – could be an easier task, especially 3 → 2, excluding `eta`

Other ideas to investigate

- Try out different normalization schemes for the input variables, e.g. z-score
- Try combining Generative Adversarial Networks into the framework (such as energy-based EBGAN)
  - useful for modeling distribution – generate more samples
  - adversarial training for robustness
- Investigate different initialization schemes for the network
  - interesting because we can use true gradients (whole dataset fits into the memory), so good initialization is crucial