Anomaly Detection for Transient Surveys

Sigrid Nissen
Lukas Felix
Konstantinos Plainos

Astronomical transient surveys

- What are transients?
  - Objects in the sky whose brightness varies with time
- What types exist?
  - In the Milky Way
    - Variable stars, novae
  - In other galaxies (our focus)
    - Super-/kilonovae, tidal disruption events, active galactic nuclei, etc.
- How do we discover them?
  - Survey telescopes monitor the sky every night looking for changes
- What do we learn from them?
  - Many things!
  - Cosmology, heavy element nucleosynthesis, etc.

The Data

Observed by ZTF (Zwicky Transient Facility), a 48-inch (1.2 m) survey telescope

Scanning entire northern sky every two days since 2018 (while it works and it's not cloudy), giving roughly 1.4 TB of data per night.

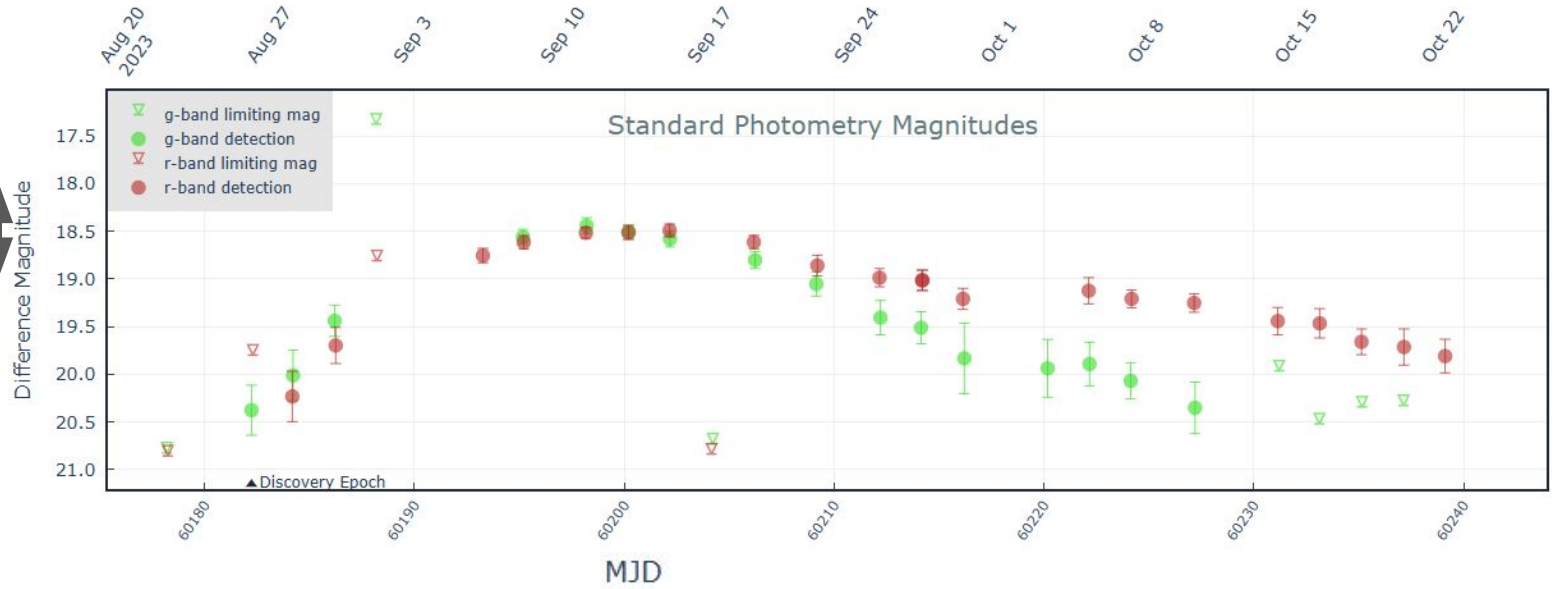
http://skyvision.caltech.edu/nightlysummary/

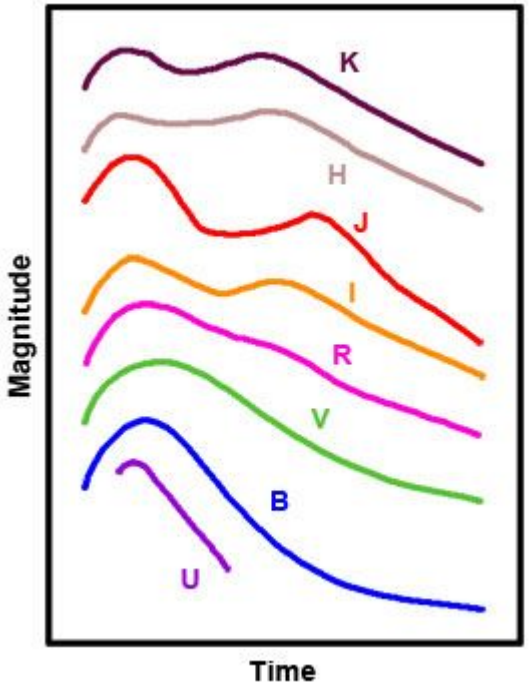
The Data

Time-series observations in 2 bands (red and green)

Apparent Magnitudes of the observed object in two colors

(difference of 1 = 2.5x as much light, lower number is brighter)

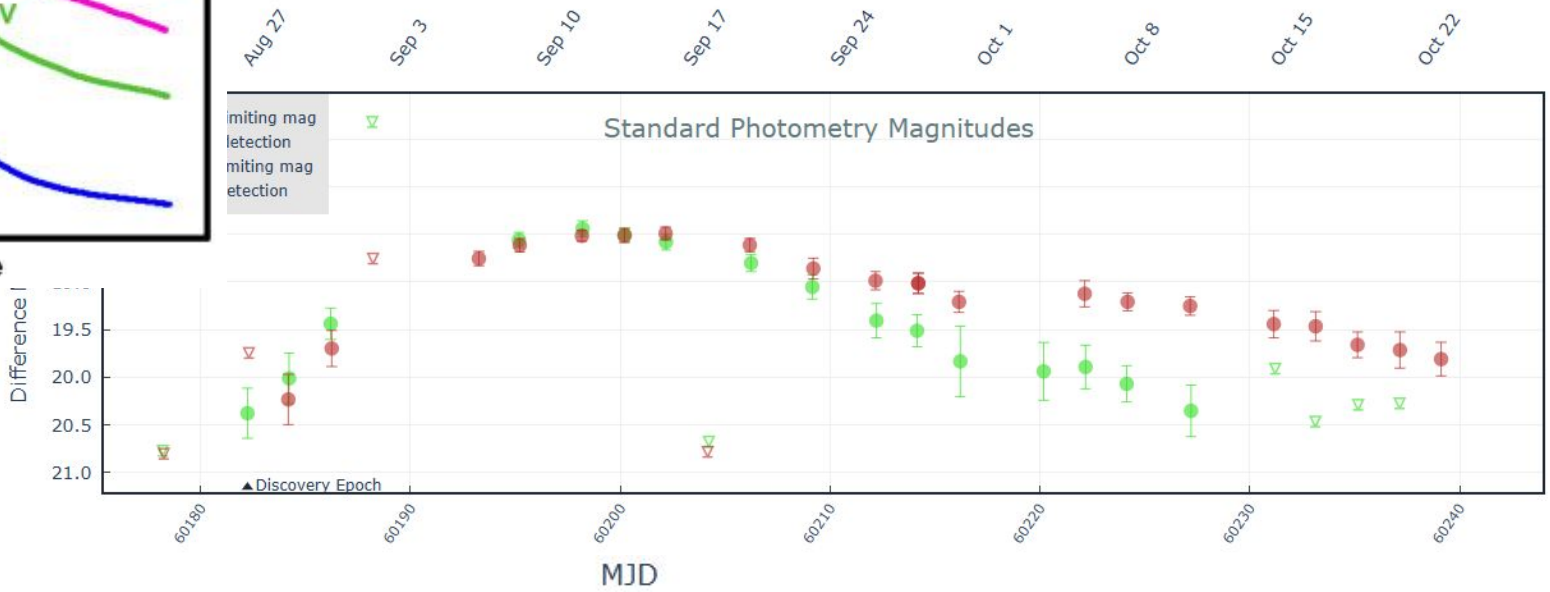




The given example here shows the **detection of a SN Ia.**

The ZTF has seen over 10’000 supernovae, ~70% of today’s known.

https://astronomy.swin.edu.au/cosmos/T/Type+Ia+supernova+light+curves



The Problem

Supernovae are cool, but we want more interesting objects, more special SN or TDEs etc.

In fact, we want the most interesting, most weird objects.

Anything that doesn’t fit in with the rest could be new astrophysics for us to explore!

Data Preprocessing

-Remove all objects with <10 photometric detections

-Remove useless metadata columns

-Constrain max time between max and minimum magnitude

Split data into table with extracted features and light curve data, to give us:

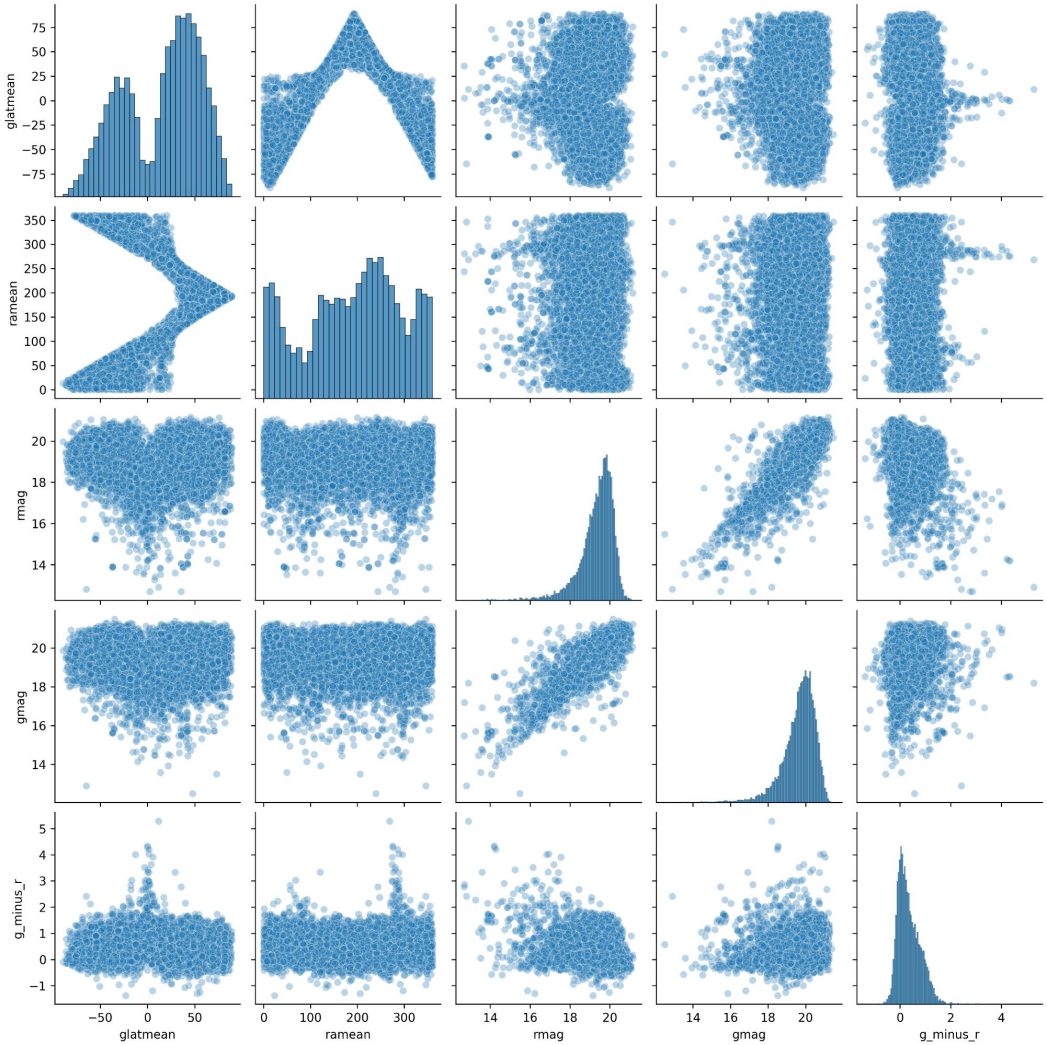
16338 objects with extracted tabulated features (up to 50 features each)
    -> ~5200 objects (~32%) have “truth” labels
10748 objects with light curves (10+ data points, ~20-30 is a median light curve)

Tabulated Features

50 features total

Some correlations clearly visible, e.g. G mag and R mag.

We typically remove 10-30 features that we deem unimportant for all training runs, but the specifics vary by person.

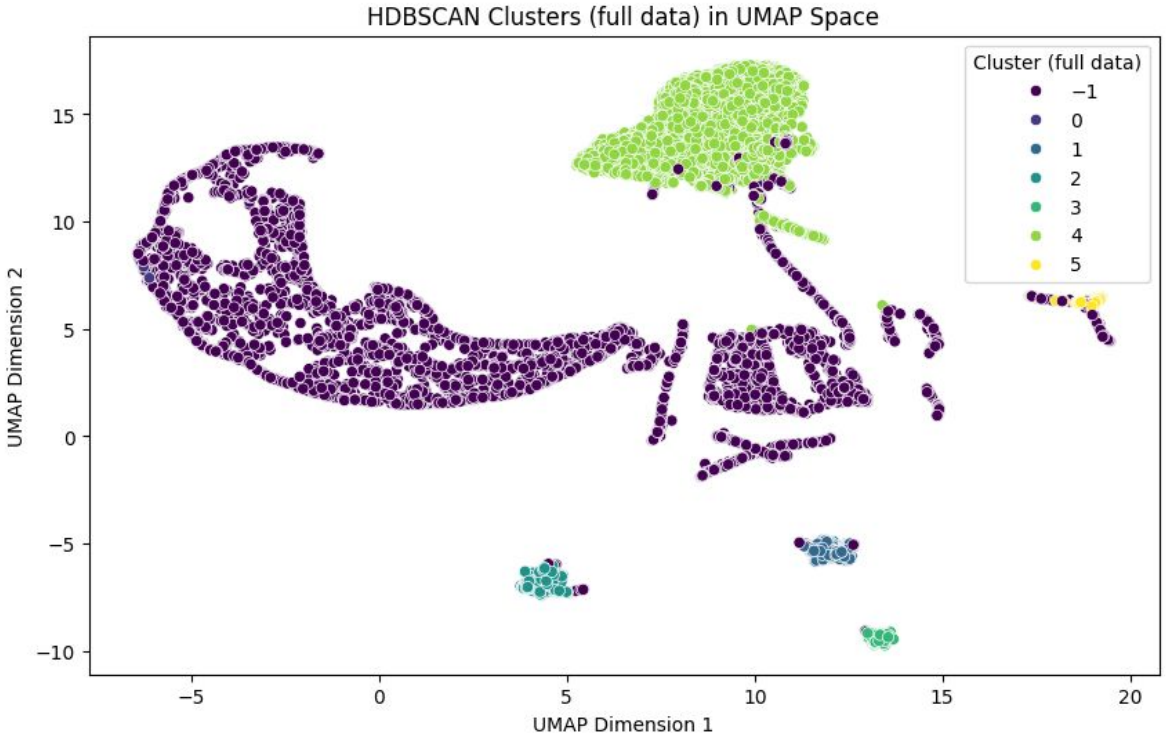


Attempts on extracted tabulated features: Clustering

Very simple idea

Cluster data using HDBSCAN
(min_cluster_size=50, min_samples=40)

Use noise category and outlier score for anomaly ranking



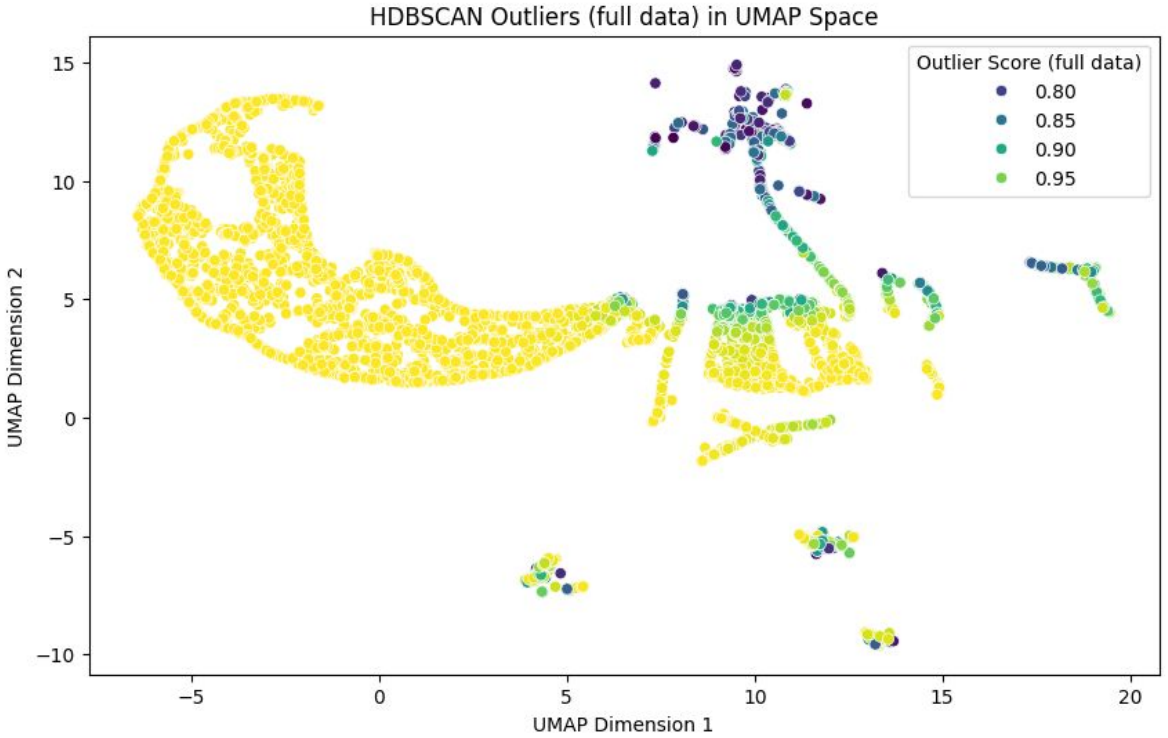
Attempts on extracted tabulated features: Clustering

Very simple idea

Cluster data using HDBSCAN
(min_cluster_size=50, min_samples=40)

Use noise category and outlier score for anomaly ranking

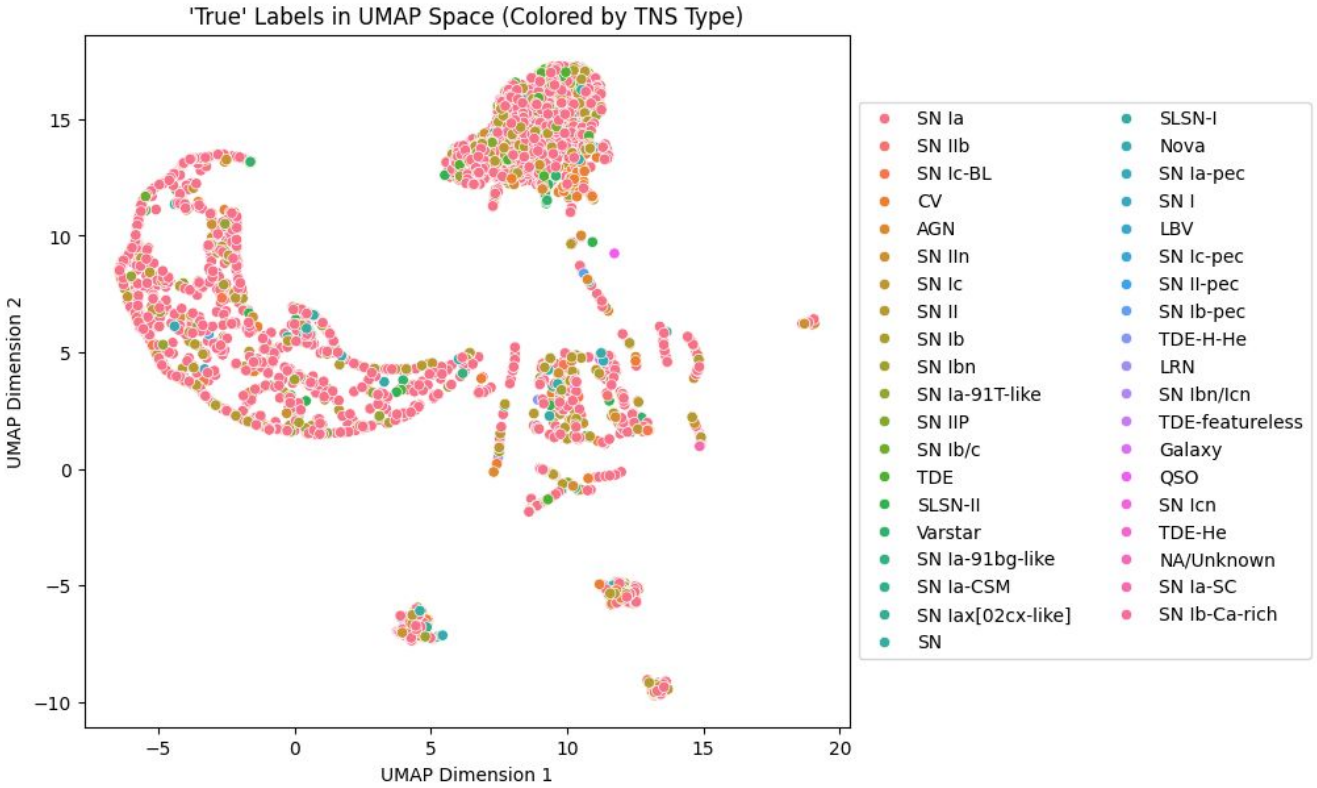
~30% of anomalies already have labels :(



Attempts on extracted tabulated features: Clustering

Why it didn't really work:

The clustering was not representative of the actual labels



Attempts on extracted features: Isolation Forest

Anomalies easier to separate

Processing: Removed AGN/Variable Star context classification

Extra features of note:

- LC duration (in days)
- “Missingness” indicators

Two passes

- With and without “missingness” features

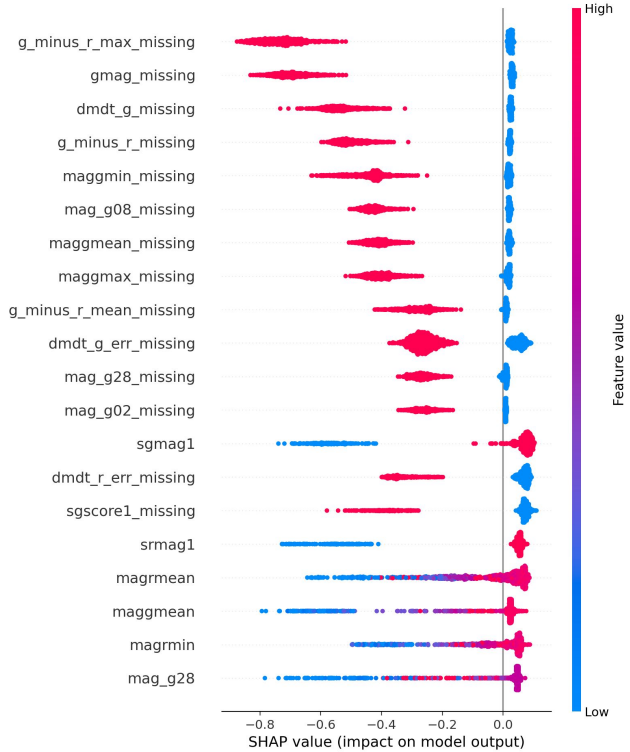
SHAP feature importance of anomalies

First pass

Selection dominated by “missingness” features

- Anomaly if no g band

Not what we are looking

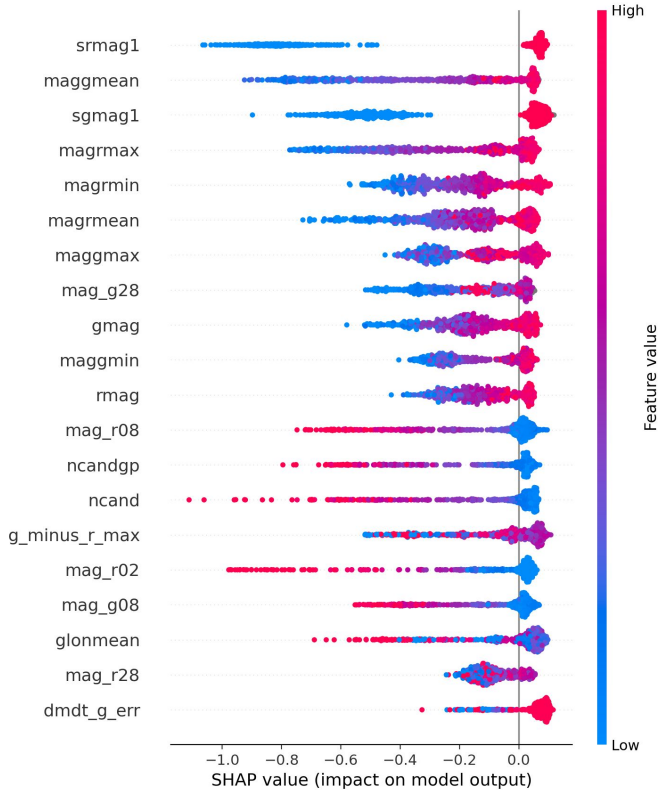


SHAP feature importance of anomalies

Second pass

- Picks out unusually bright objects/objects associated with bright galaxies
- More interesting!

Only 8% overlap between top 100 of passes
→ different anomalies found

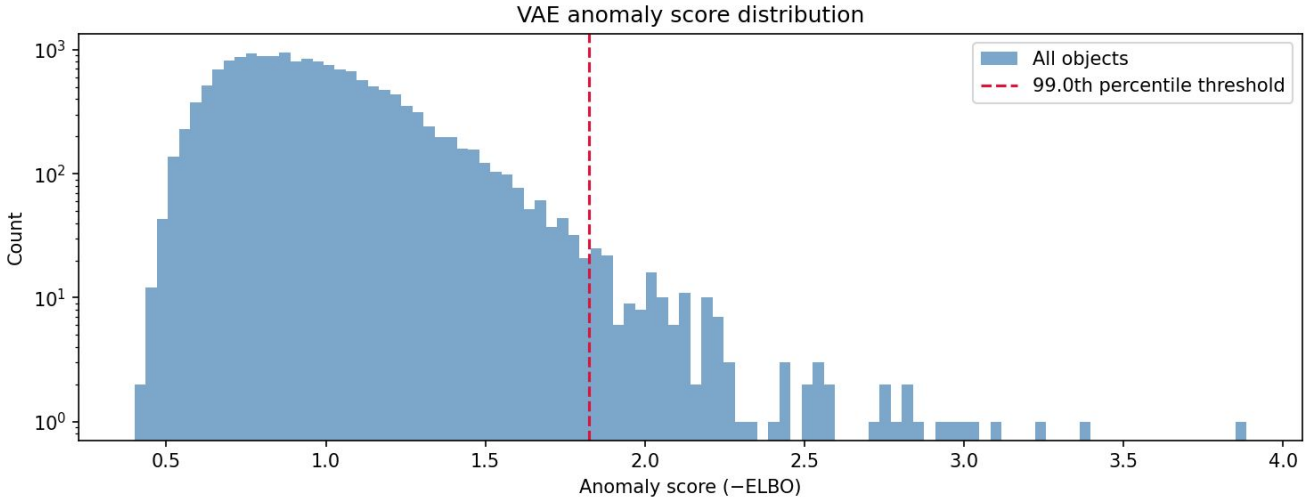


Attempts on extracted tabulated features: Auto-Encoder

Idea: try neural net approach based on a Variational AutoEncoder (VAE)

Train using reconstruction error + KL divergence

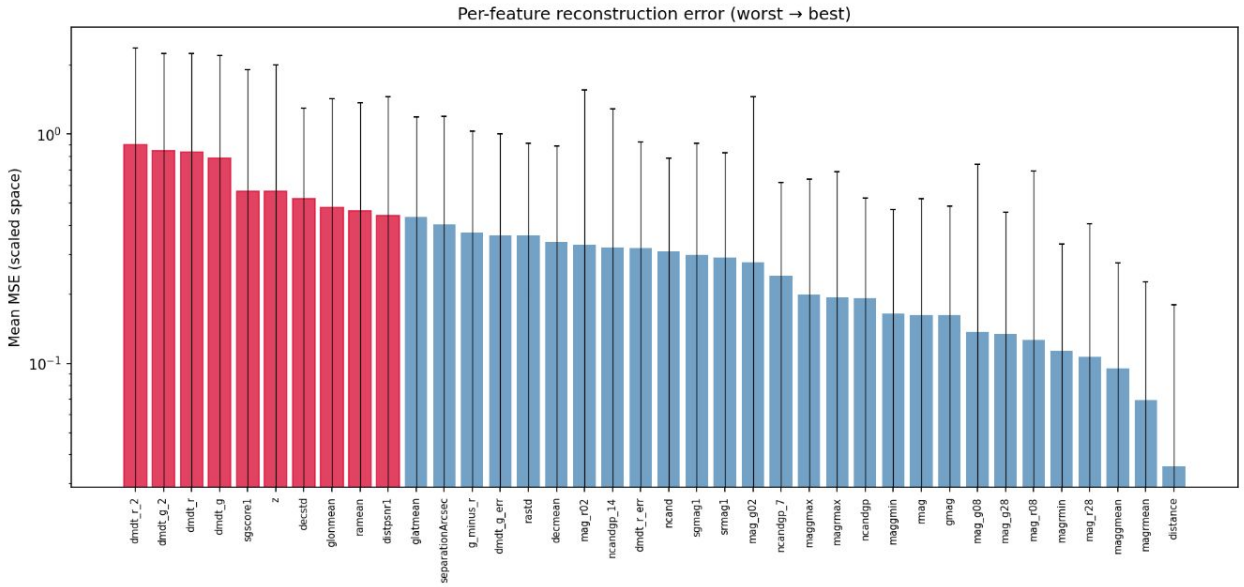
-> difficult to reconstruct objects (99th percentile) are flagged as anomalies



Attempts on extracted tabulated features: Auto-Encoder

Most difficult to reconstruct features were actually the most informative about lightcurve behaviour, their slopes!

But the highly ranked anomalies were actually anomalous light curves!



Attempt on the original light curves

Idea: use the actual light curves in an autoencoder, difficult to reconstruct objects are then classified as anomalies

Difficulty: how to handle time?

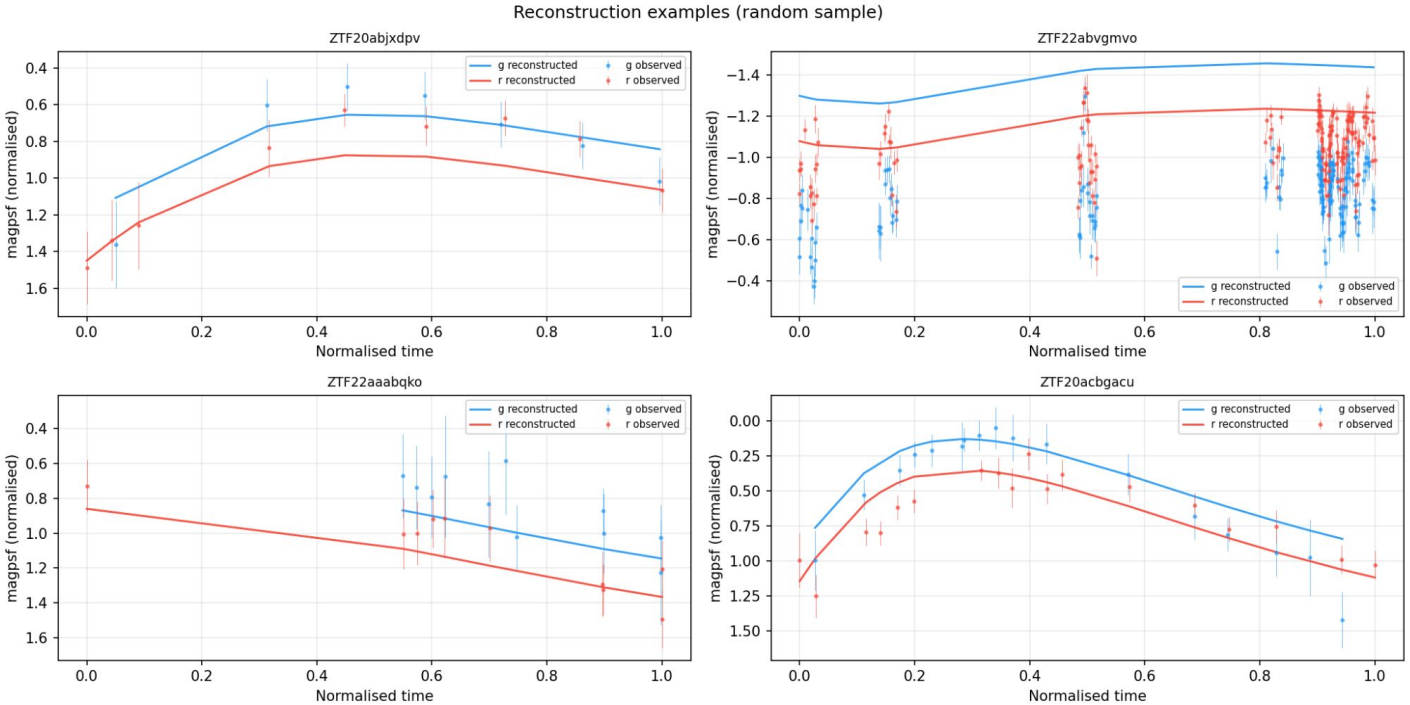
Attempt here: Normalize time for each observation within [0, 1].
Keep irregular time series, so no binning/interpolating.

Model choice: mTAN (Multi-Time Attention Network) Autoencoder

-> allows mapping of our irregular time series into a common latent space

Attempt on the original light curves: Reconstruction

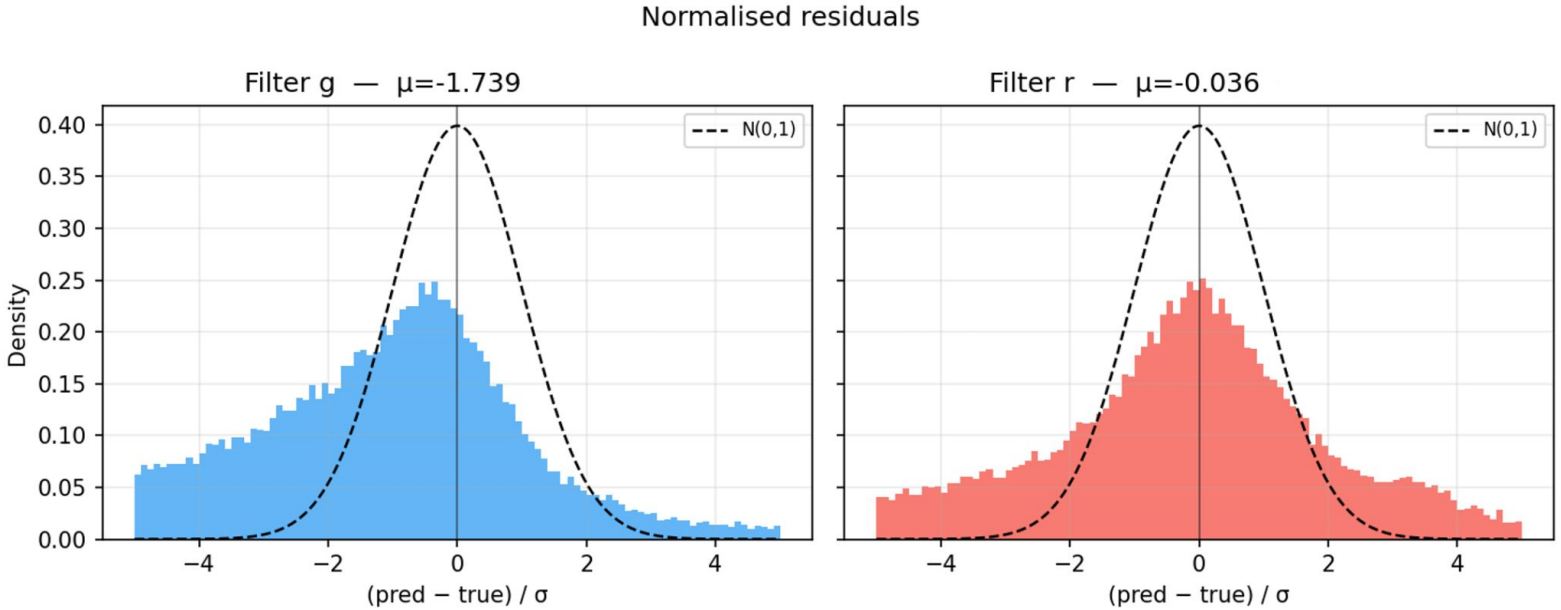
Reconstruction works decent-ish, although struggles for high cadence targets.



Attempt on the original light curves: Reconstruction Resid.

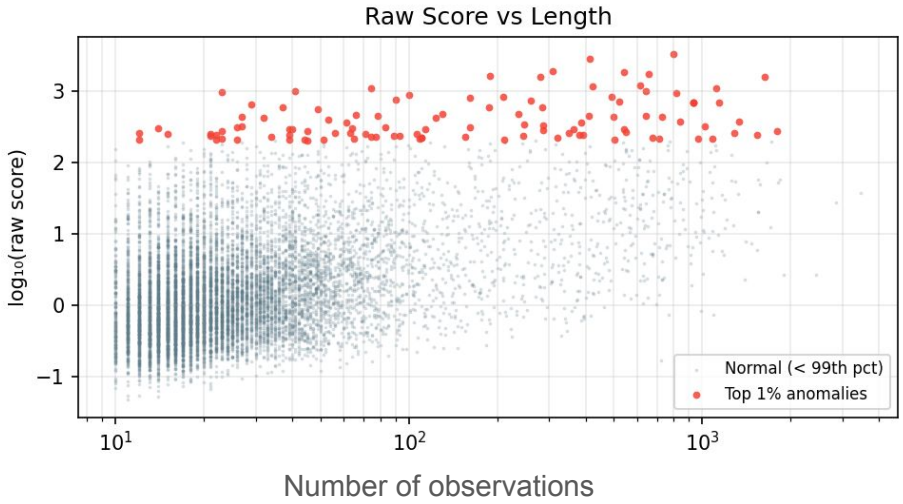
In the residuals, we see that we don’t reproduce fits that are on par with the data quality.

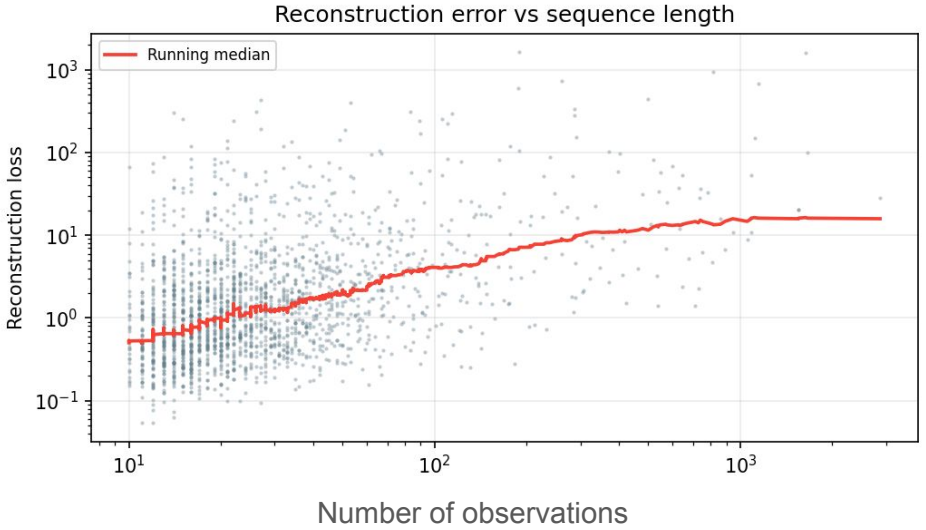
Our residuals show wide wings, deviating from a true gaussian distribution.



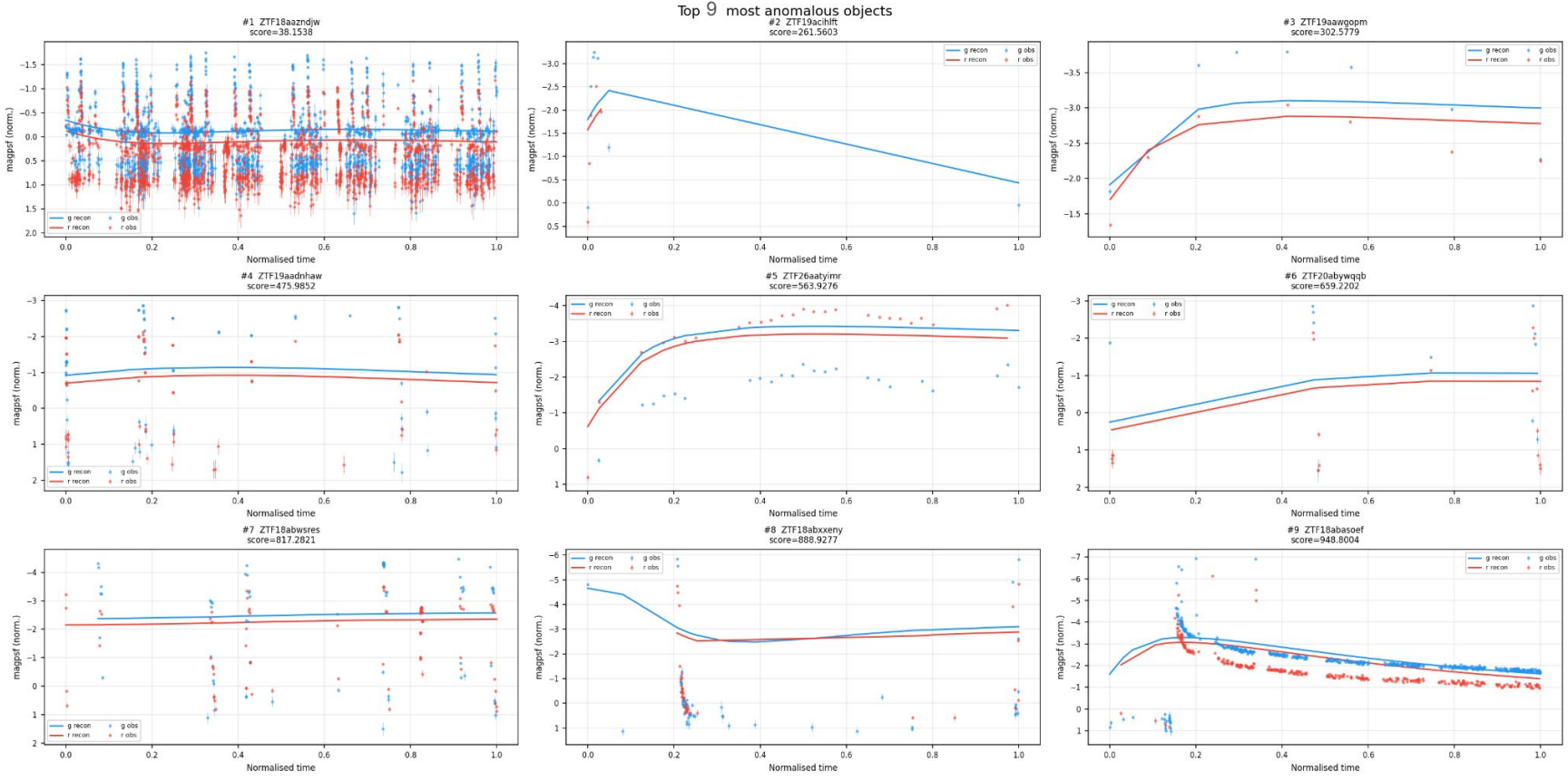
Attempt on the original light curves: Anomalies

The reconstruction is biased towards shorter time series, leading to more anomalies for objects with more data.

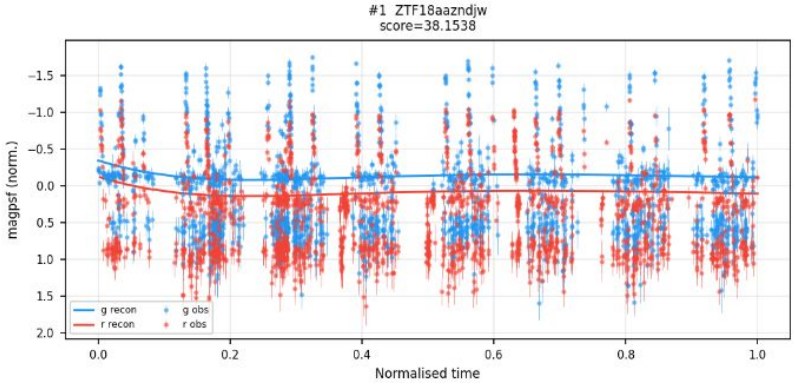


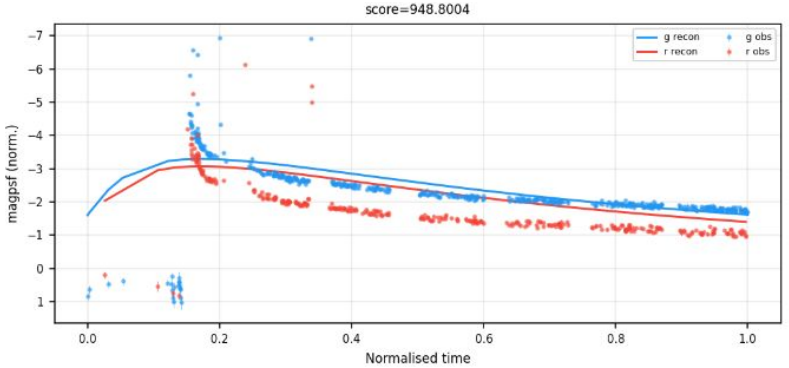


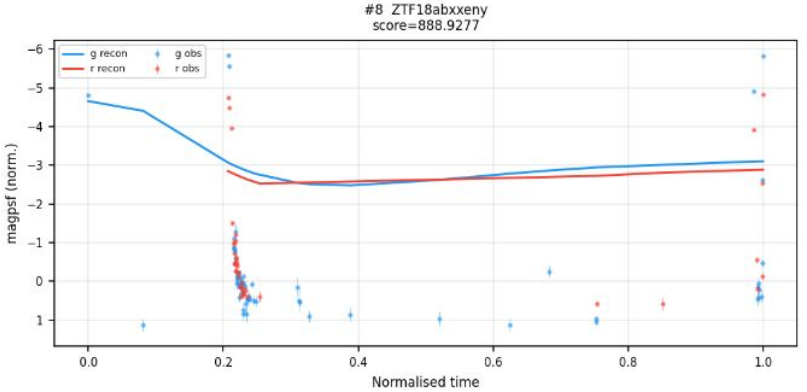
Finally, some anomalies



Finally, some anomalies







Conclusions

None of the methods worked all that great (yet)

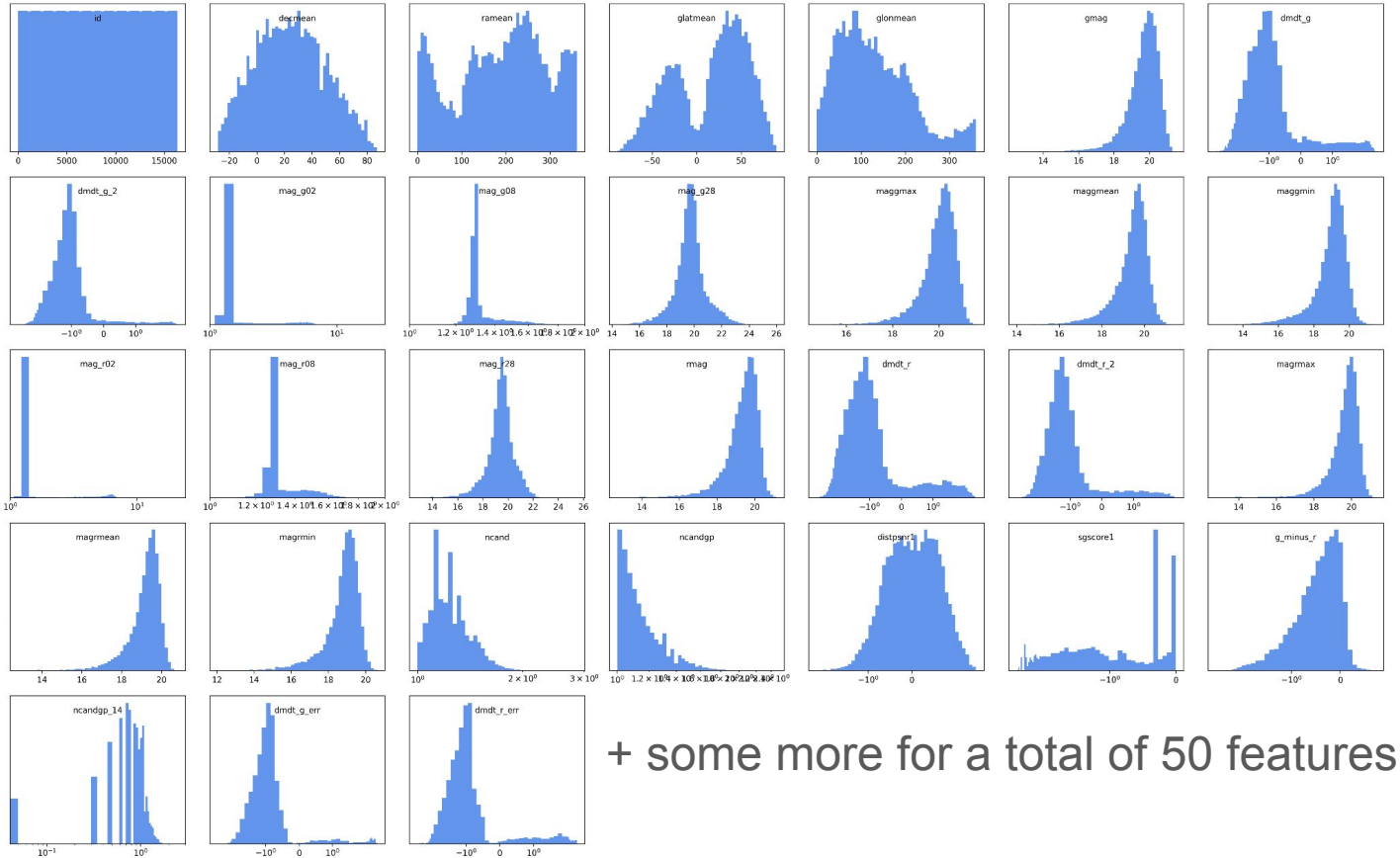
The most true representation of what is actually an anomaly comes from the overlap of different methods!

This will get way more important with LSST (Legacy Survey of Space and Time) at the Vera Rubin observatory starting up, because the numbers will get out of hand! (10 million objects observed per day, 15 TB of data per night)

Appendix

Data

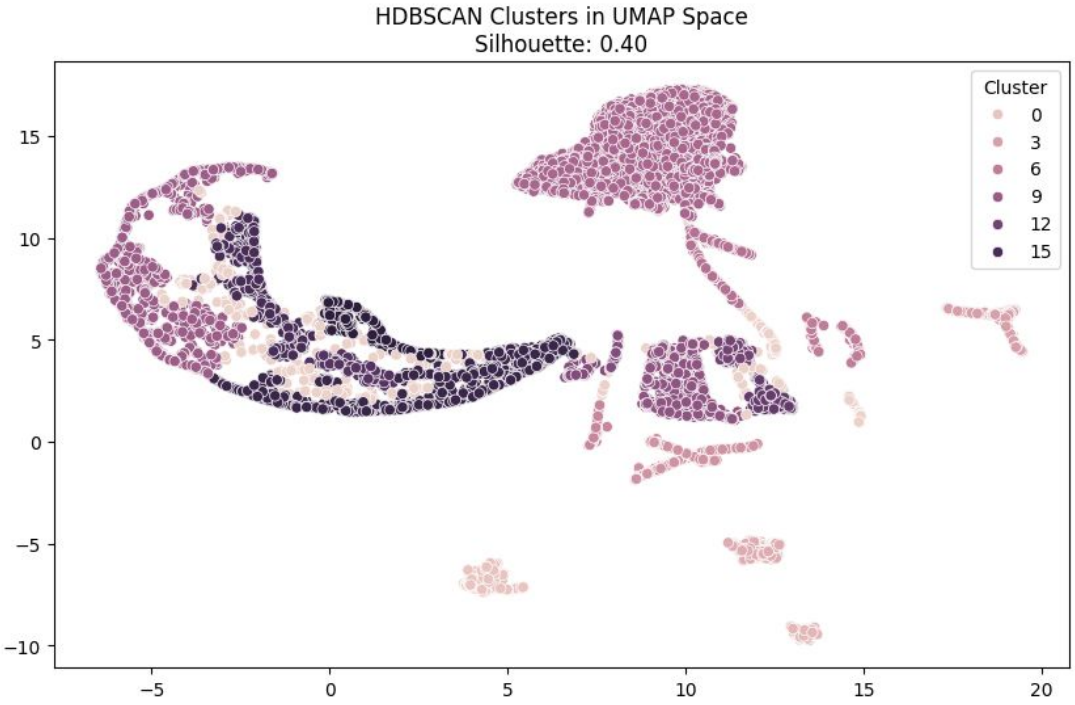
Tabulated Features, showing more available features



+ some more for a total of 50 features

Clustering

Also tried clustering in UMAP space directly, which gave a better Silhouette score



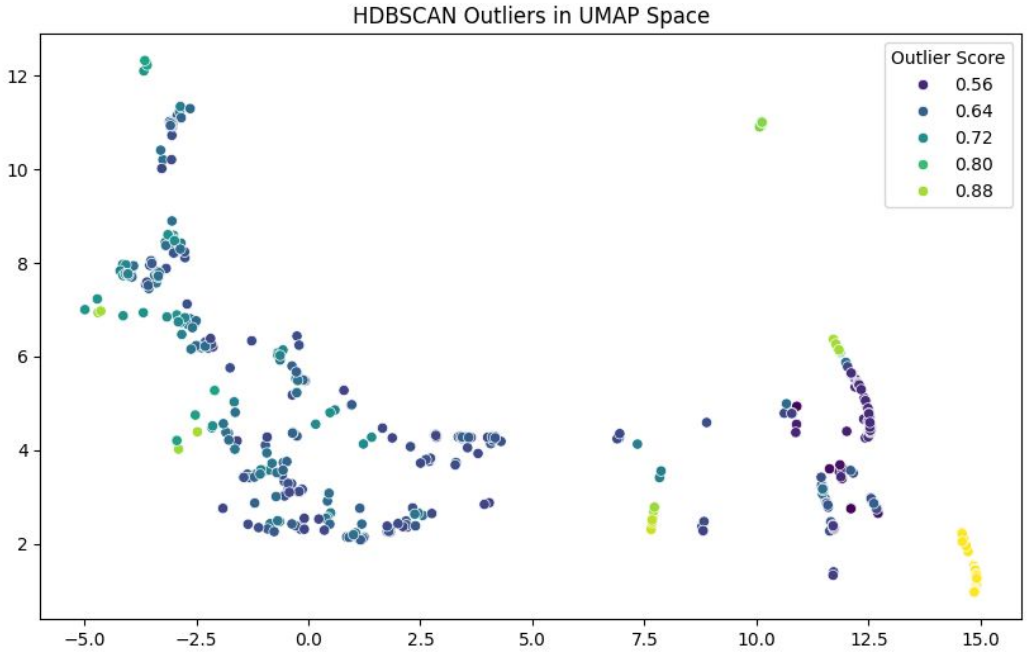
Clustering

Also tried clustering in UMAP space directly, which gave a better Silhouette score

And the anomalies were now largely constrained to the points that were previously labeled as noise by HDBSCAN

Also, only ~50 out of 410 detected anomalies are already labeled (12%), much better than in our presented attempt!

But hard to justify why this worked better, it shouldn't.



Clustering

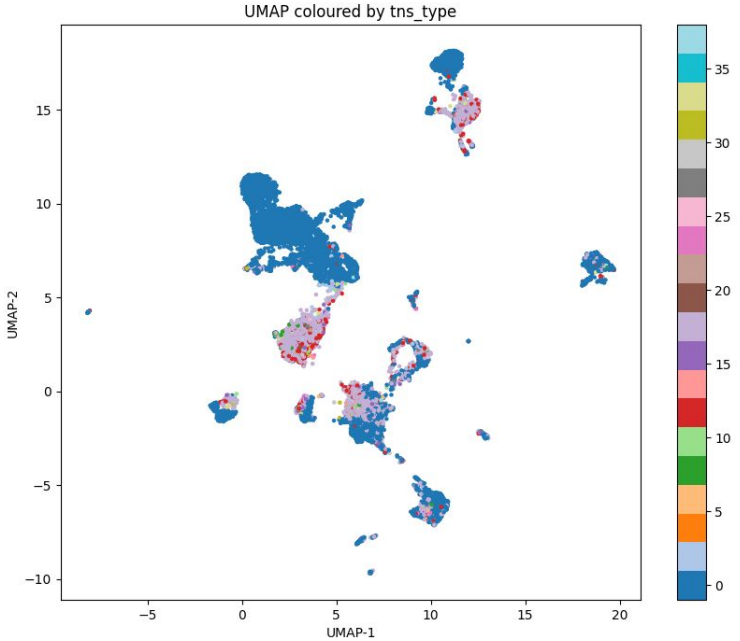
Another attempt with the same extra features as in the isolation forest → aiming for atypical supernovae specifically

HDBSCAN

- StandardScaler + “missingness”
- min_cluster_size=50,
- cluster_selection_epsilon=5.0

UMAP visualization

- No clear separation within clusters



Isolation forest

Default sklearn iForest settings

- Tried varying contamination levels, but decided to just choose top 100 highest anomaly scores

Anomalies are generally: bright, “red” (brighter in r than g), far from any associated galaxy

These are all interesting scientifically

Majority (68%) unclassified on TNS - slightly more than full sample (62%)

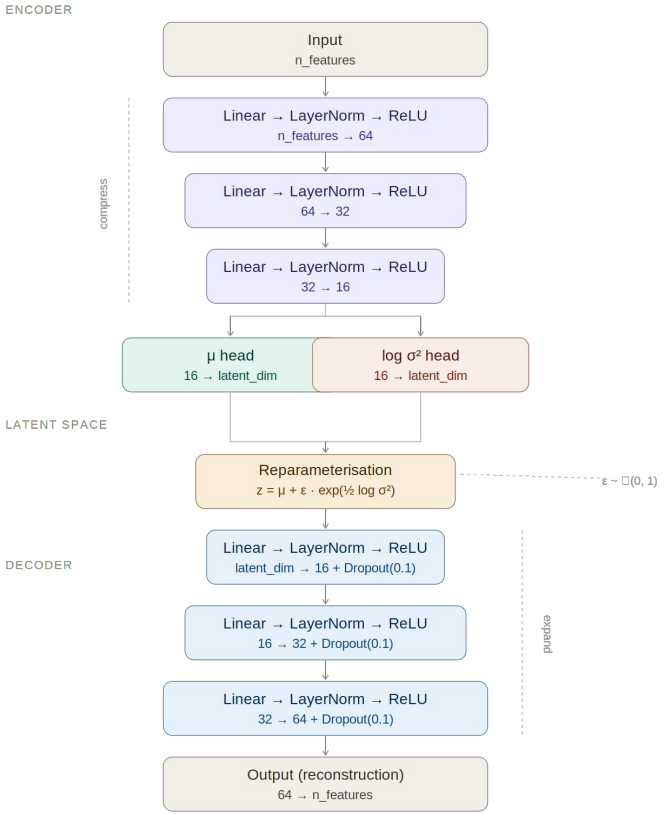
- Good

Variational Autoencoder from presentation

n_features = 36
latent_dim = 8

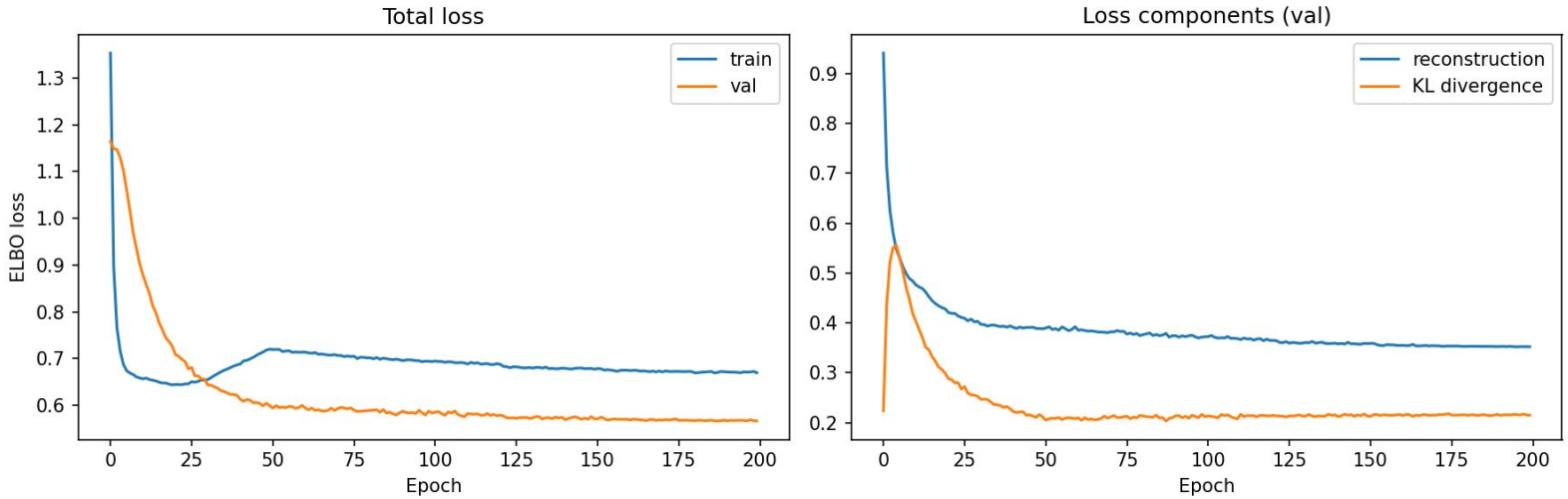
Training

Hyperparameter	Value	
epochs	100	
batch_size	256	
lr	1e-3	
beta	1.0	loss = MSE_mean + beta x KL_normalised
warmup_epochs	50	Linear beta warmup
weight_decay	1e-4	Adam regularisation
grad_clip	5.0	



Variational Autoencoder Training

A warmup (linear increase for the first 50 epochs from 0 to beta, typically 1) for the KL divergence component component of the loss was used, to allow the autoencoder to first train reconstructing the training data

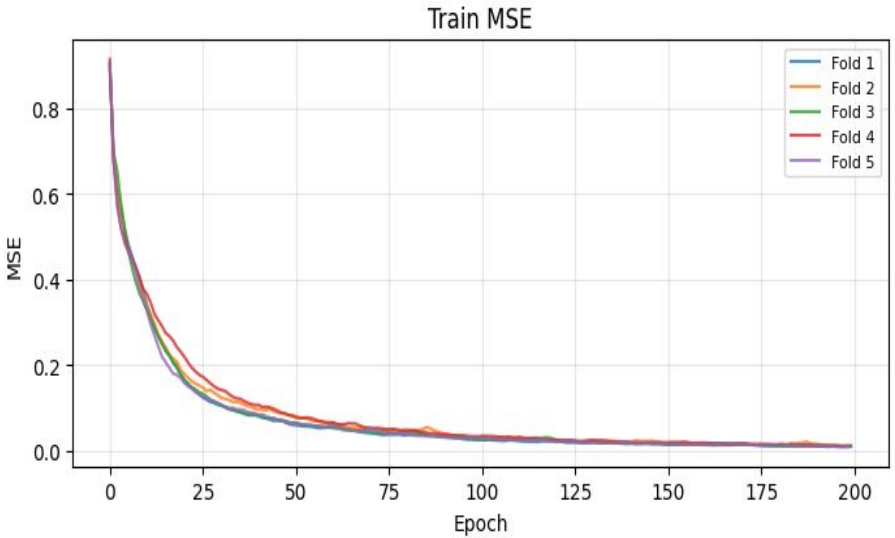


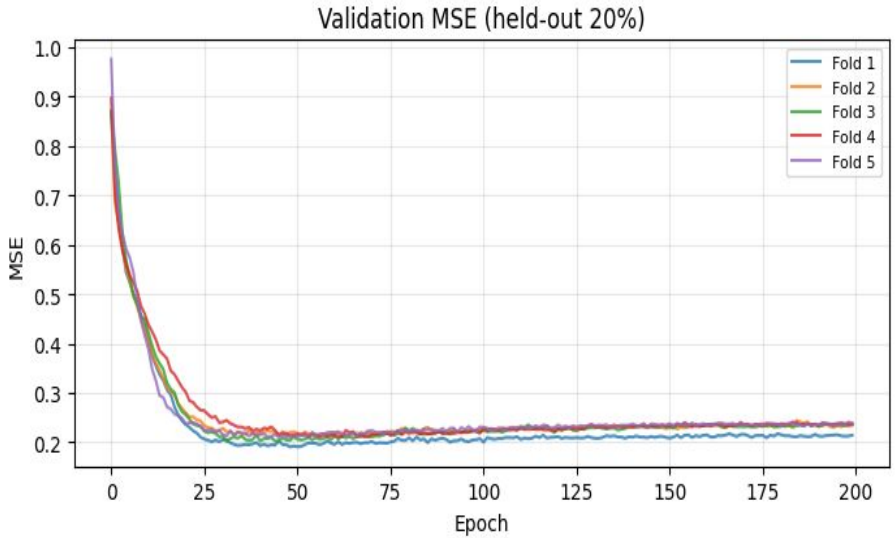
Autoencoder

- Take 32 photometric features per object
- Compress down to 8 numbers (latent space) → then reconstruct back to 32
- Symmetric Architecture: 32 → 1024 → 512 → 256 → 128 → **8** → 128 → 256 → 512 → 1024 → 32
- Trained only on normal objects (SN, AGN, variable stars)
- Score = average squared difference between input and reconstruction
- 5-folds so every object is scored by a model that never trained on it (80% train, 20% validation)

$$\text{anomaly score} = \frac{1}{d}\sum(x_j - \hat{x}_j)^2$$

Autoencoder

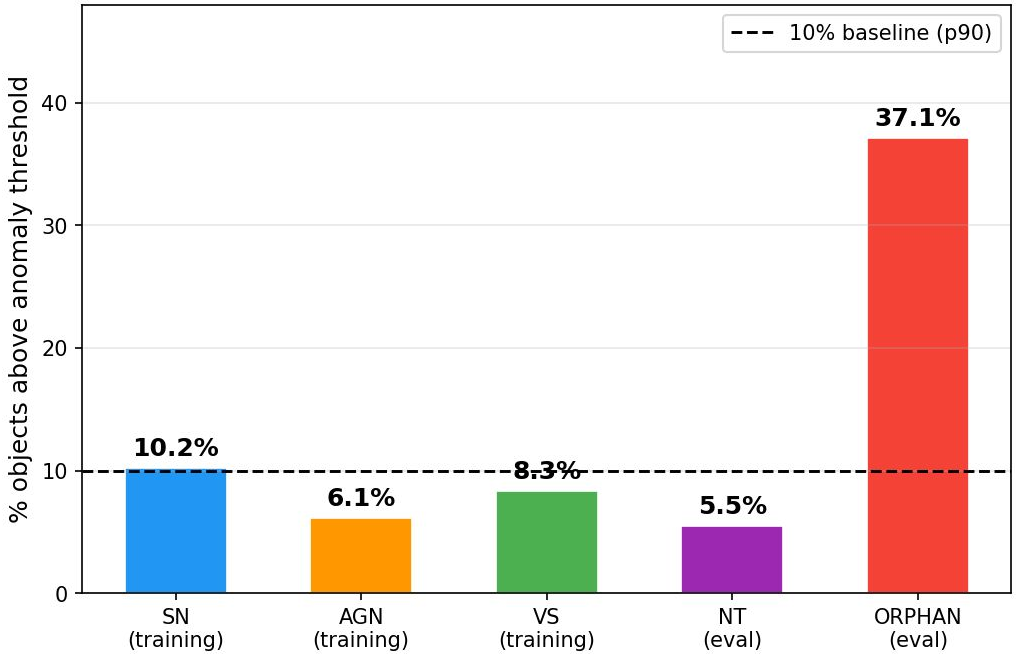




- All 5 folds converge consistently - the model isn't sensitive to which objects end up in which fold
- Validation MSE stabilises around epoch 50 and stays flat - no overfitting, training to 200 epochs is safe

Autoencoder - Some Results

Threshold = 90th percentile of all scores → by construction 10% of objects fall above it



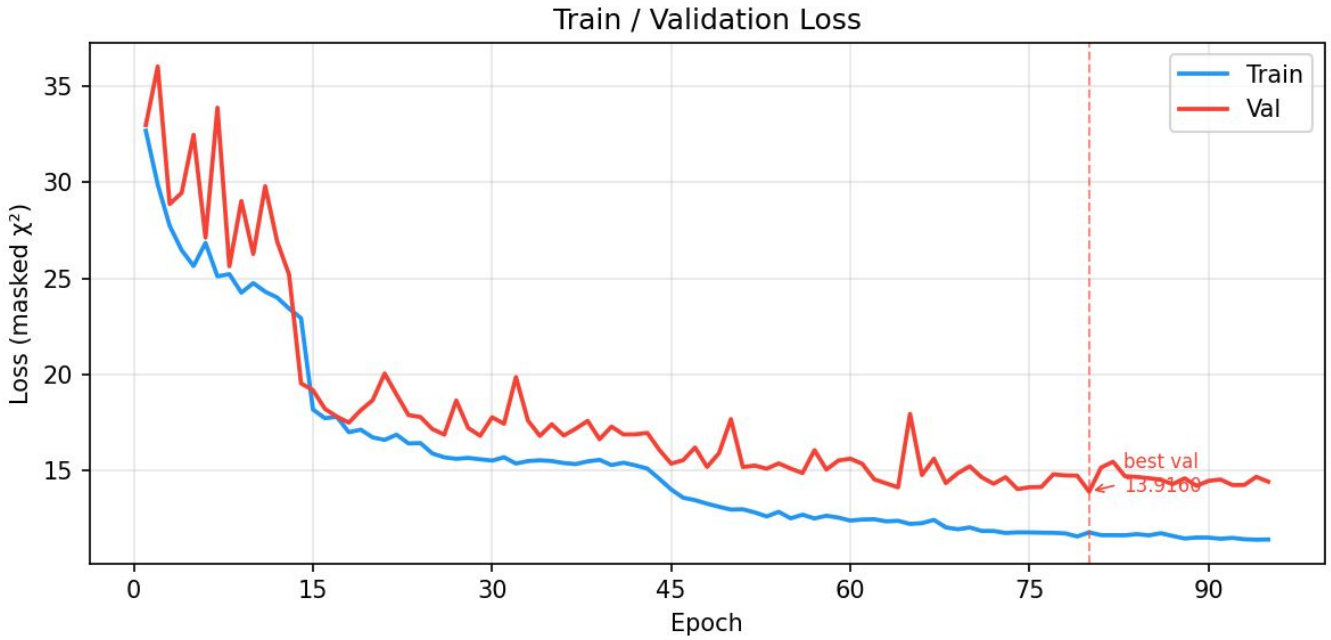
mTAN Autoencoder for time series

Stopped early as validation loss didn’t improve.

The loss function here is given as follows:

$$L = \alpha \cdot MSE + (1-\alpha) \cdot \chi^2 + \beta \cdot L\_deriv$$

Where



mTAN Autoencoder setup

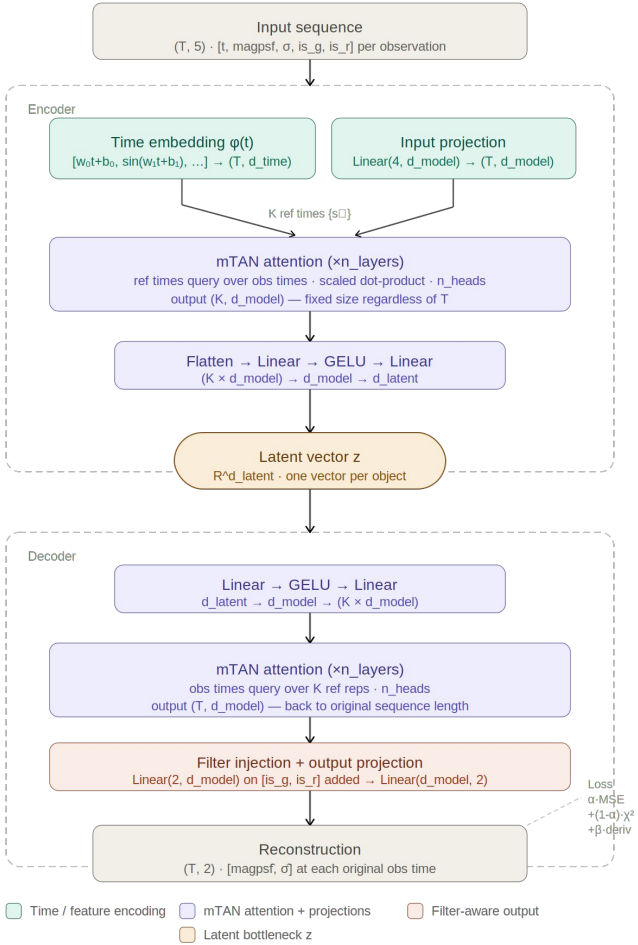


*figure generated by claude