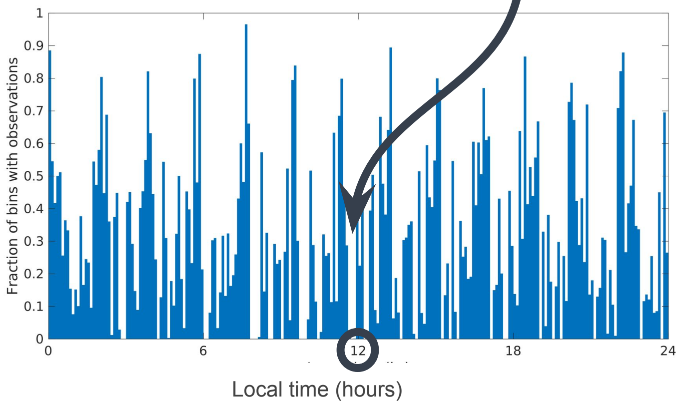
Interpolating lunar surface temperature profiles

TABLE OF CONTENTS

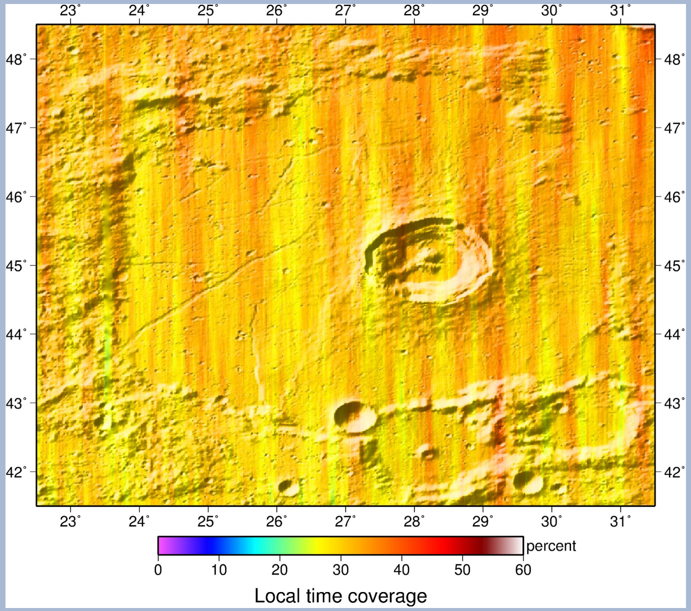
01 PROBLEM

“striping”

Surface temperature data coverage is not uniform **spatially** or in **local time**



Often missing observations of peak temperature (occurs at 1200 local time)



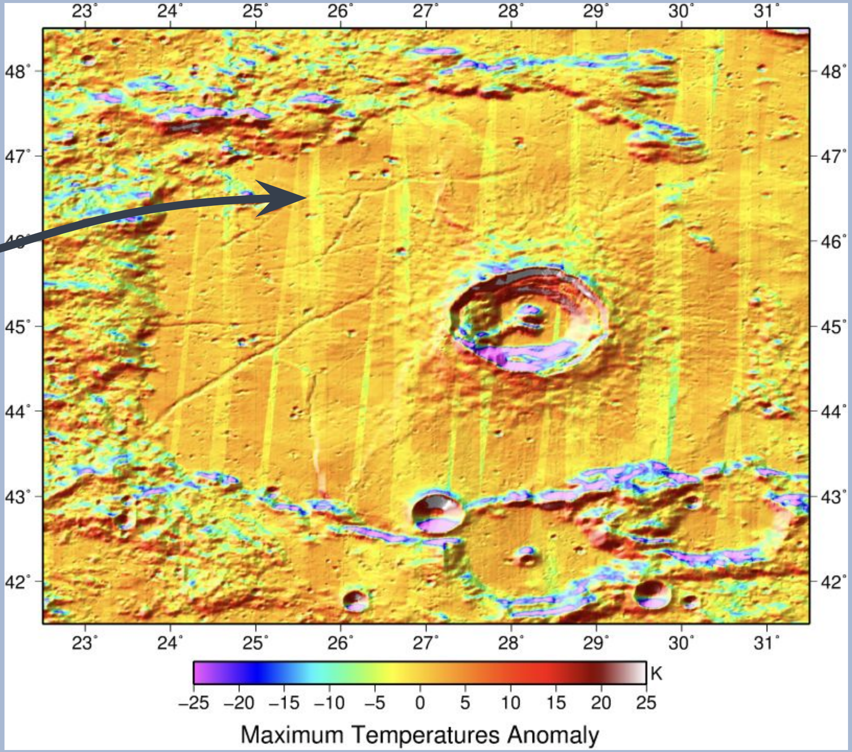
SOURCE: Jean-Pierre Williams

observed T_{max} < true T_{max}

Know this intuitively, based on comparison with surrounding T_{max}

Striping affects data products depending on:

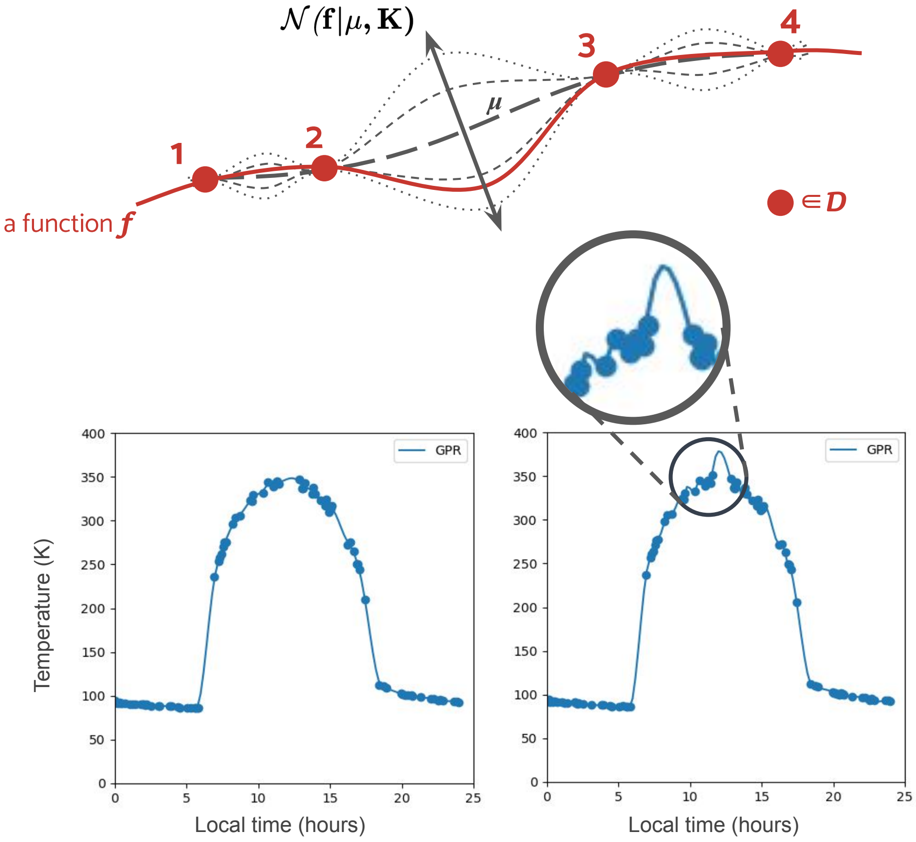
- T anomalies
- T range/variation
- T normalization



$$T_{bol}^{max} - T_{bol}^{zonal\ mean}\,(K)$$

T_{MAX} normalized by latitude

SOURCE: Jean-Pierre Williams



One-size-fits-all interpolation often overfits

Example: **Gaussian Process Regressor (GPR)**

- Nonparametric, Bayesian regression
- Assumes

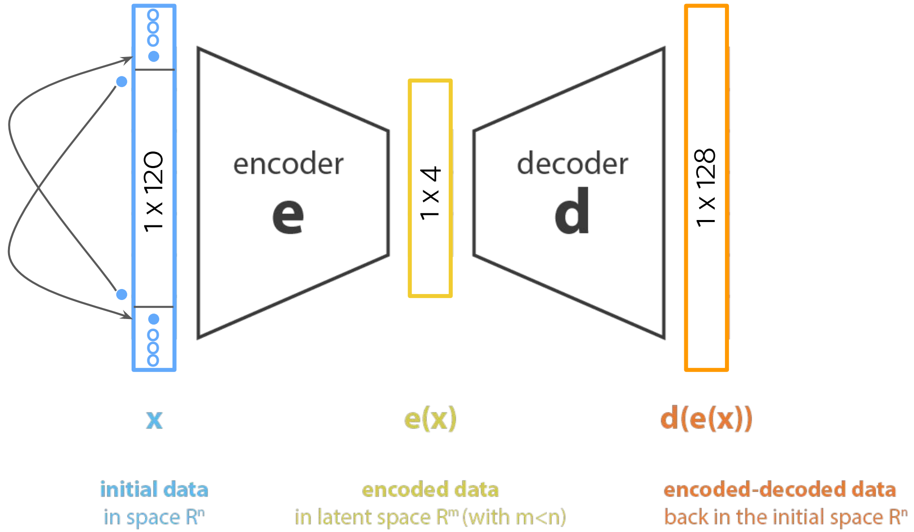
$$P(\mathbf{f}|D) = \mathcal{N}(\mathbf{f}|\mu, \mathbf{K})$$

i.e. probability of regression function, f, fitting data, D, is multivariate normal centered around mean function, μ, with shape/smoothness $\mathbf{K}$

Sensitive to data gaps

Larger gap = greater variance in distribution of fits = greater probability of a poor fit

Non-linear dimensionality reduction



$x = d(e(x))$ → **lossless encoding** no information is lost when reducing the number of dimensions

(ideal case)

$x \neq d(e(x))$ → **lossy encoding** some information is lost when reducing the number of dimensions and can't be recovered later

(realistic case)

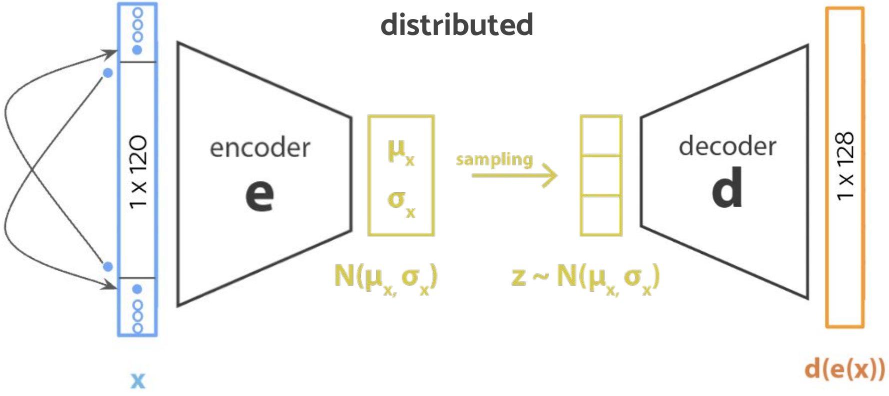
x = 1D array of interpolated temperatures at 12 minute intervals, padded to enforce periodicity + zero-padded to avoid edge effects of first convolutional filter (1 x 128)

e(x) = latent vector (1 x 4)

Variational autoencoder = probabilistic

Convolutional filters extract features from input, **x**
Encoded as a *probability distribution* into the latent space
A vector, **e(x)**, is sampled from the latent space distribution
Sampled vector is decoded, **d(e(x))**

VAE assumes that latent variables are independent & normally distributed



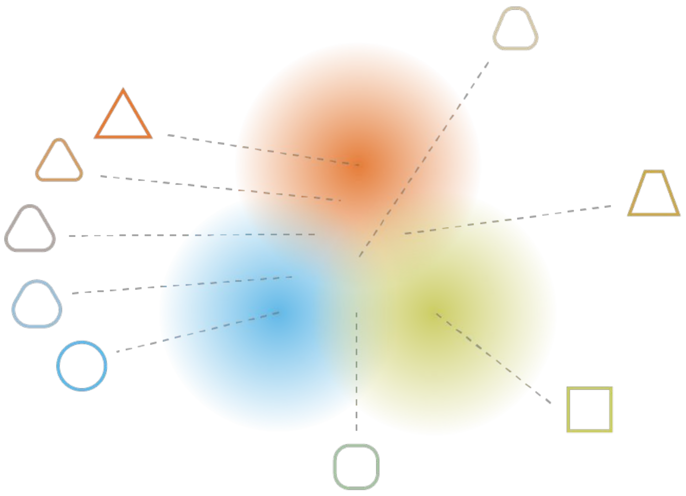
Loss functions penalize **x ≠ d(e(x))** and **"KL divergence"**

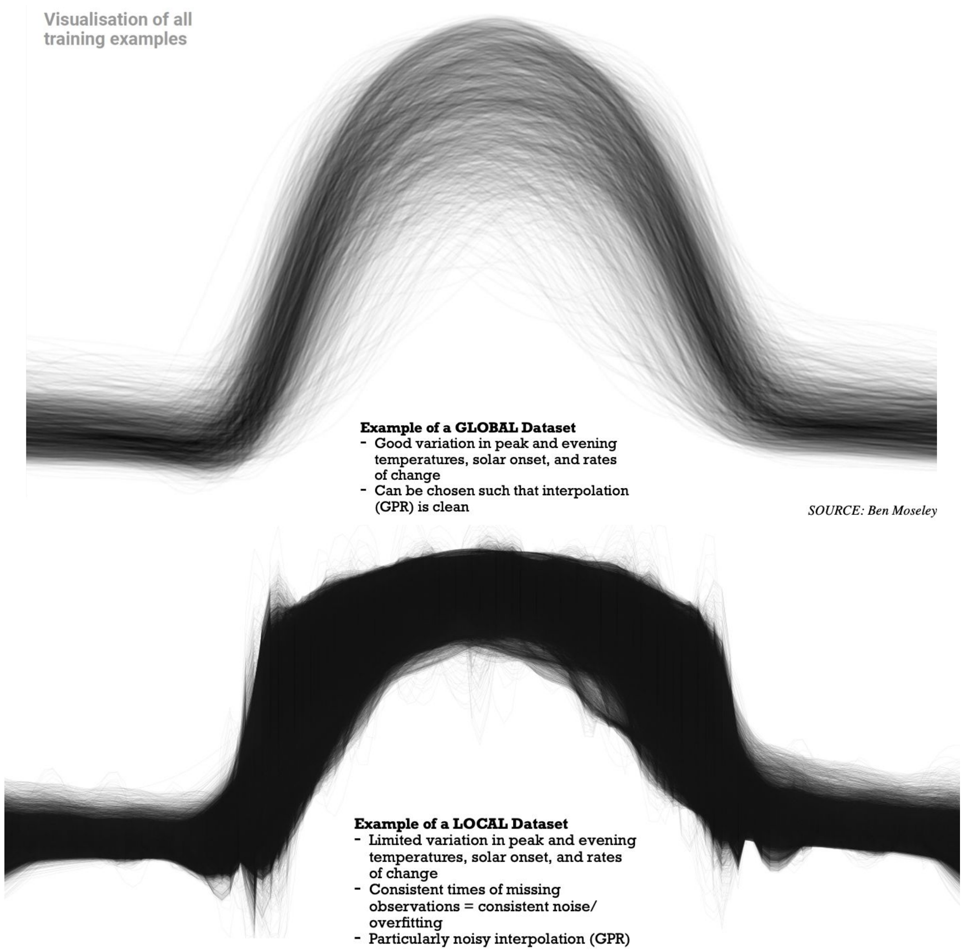
Think of this as something that introduces **continuity** to the latent space

i.e. two "similar" profiles should be near each other in the latent space

- Peak temperature,
- Solar onset time,
- Evening temperature,
- Rates of temperature increase/decrease, etc

Want to know:
how is the latent space organized?

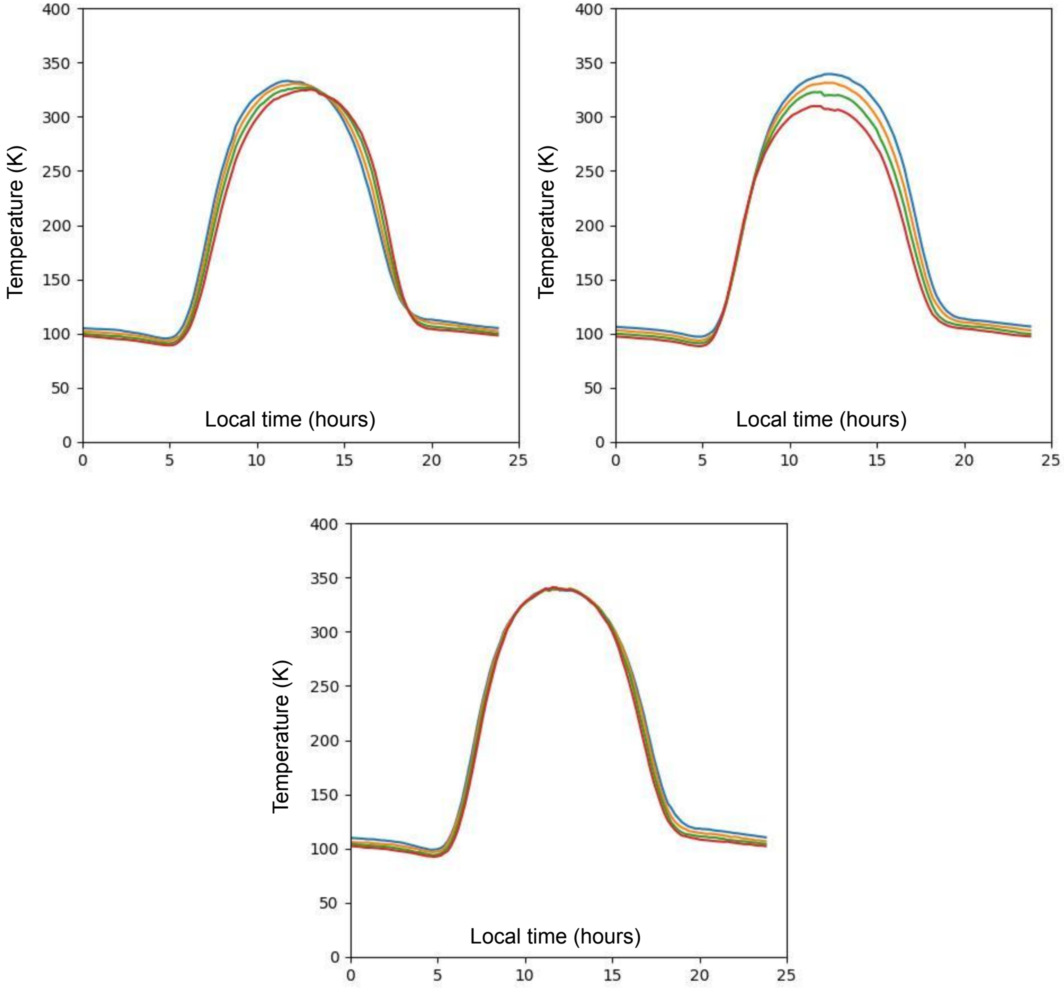




Need a carefully chosen input dataset that exhibits the thermophysics that we want to separate

Test 1: global dataset

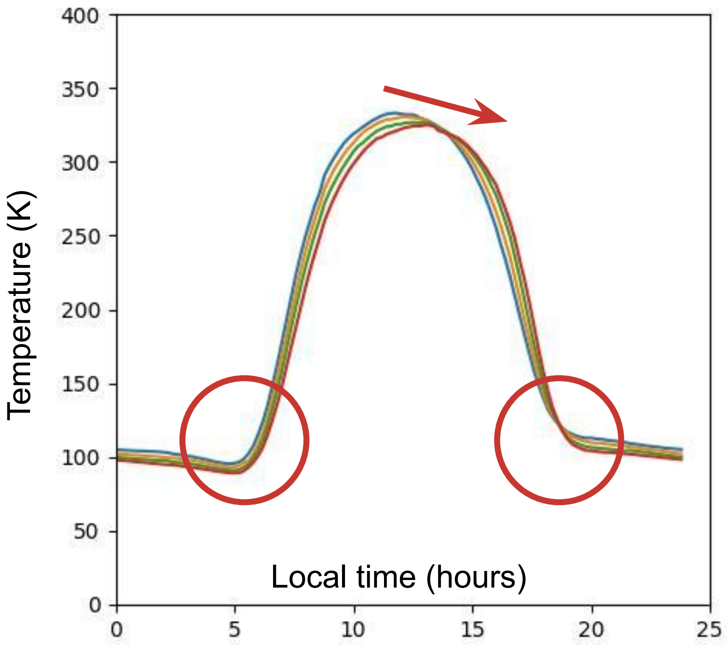
Test 2: “regional” dataset



Test 1: global dataset

- Decent separation of
  (1) solar onset delay
  (2) effective albedo
  (3) thermal retention

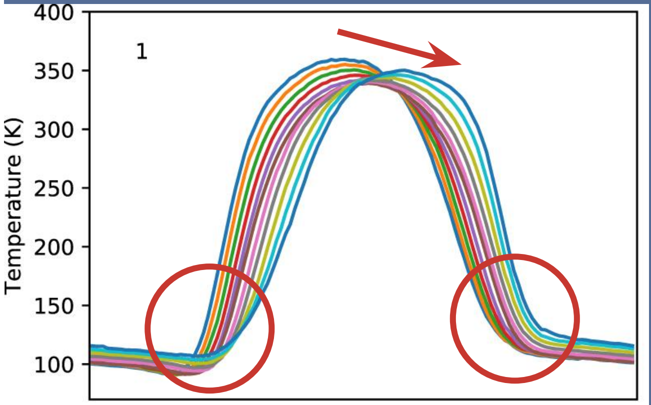
03 PRELIMINARY RESULTS

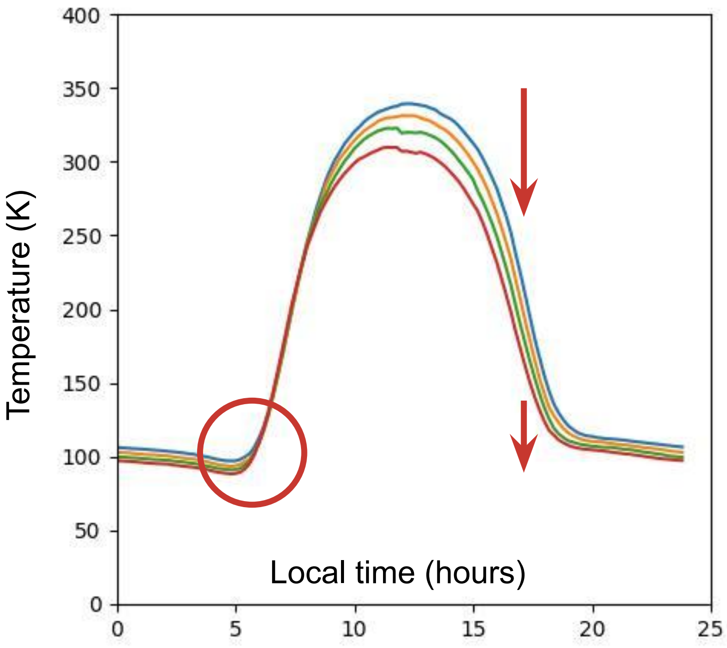


Test 1: global dataset

- Decent separation of
  (1) solar onset delay
  (2) effective albedo
  (3) thermal retention

c.f. Ben Moseley's result

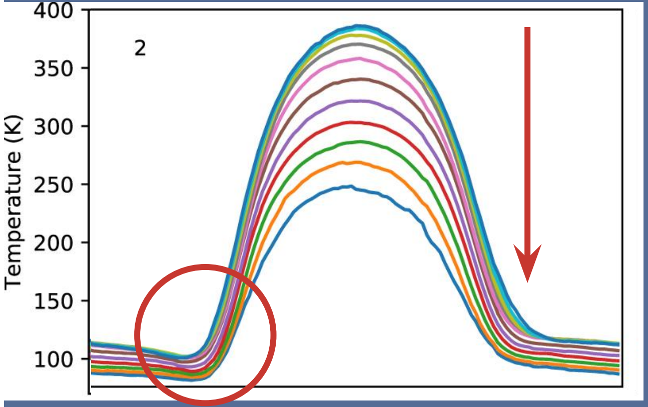


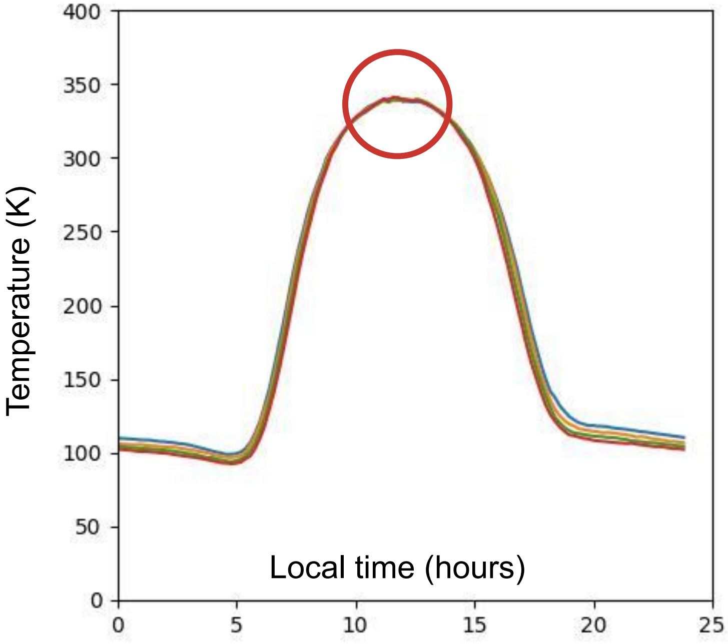


Test 1: global dataset

- Decent separation of
  (1) solar onset delay
  (2) effective albedo
  (3) thermal retention

c.f. Ben Moseley's result

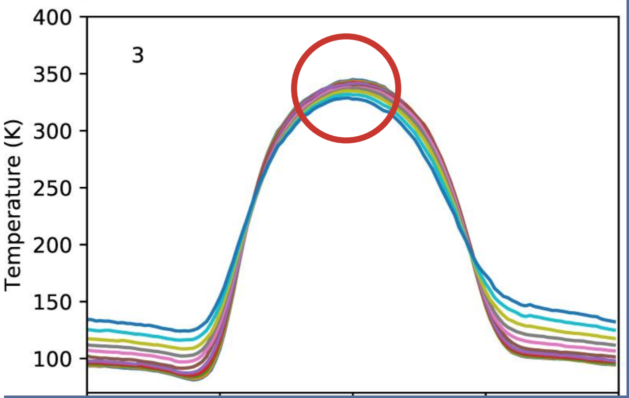




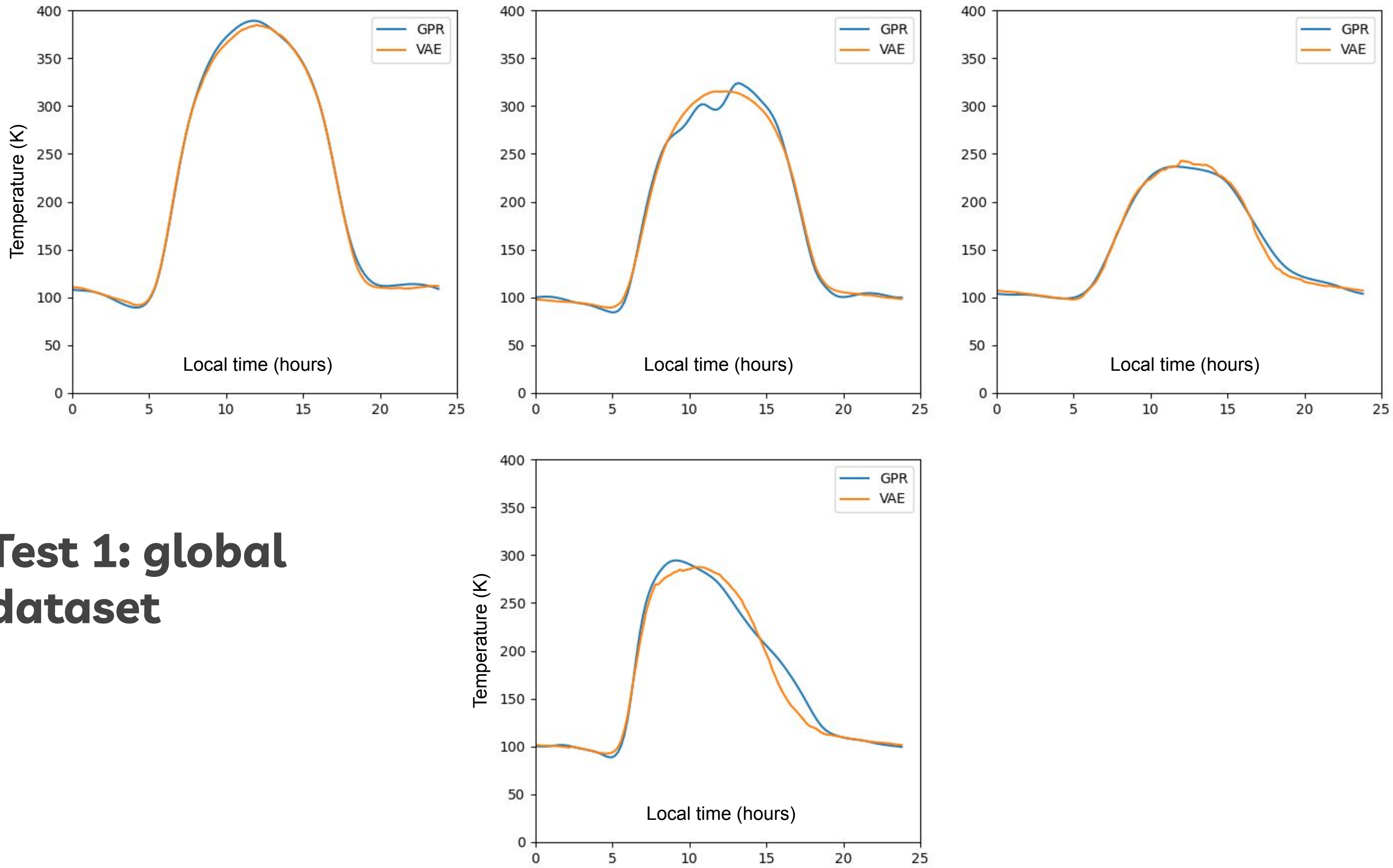
Test 1: global dataset

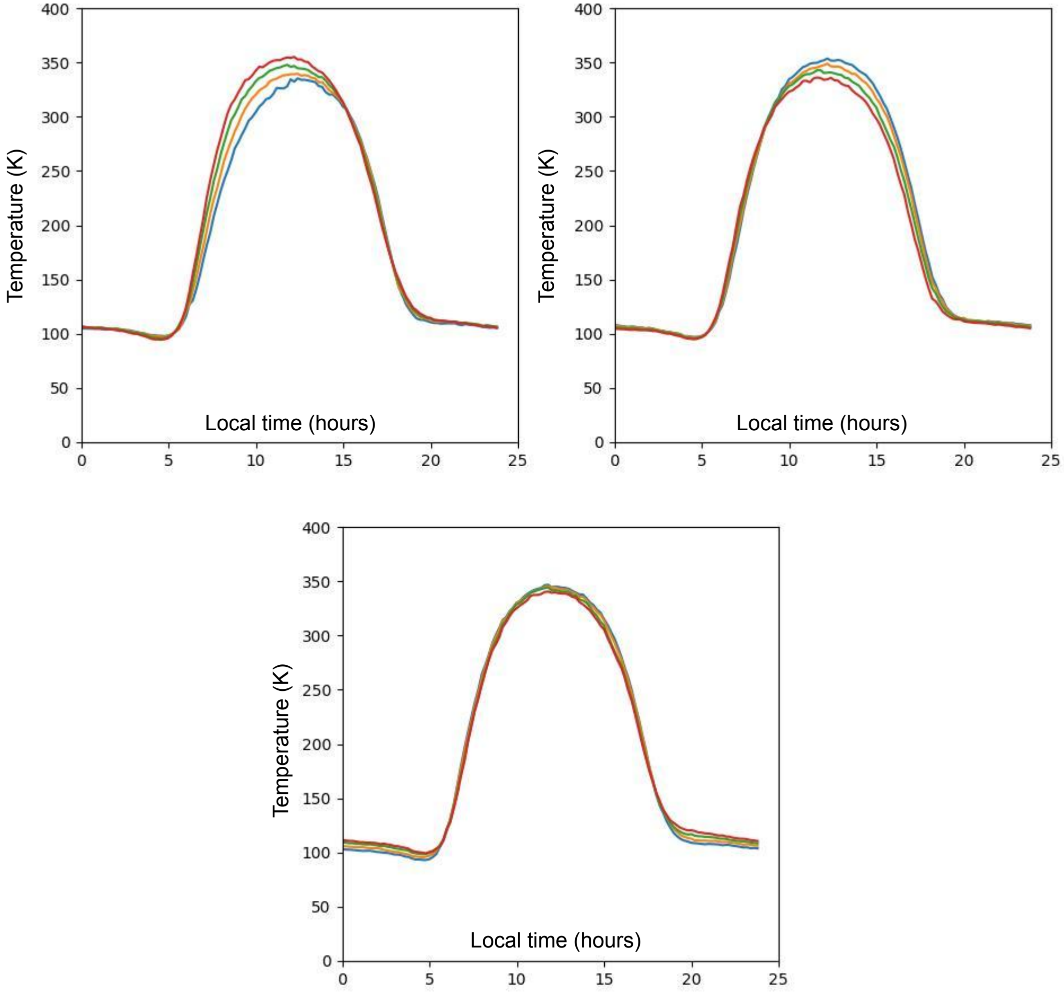
- Decent separation of
  (1) solar onset delay
  (2) effective albedo
  (3) thermal retention

c.f. Ben Moseley's result



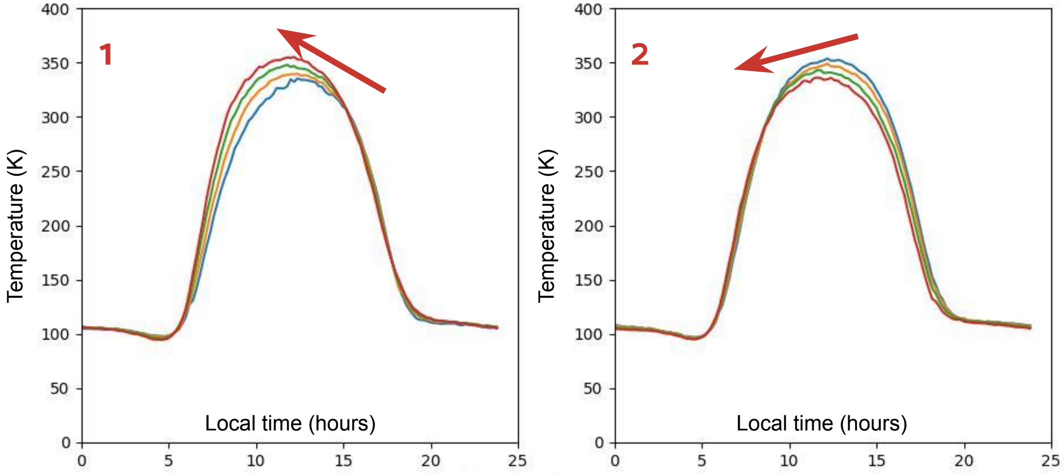
Test 1: global dataset

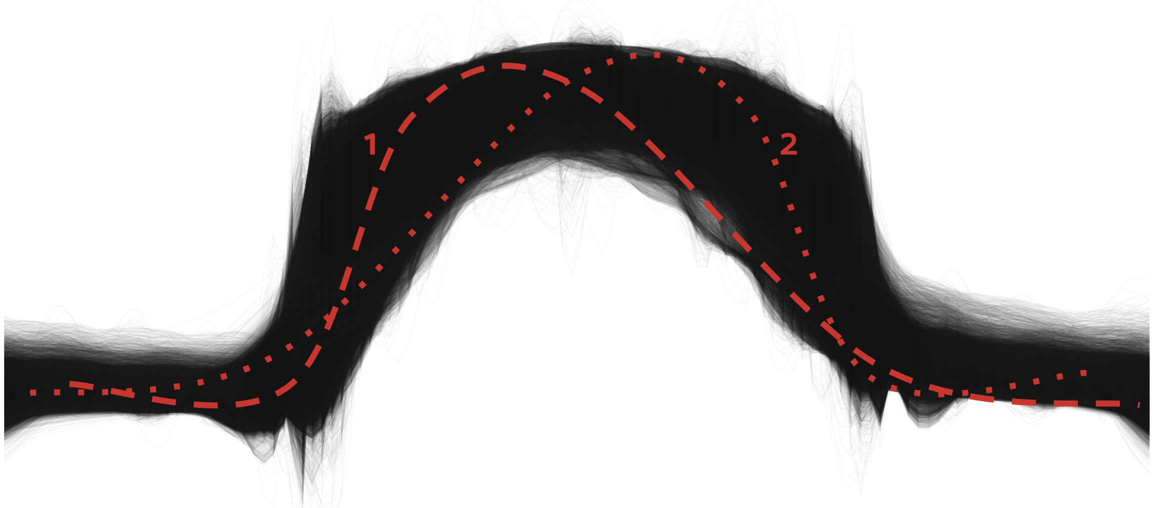




Test 2: “regional” dataset

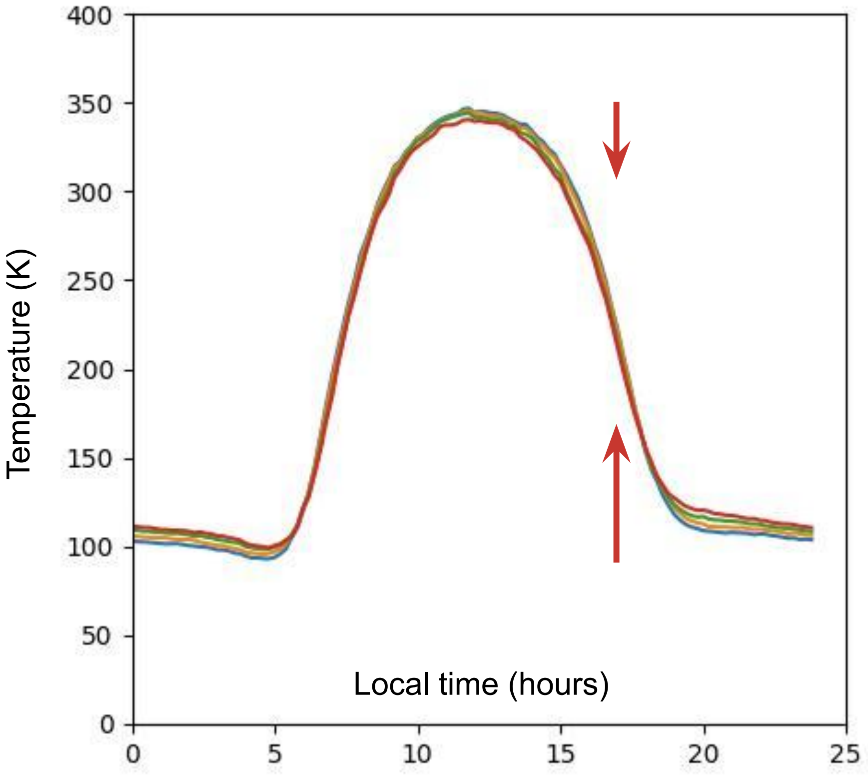
- Separates “solar onset” into two independent latent variables
- Evening T dealt with separately
- LOOONG runtime to smoothen the effect of similar patterns of noise





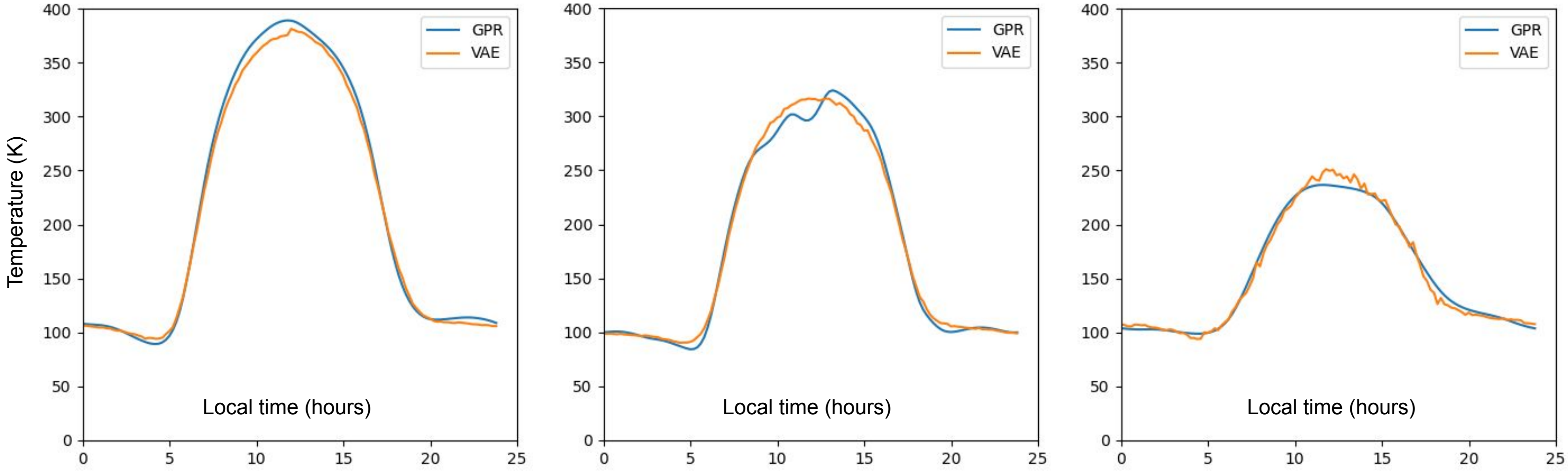
Test 2: “regional” dataset

- **Separates “solar onset” into two independent latent variables**
- Evening T dealt with separately
- LOOONG runtime to smoothen the effect of similar patterns of noise

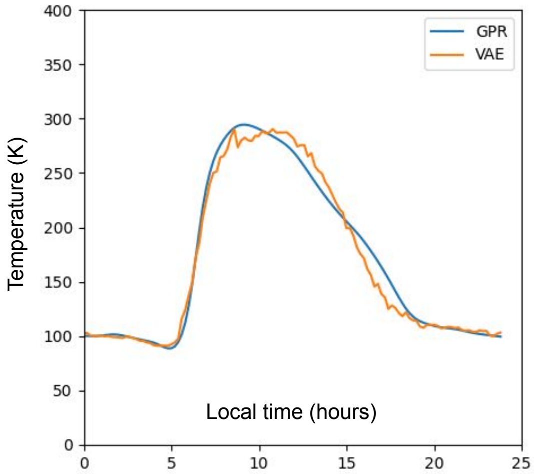


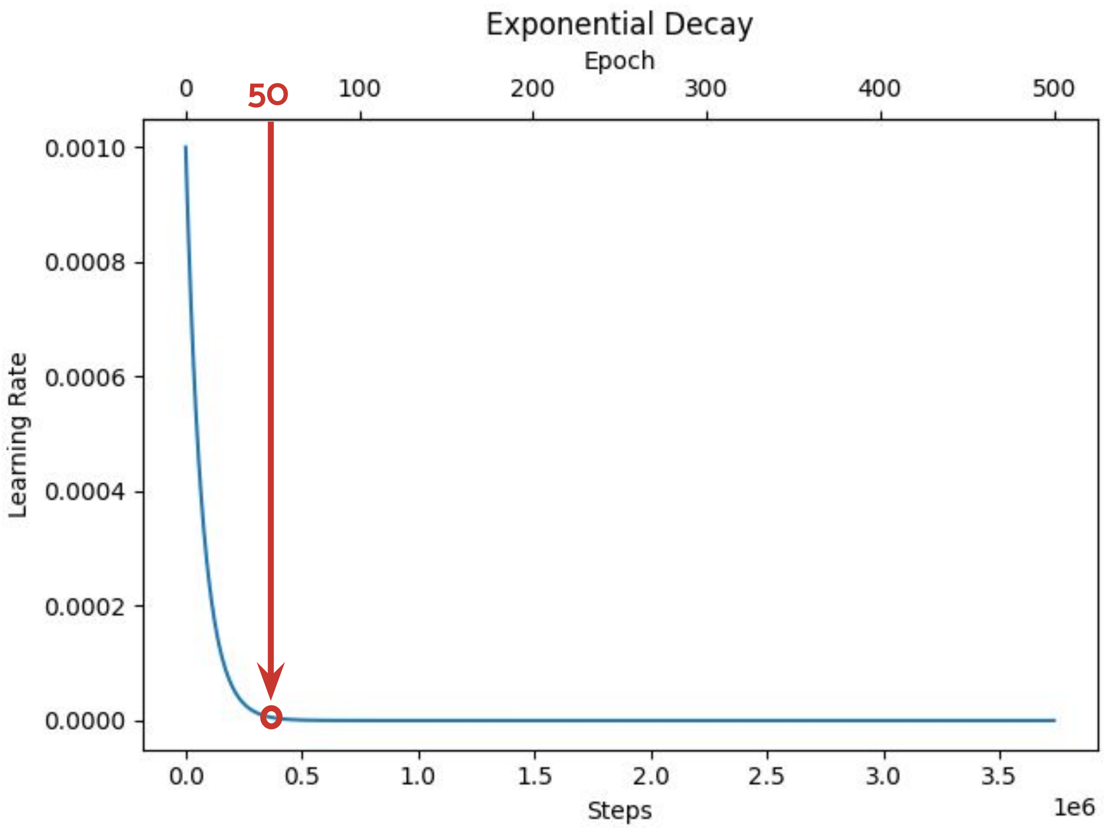
Test 2: “regional” dataset

- Separates “solar onset” into two independent latent variables
- **Evening T dealt with separately**
- LOOONG runtime with adjusted learning rate to smoothen the effect of similar patterns of noise



Test 2: “regional” dataset





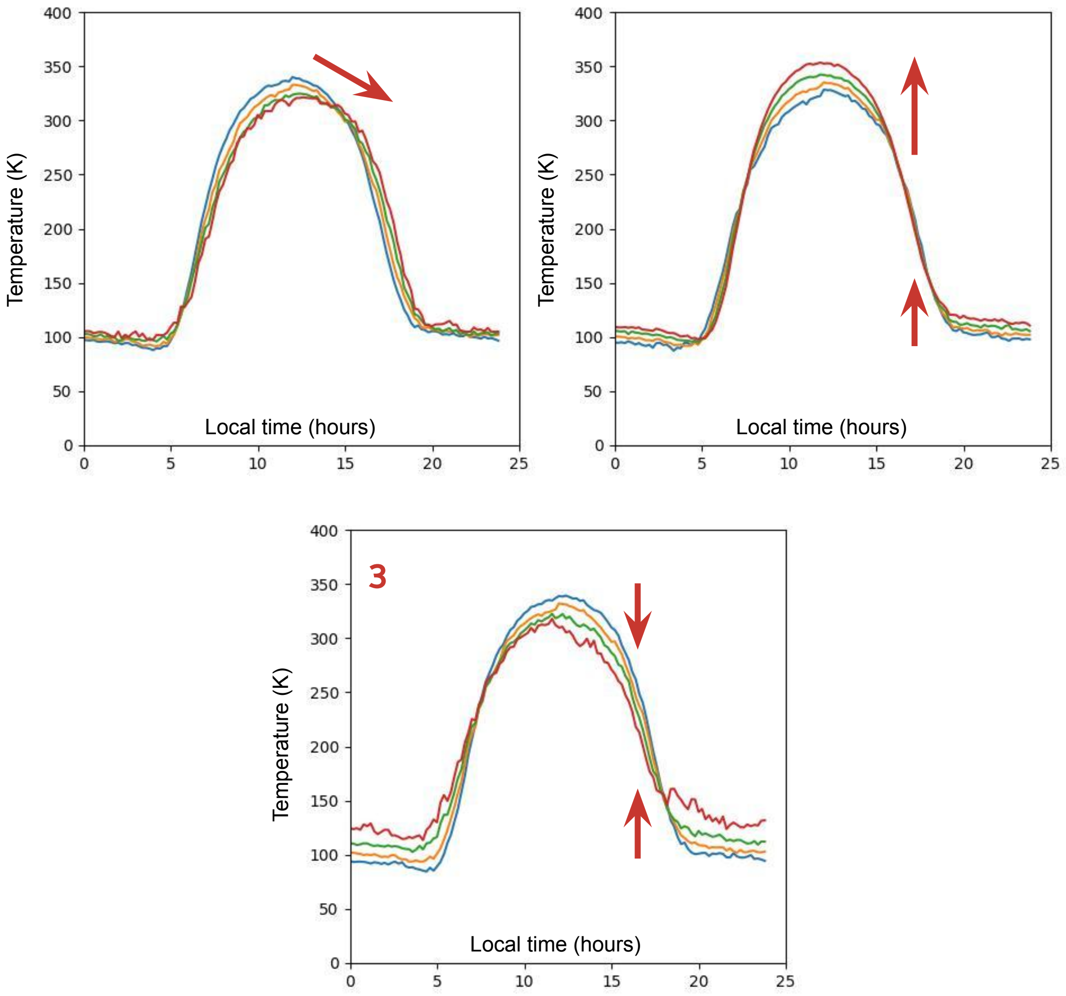
Sensitive to ratio of importance given to $x \neq d(e(x))$ VS "KL divergence"

- ~5 : 1 gives good results

Sensitive to method of normalization

- Normalize each profile
- Normalize with a single mean T, sigma T
- Normalize with a mean profile (mean **T**, sigma **T**)

Sensitive to learning rate

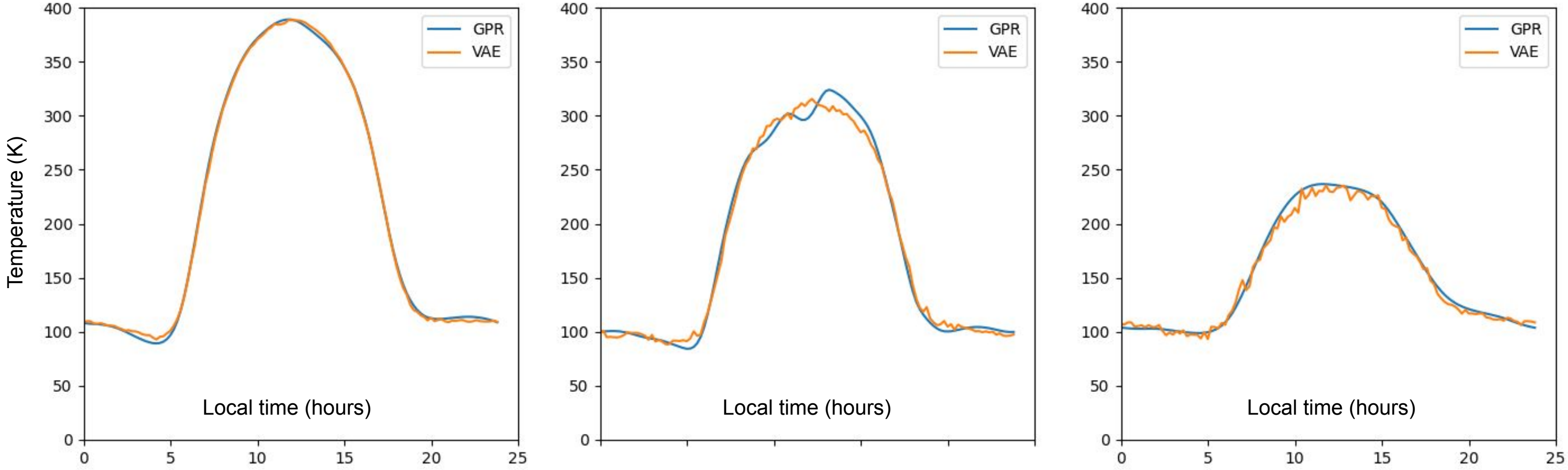


Low learning rate + long learning time

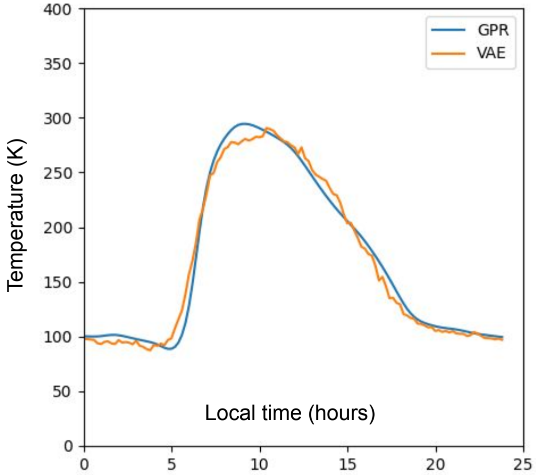
- Set to be exponentially decreasing from 1e-3 at a rate of 0.9 per epoch
- Graphed = 50 epochs
- Learning rate is 2.86e-23

Yet to:

- “smoothen” the latent space
- learn independence of evening T (thermal retention) from peak T (effective albedo) – **#3**

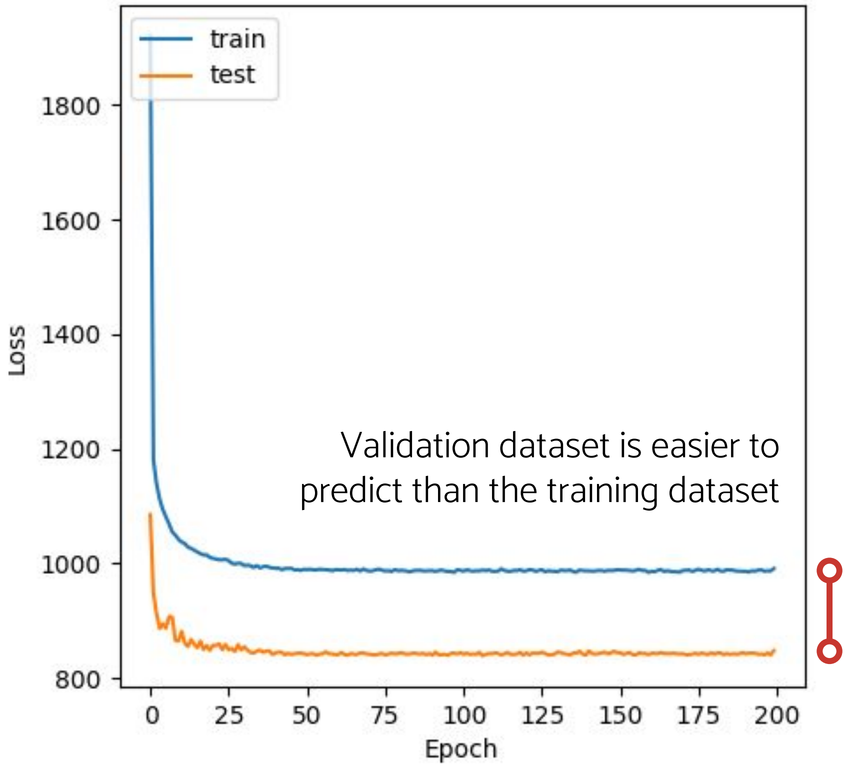


Test 3: early stop



04 SOME FEARS

Train = how well the model is fitting the training data
Test = how well the model is fitting the validation data



Overfitting

- Loss is never increasing so not a concern (yet)

⇒ what are the new learnings after 200 epochs?

Bias / Generalizability

- Same shape = validation dataset provides enough information to evaluate ability of the model to generalize
- But, validation set does not represent training set

⇒ decrease training : testing

Physics-based Loss Functions

- Would help make the model more generalizable to smaller datasets
- Why is learning best with a global dataset?
    - Data-driven approach = absorb as much data as possible to avoid bias
    - More orbits = increased spatio-temporal resolution of data

Pre-define a Latent Variable

- Effect on speed of convergence
- Effect on the determination of the rest of the latent space
- Would help make the model more generalizable to smaller datasets

Other “Explainable” Algorithms

- Benefit of VAE = explainability
    - Understanding the latent space is a step towards understanding the “black box”
    - BUT there are numerous latent spaces (many that don’t make sense) that still seem to provide a good fit for the data
- E.g. gradient boosting (must be non-linear)