UNIVERSITY OF CAMBRIDGE

Supernovae: An Explosion in Data

Matt Grayling
Institute of Astronomy

ioa

Kavli Institute for Cosmology Cambridge
KICC

European Research Council
Established by the European Commission

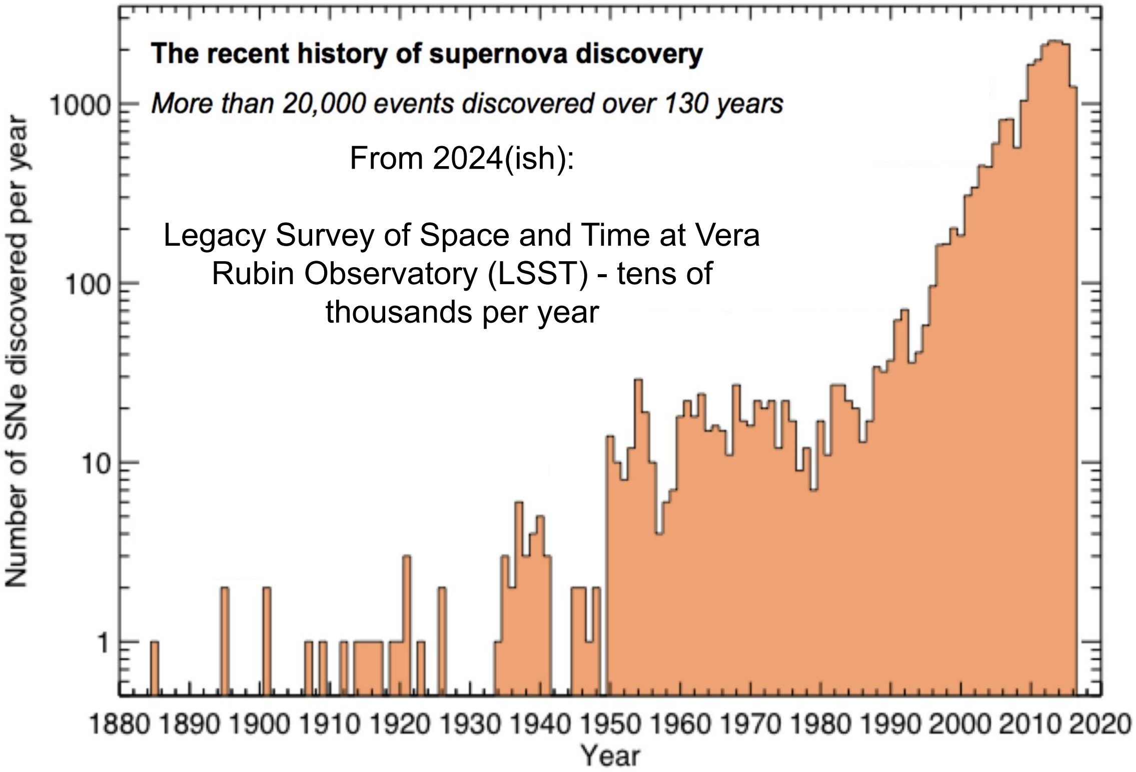
Talk Overview

An exploration of a variety of problems in the areas of machine learning and statistical inference

1) Generative models and supernova classification
2) Scaling Bayesian inference techniques for next generation surveys
3) A (brief) look at simulated-based inference

Part 1: Supernova Classification

The Problem with Supernova Classification



The recent history of supernova discovery

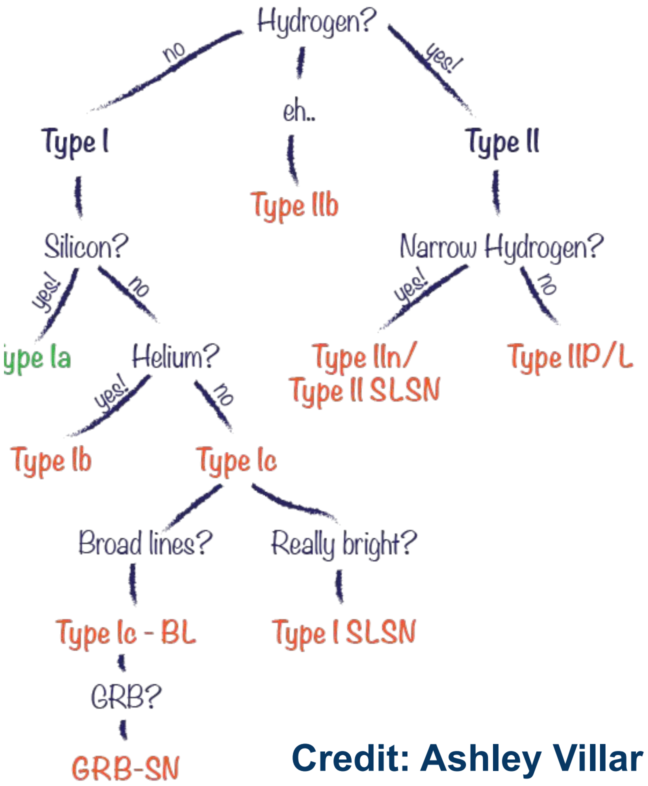
More than 20,000 events discovered over 130 years

From 2024(ish):

Legacy Survey of Space and Time at Vera Rubin Observatory (LSST) - tens of thousands per year

Credit: Mark Sullivan

The Supernova Zoo



Credit: Ashley Villar

Neural networks are great…

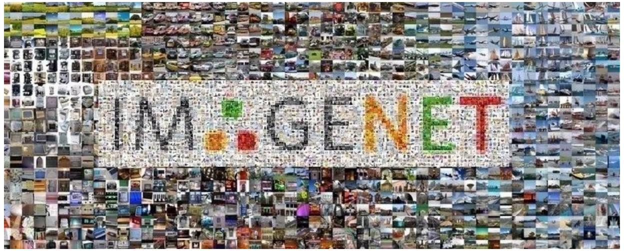
Neural network-based classification of supernova light curves has been a major research goal over the last few years e.g. PlastiCC, ElastiCC:

- Recurrent neural networks
  - RAPID (Muthukrishna+2019)
- Convolutional neural networks
  - Scone (Qu+2021)
- Bayesian neural networks
  - SuperNNova (Möller, de Boissière 2019)
- Variational autoencoders
  - ParSNiP (Boone 2021)

Not an exhaustive list!

…if you have the data to train them

- State-of-the-art industry ML tools train on millions of examples
- For some classes of transient, we have sample sizes ~10-100
- For many scientific applications, we train on large simulated data sets



ImageNet: ~14 million images

OpenAI

GPT - 4

GPT-4: 45 GB of text

Training on Simulations

UNIVERSITY OF CAMBRIDGE

- Good performance on real data requires simulations which are representative of the true population
  - Accurate simulations require good physical understanding of population
  - Supernovae are complex and diverse, simulations fail to reproduce full variability of population
  - Consistent drop in performance when applying models trained on simulations to real data

Generative Models

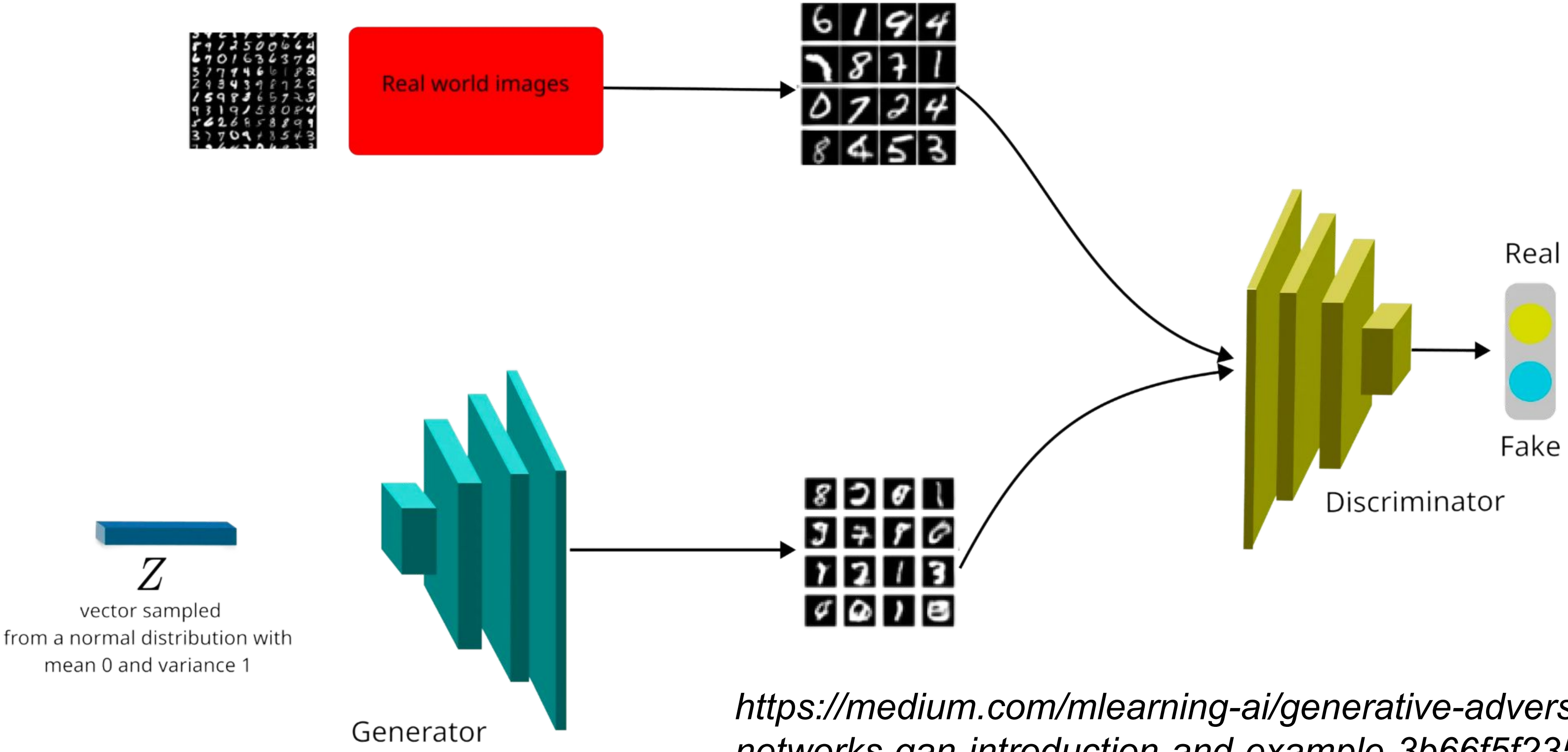
- Generative models allow us to sample from the distribution of supernovae, without assuming any physics

- If trained well, we can generate large synthetic data sets to train classifiers which reproduce full variability of real population

- Augmenting data-sets using generative adversarial networks (GANs) has been shown to improve classification performance within astronomy and beyond (e.g. Motamed+21, Garcia-Jara+22)

“An astronomer using machine learning to study supernovae” - Generated by DALLE 3

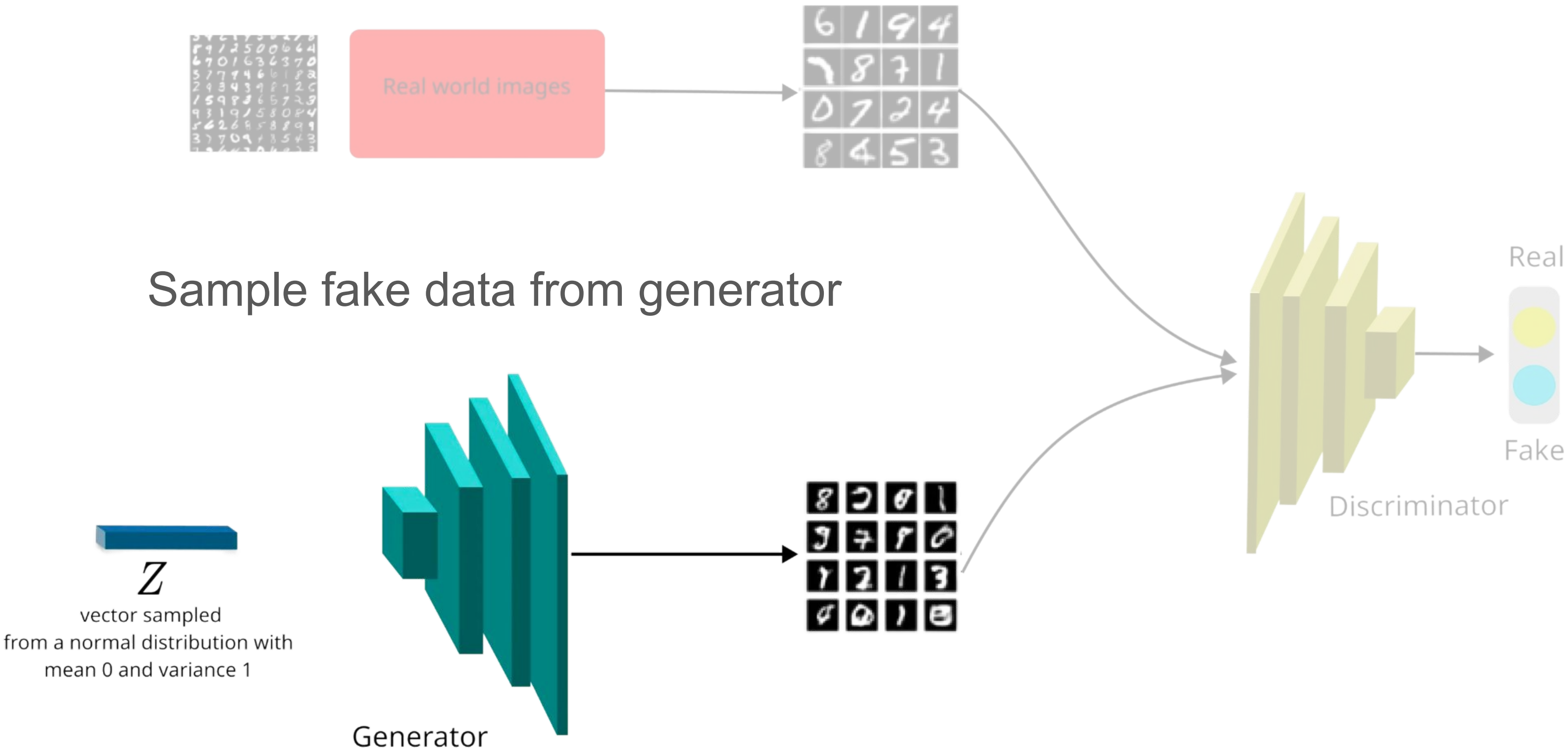
Generative Adversarial Networks (GANs)



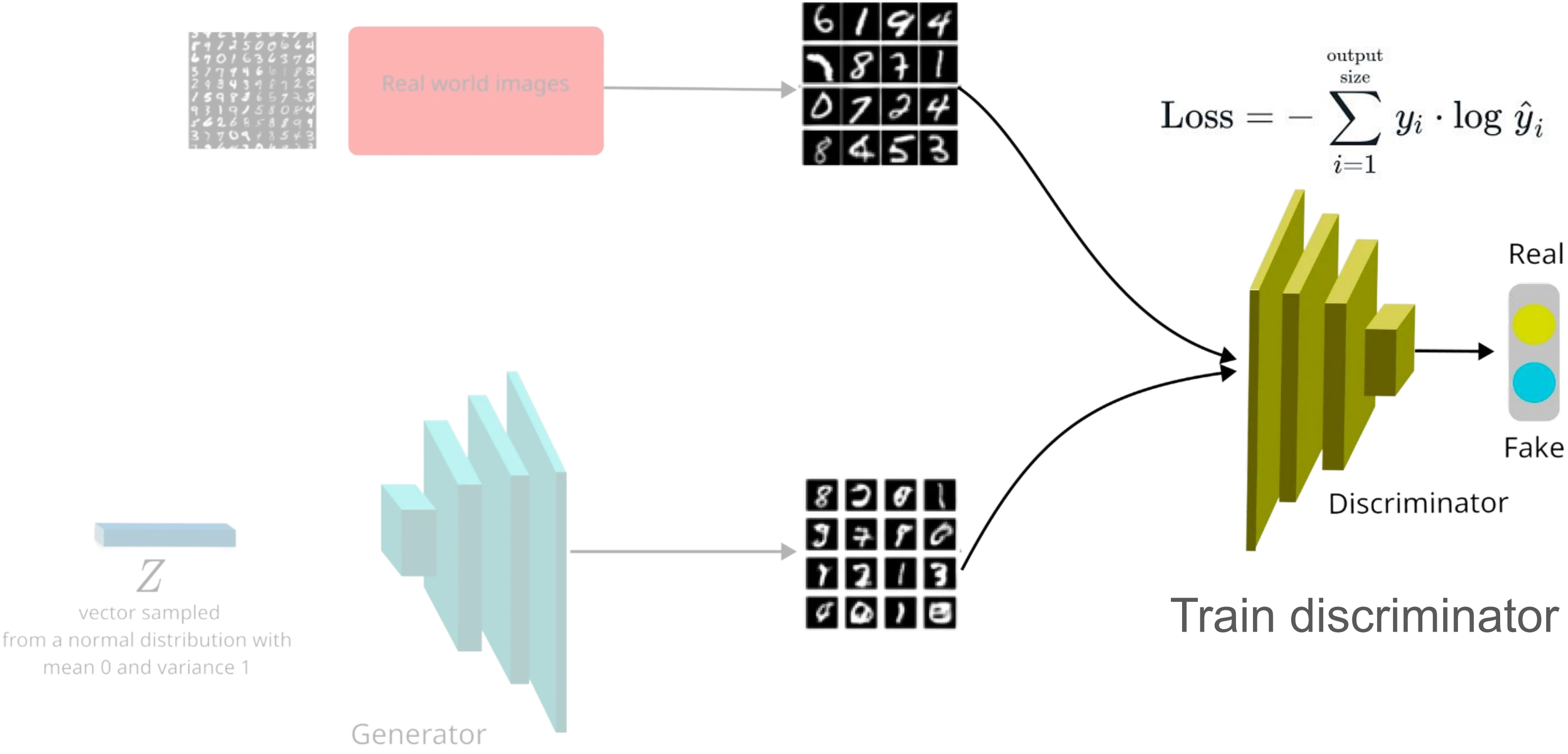
https://medium.com/mlearning-ai/generative-adversarial-networks-gan-introduction-and-example-3b66f5f235e9

Generative Adversarial Networks (GANs)

Sample fake data from generator



Generative Adversarial Networks (GANs)

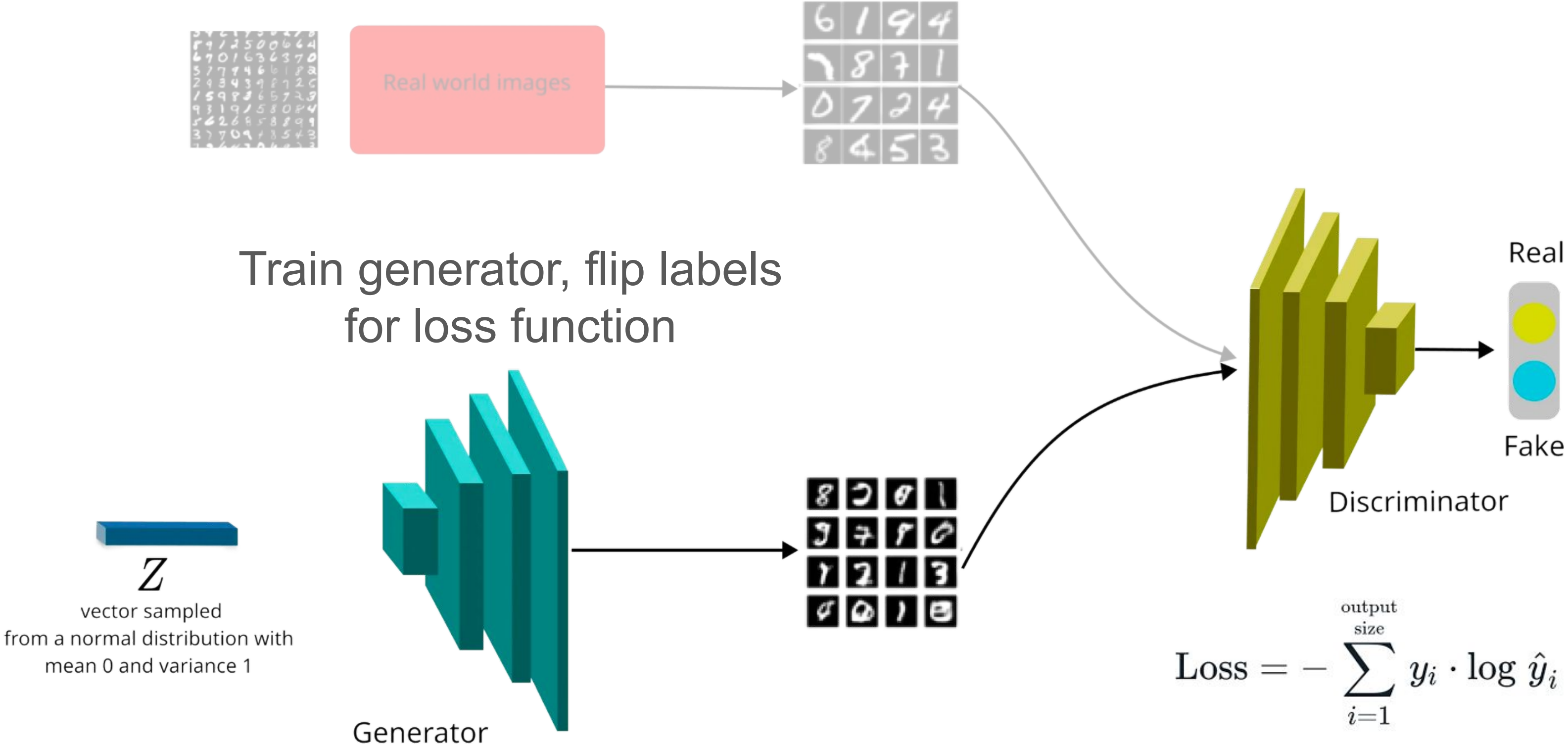


$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i$$

Train discriminator

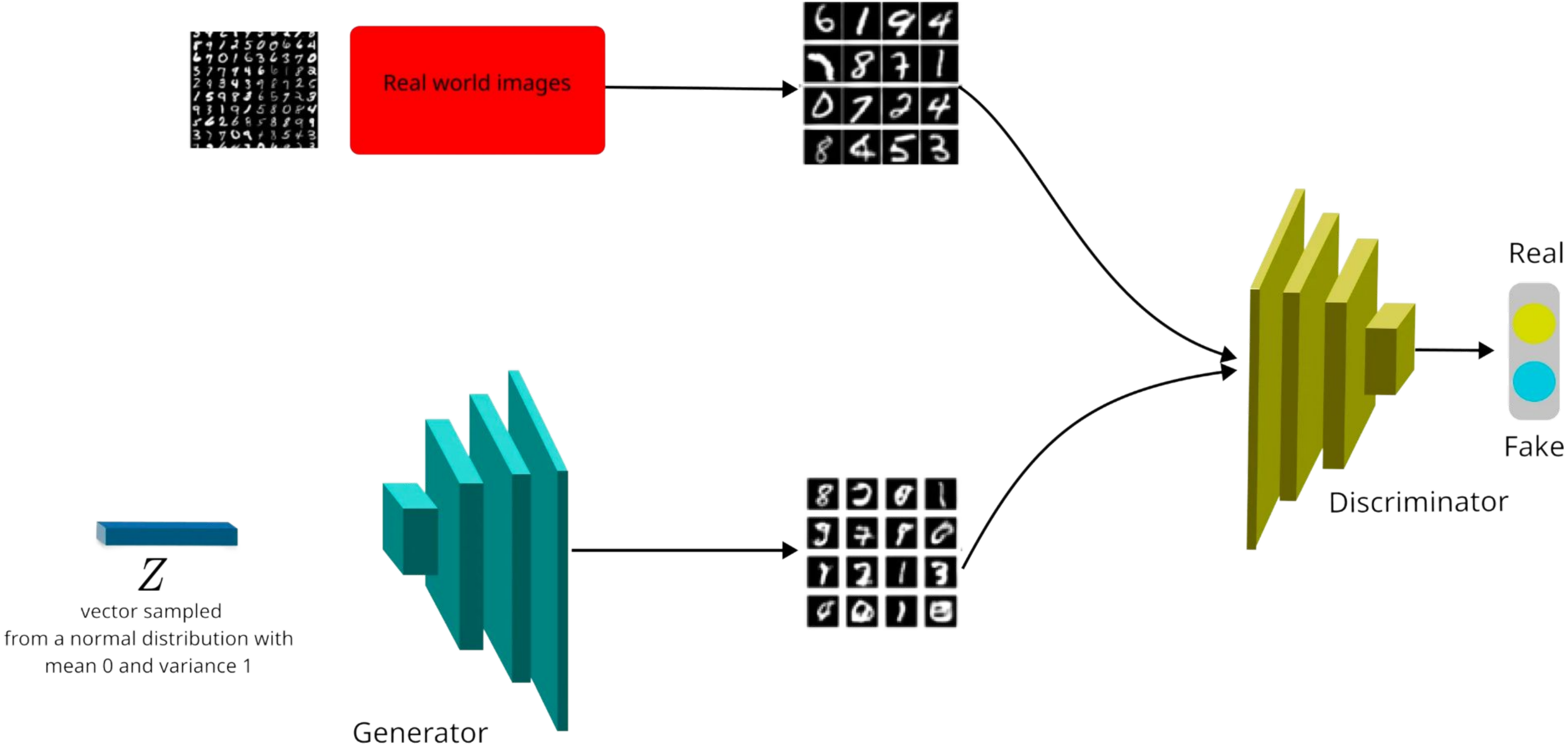
Generative Adversarial Networks (GANs)

Train generator, flip labels for loss function


Real world images
Real
Fake
Discriminator
Z
vector sampled from a normal distribution with mean 0 and variance 1
Generator

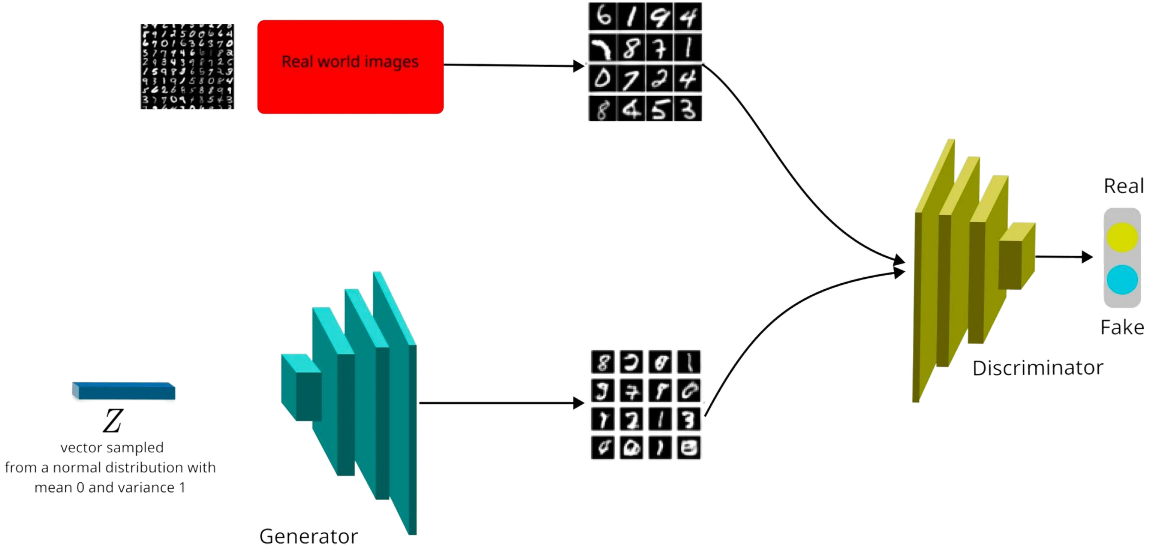
$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\ \text{size}}} y_i \cdot \log \hat{y}_i$$

Generative Adversarial Networks (GANs)

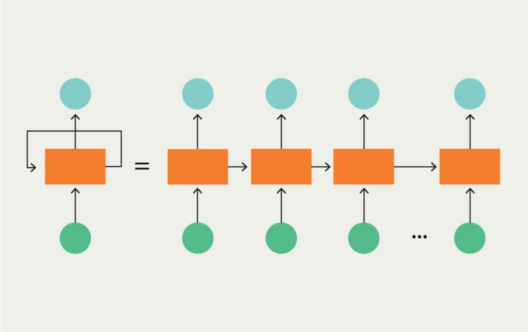


Adapting to variable length sequences

- Typically, generative models focus on generating data with a fixed structure e.g. image of fixed pixel size
- In order to generate realistic SN light curves, we need to generate variable-length time series

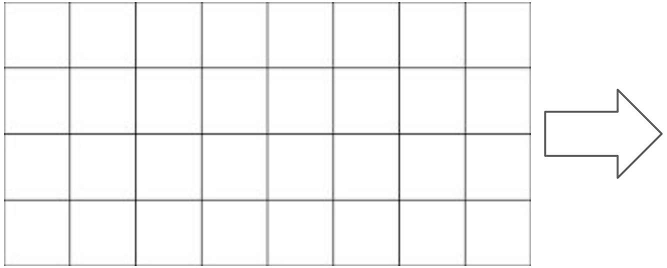


Adapting to variable length sequences



Recurrent neural networks (RNNs) work with variable length sequences, can use RNNs for generator and discriminator

Image generation, pixel grid

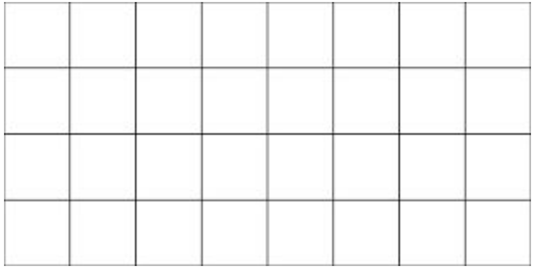


At each timestep, generate phase, photometry and uncertainty

t_g	t_r	t_i	t_z	g	r	i	z	σ_g	σ_r	σ_i	σ_z
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Adapting to variable length sequences

Image generation, pixel grid



$$\mathbf{z} \sim N\left(\mathbf{0}, \underline{\underline{I}}\right)$$

Input noise sampled from n-dimensional latent space

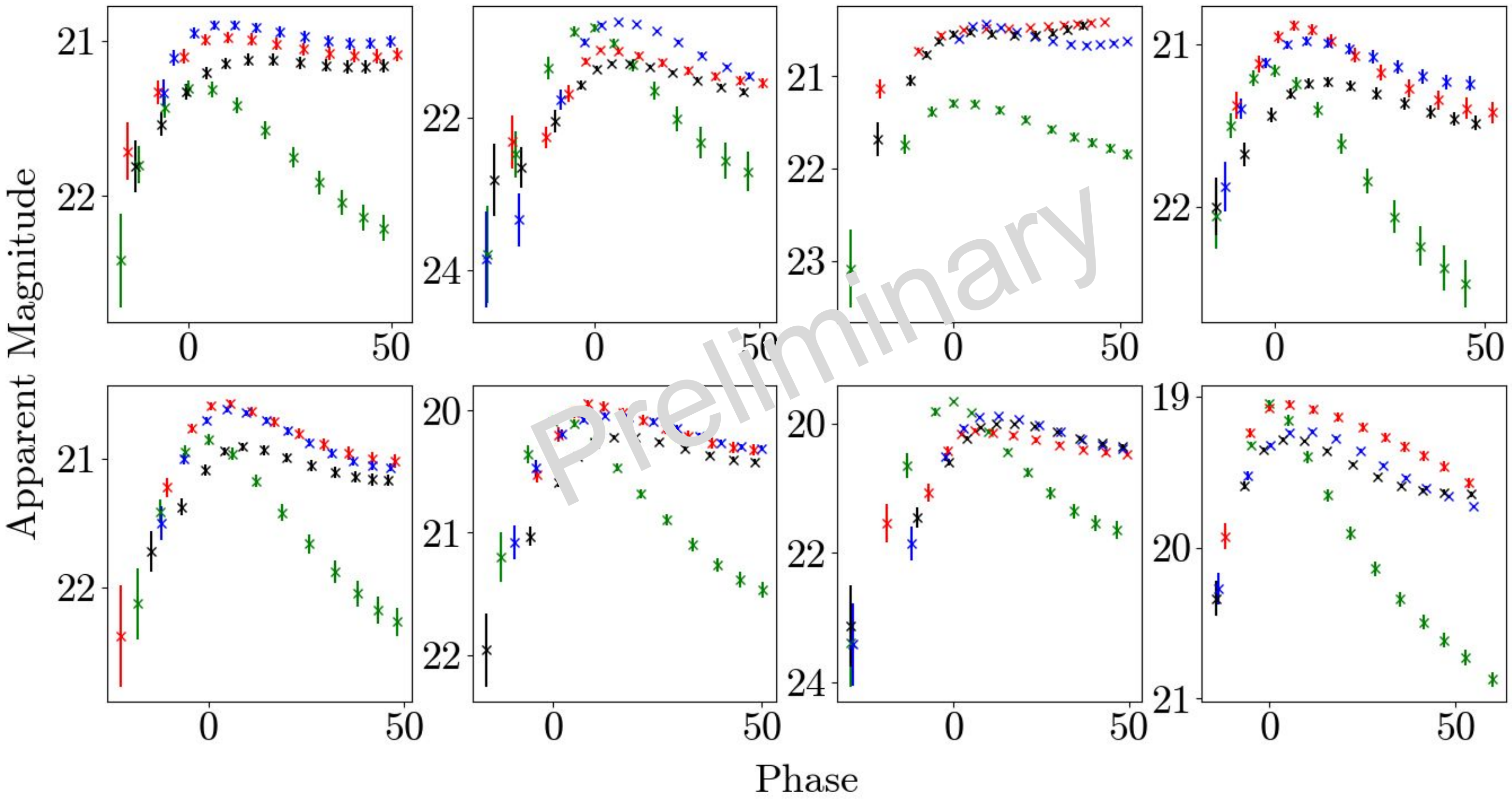
At each timestep, generate phase, photometry and uncertainty

t_g	t_r	t_i	t_z	g	r	i	z	σ_g	σ_r	σ_i	σ_z
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

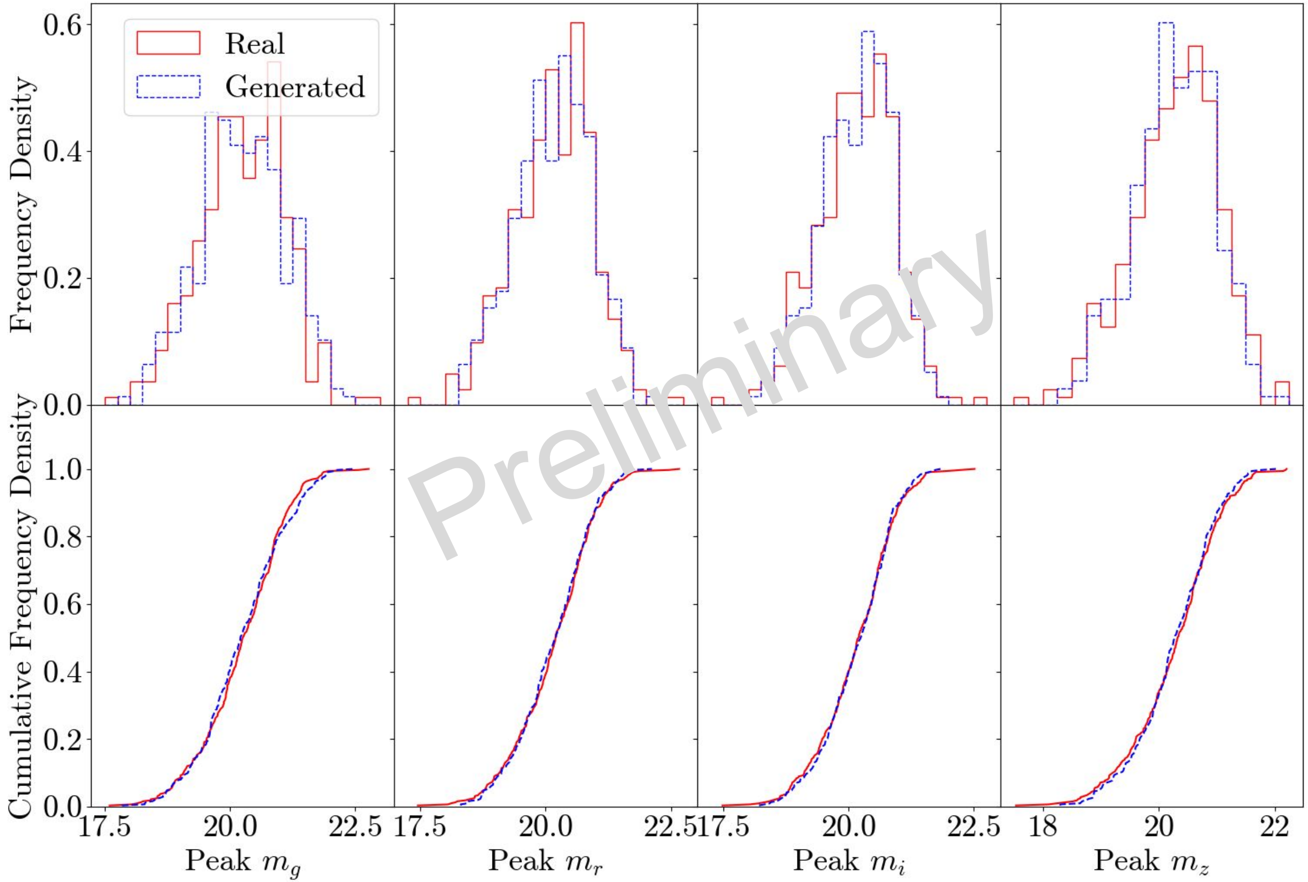
$$\underline{\underline{Z}} = \begin{bmatrix} \mathbf{z} \\ \mathbf{z} \\ \mathbf{z} \\ \vdots \\ \mathbf{z} \end{bmatrix}$$

Input noise sampled from n-dimensional latent space, then repeated N times. Shape of input noise determines length of time series

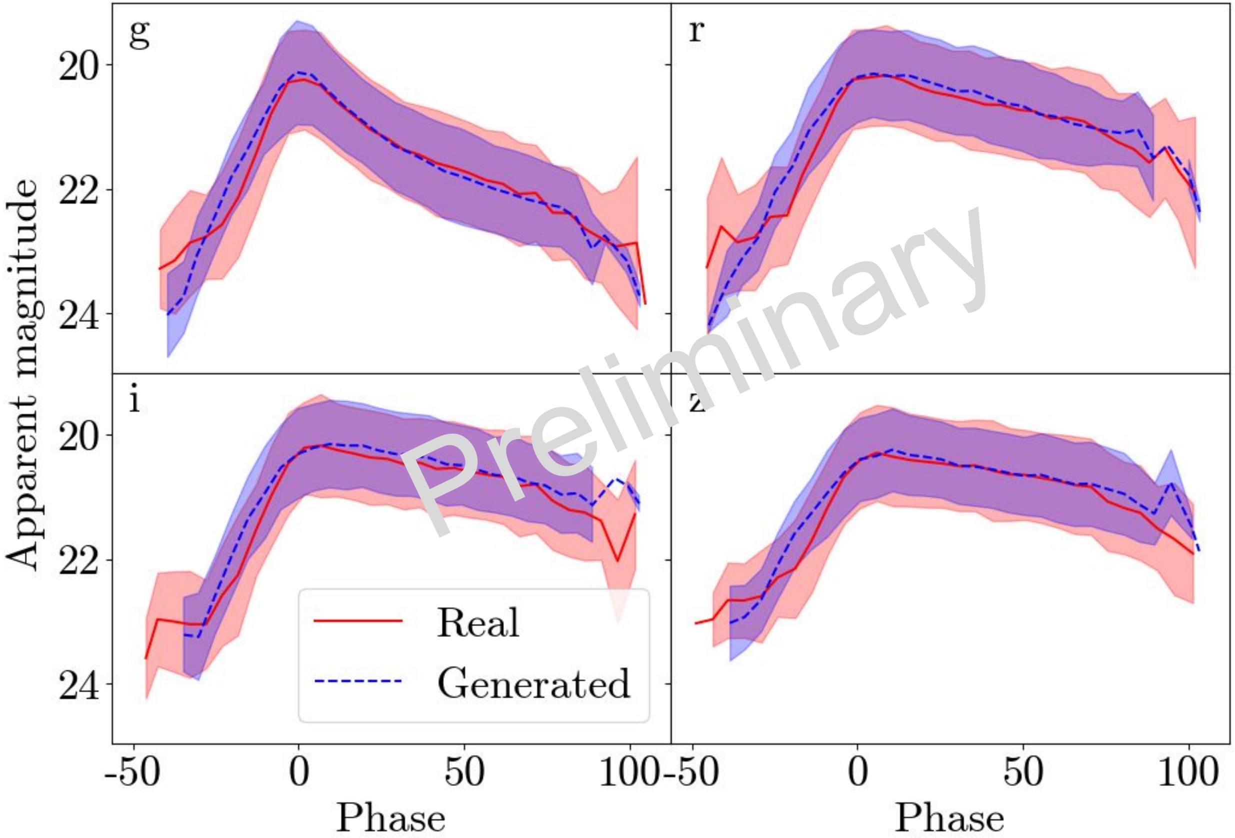
Generated Light Curve Examples



Peak Observed Brightness



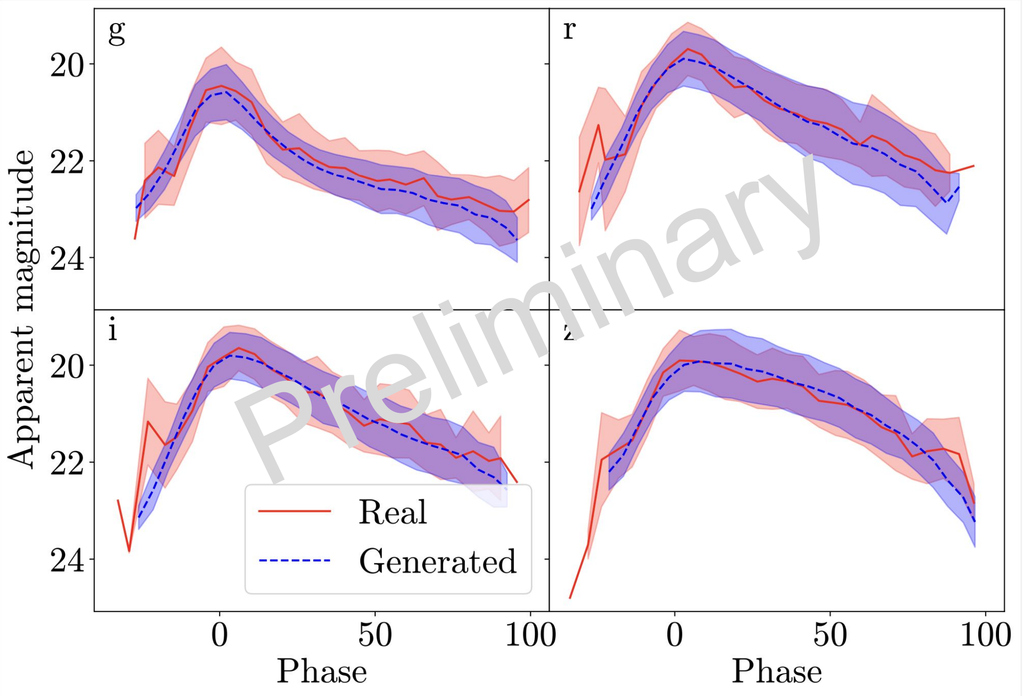
Light Curve Populations



Augmenting Rare Supernova Classes

UNIVERSITY OF CAMBRIDGE

- Previous plots were for sample of ~350 simulated supernovae
- Training on a much smaller data set of ~30 supernovae still produces comparable results
- For rare SN classes with few examples, this approach has the potential to augment very sparse training sets



Next Steps

- GANs have the ability to generate realistic supernova light curves and could help bridge the gap between synthetic and real data
- Next steps:
  - Train a classifier with generated data and assess performance
  - Retrain on real data
  - Explore conditional models, allowing one model to generate all types of supernovae
  - Industry generative models have come on a long way since GANs, explore new architectures e.g. transformers

UNIVERSITY OF CAMBRIDGE

Part 2: Fast Bayesian Inference

Bayesian Inference

$$\mathscr{L} = \frac{1}{\sqrt{2\pi\sigma_f^2}} \exp\left[\frac{(f_m - \hat{f})^2}{2\sigma_f^2}\right]$$

$$\underbrace{P(\theta|D)}_{\text{Posterior}} = \frac{\overbrace{P(D|\theta)}^{\text{Likelihood}}\overbrace{P(\theta)}^{\text{Prior}}}{\underbrace{P(D)}_{\text{Evidence (normalising factor, integral over numerator)}}}$$

Hierarchical Bayesian Modelling

UNIVERSITY OF CAMBRIDGE

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

- For typical Bayesian inference, each supernova would be fit separately

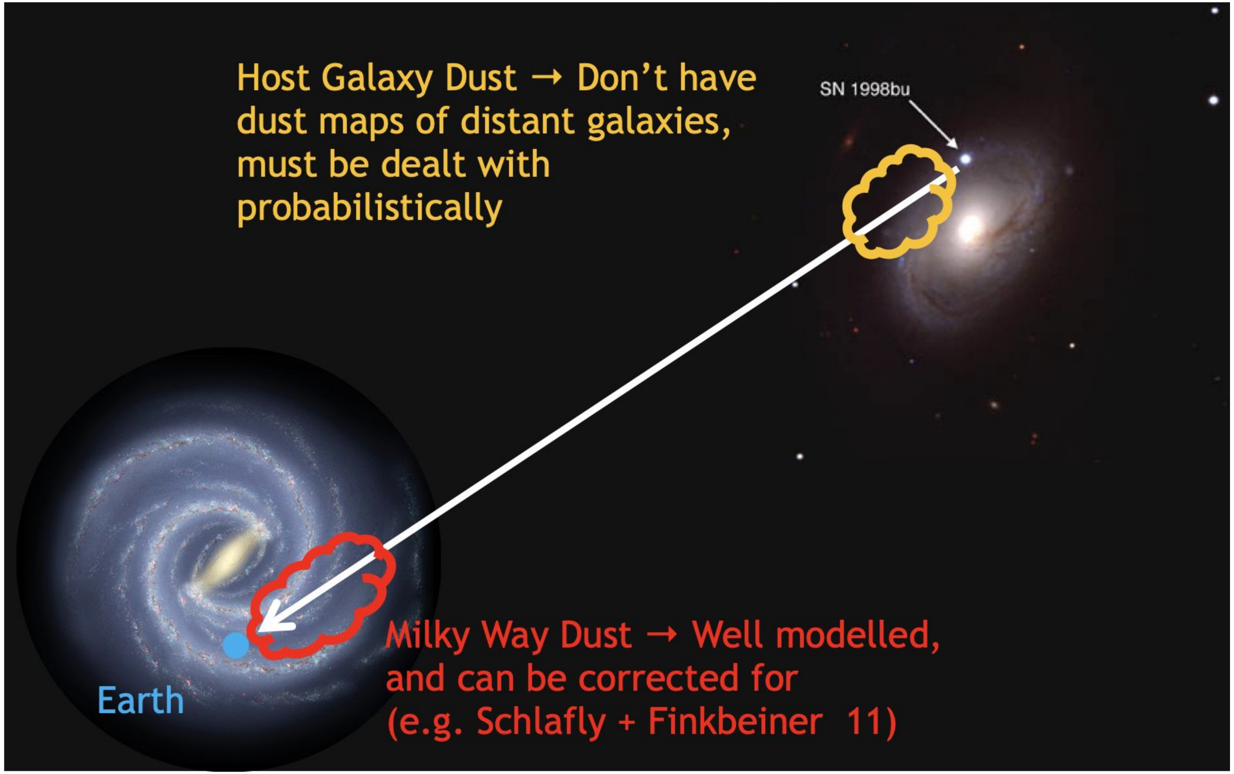
$$R_V \sim U(1,5)$$

- For a hierarchical model, we jointly the sample the posteriors of global and individual supernova properties

$$R_V \sim N(\mu_R, \sigma_R^2)$$

- We can model intrinsic variation and dust properties at the population level as two separate effects

Dust

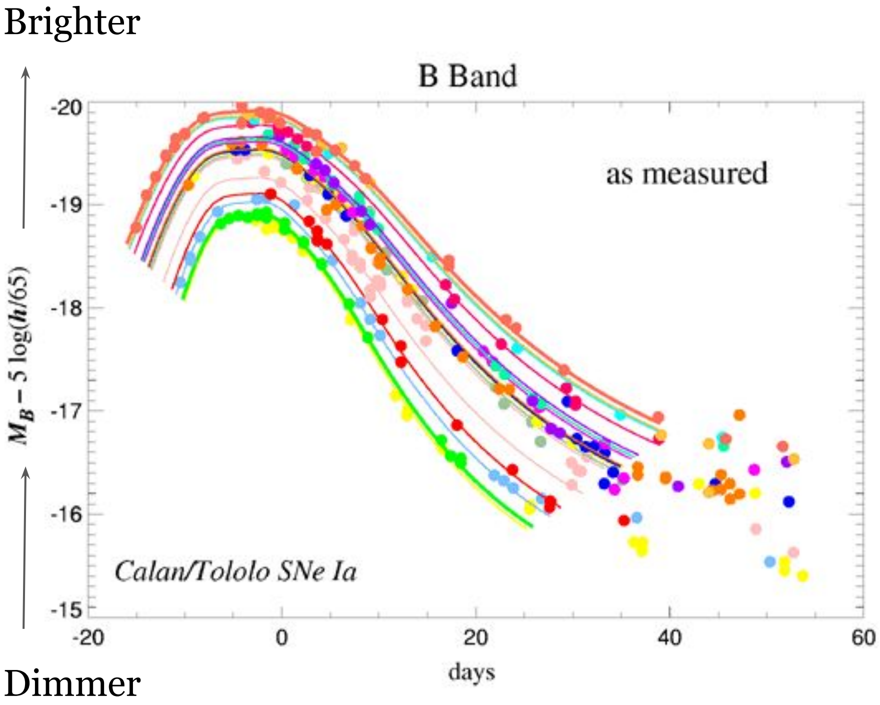


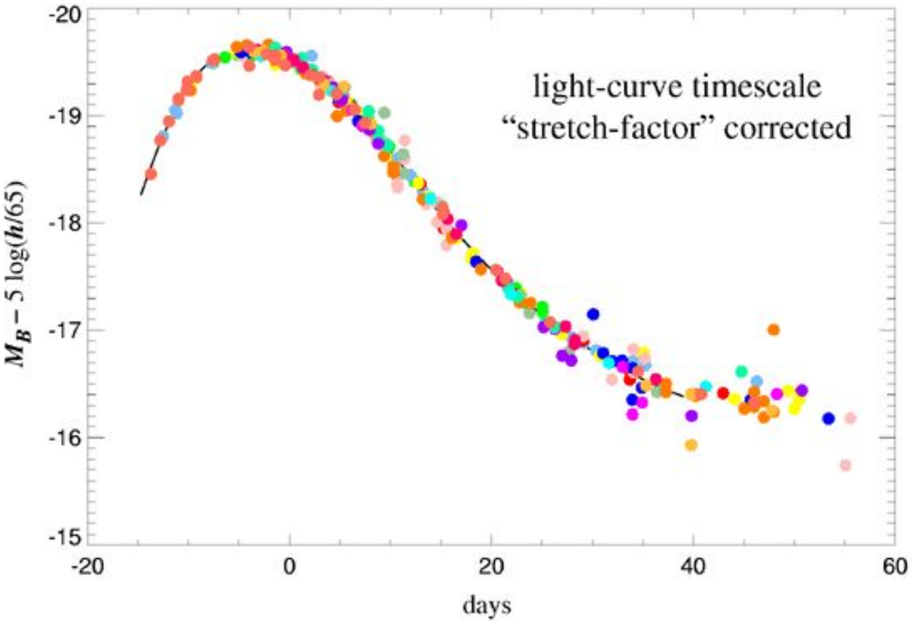
Host Galaxy Dust → Don't have dust maps of distant galaxies, must be dealt with probabilistically

Milky Way Dust → Well modelled, and can be corrected for (e.g. Schlafly + Finkbeiner 11)

NASA/JPL-Caltech/ESO/R. Hurt
Nicholas B. Suntzeff

SNe Ia - Standard(isable) Candles





From Supernova Cosmology Project

Current Approach

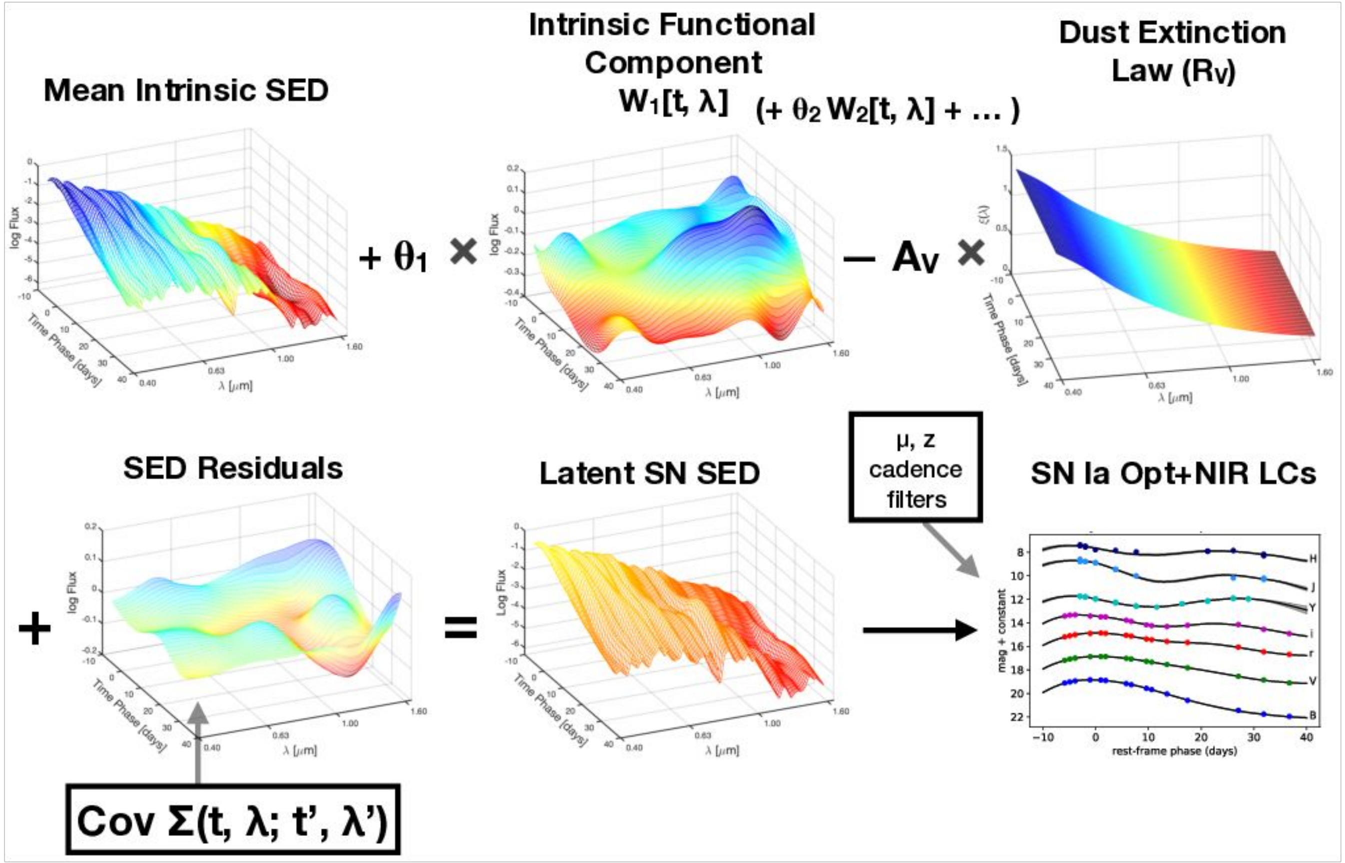
UNIVERSITY OF CAMBRIDGE

Tripp formula

$$\underset{\text{Distance}}{\mu_s} = \underset{\text{Observed brightness}}{m_s} - \overset{\text{“Standard” brightness}}{M} + \overset{\text{Stretch correction coefficient}}{\alpha}\,\underset{\text{Stretch}}{x_s} + \overset{\text{Colour correction coefficient}}{\beta}\,\underset{\text{Observed B-V colour}}{c_s}$$

- “c” - one parameter for two independent effects (dust and intrinsic differences)
- Correctly handling dust (and SN Ia colour–luminosity correlations more generally) is key to correctly estimating SN Ia distances
- If intrinsic effect misattributed to dust, could lead to bias

The BayeSN Model



Mean Intrinsic SED

+ θ_1 ×

Intrinsic Functional Component $W_1[t, \lambda]$ (+ θ_2 $W_2[t, \lambda]$ + …)

− A_V ×

Dust Extinction Law (R_V)

+

SED Residuals

Cov $\Sigma(t, \lambda; t', \lambda')$

=

Latent SN SED

μ, z
cadence
filters

SN Ia Opt+NIR LCs

Mandel (2020)

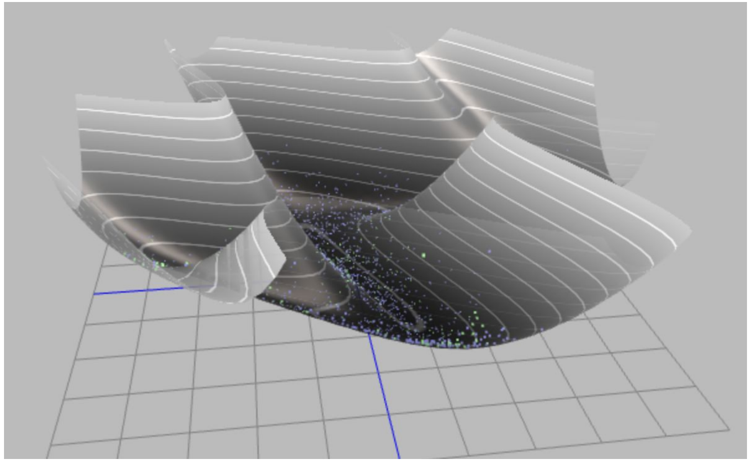
Challenges with BayeSN

UNIVERSITY OF CAMBRIDGE

- Lots of parameters (e.g ~400 global, ~4200 latent parameters for Thorp 2021)
- Complex likelihood evaluation:
    - Compute SED model
    - Evaluate numerical integrals through different filters, for each time of observation and each supernova
- Computationally expensive

Isn't this a machine learning workshop?

- A model with thousands of parameters, fed through a series of large tensor operations to calculate a performance metric
  - Sounds a little like a neural network
- For a neural network, we use gradient-based optimisers to explore the parameter space, looking for the optimal solution
- For Bayesian inference, we usually want a posterior distribution over those parameters
  - Use Hamiltonian Monte Carlo (HMC) to sample posteriors or Variational Inference to fit them



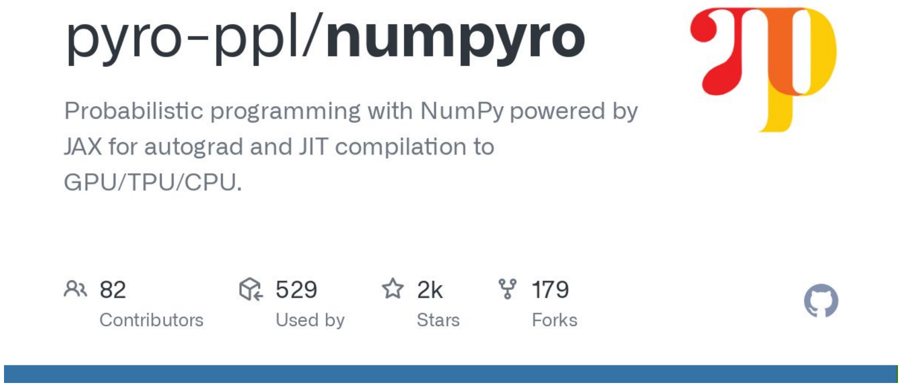
Credit: Alex Rogozhnikov

Scaling BayeSN

Problem:

Scaling BayeSN for next generation data sets without compromising functionality

Solution:



pyro-ppl/**numpyro**

Probabilistic programming with NumPy powered by JAX for autograd and JIT compilation to GPU/TPU/CPU.

82 Contributors

529 Used by

2k Stars

179 Forks

Jax and Numpyro

UNIVERSITY OF CAMBRIDGE

Jax

- Python package very similar to numpy, but includes JIT compilation at runtime for any device including GPUs
- Supports autodiff (automatic calculation of exact gradients), ideal for HMC or variational inference
- Very efficient matrix/tensor operations, perfect for numerical integrals

Numpyro

- Probabilistic programming package for Python built on Jax

Vectorized likelihood evaluation + GPUs = Fast Bayesian inference

Examples

Mandel+20:

- Trained on 79 *BVriYJH* light curves compiled in Avelino+19
- Previous training ~5 days → now ~20 minutes

Thorp+21:

- Trained on 157 *griz* light curves from Foundation DR1
- Previous training ~1 day → now ~10 minutes

1000 simulated optical SNe:

- Training (conditioning global parameters on data)
  - 45 mins
- Fitting (inference of supernova properties)
  - 15 mins
- Fitting with variational inference (work by Ana Sofia Uszoy)
  - 2.5 mins

Scaling to Next Generation Surveys

- Many overlaps between neural networks and Bayesian inference
- Previously, applying hierarchical Bayesian models to data-sets > 10,000 SNe would have been computationally unfeasible
- The use of numpyro + GPUs makes this achievable in relatively short timescales
- We are able to scale Bayesian inference approaches to LSST-size data sets

Part 3: Simulation-based Inference

Simulation-based Inference (SBI)

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

Likelihood

$$\mathscr{L} = \frac{1}{\sqrt{2\pi\sigma_f^2}} \exp\left[\frac{(f_m - \hat{f})^2}{2\sigma_f^2}\right]$$

Gaussian likelihood - easy to calculate

$$\mathscr{L} = ?$$

Sometimes the likelihood is too complex to write down

Selection Effects

Examples of selection criteria

- More than N observations with signal-to-noise > 3/4/5
- More than M observations before/after peak
- $|x_1| < 3$, $|c| < 0.3$
- Need a spectroscopic host redshift

We have developed tools to simulate supernova surveys well, but expressing their selection effects analytically is impossible

One example - ratio estimation

$$P(\boldsymbol{\Theta}|\mathbf{D}) = \frac{P(\mathbf{D}|\boldsymbol{\Theta})P(\boldsymbol{\Theta})}{P(\mathbf{D})} \equiv r(\boldsymbol{\Theta}, \mathbf{D})P(\boldsymbol{\Theta})$$

$$r(\boldsymbol{\Theta}, \mathbf{D}) = \frac{P(\mathbf{D}|\boldsymbol{\Theta})}{P(\mathbf{D})} = \frac{P(\boldsymbol{\Theta}, \mathbf{D})}{P(\boldsymbol{\Theta})P(\mathbf{D})}$$

Can estimate with neural network

Just train a binary classifier on simulations - “does this data set match this set of parameters?

Why use SBI?

- Many cases arise in astronomy where we can simulate an effect we cannot write down analytically
- SBI provides a tool for doing statistical inference in cases which would otherwise be impossible or require a number of (incorrect) assumptions
- Loads of SBI tools in existence:
    - Normalising Flows, Neural Posterior Estimation, Neural Ratio Estimation, …
- SBI++ package (https://github.com/wangbingjie/sbi_pp) is a good place to get started if you are interested

Conclusions

Conclusions

UNIVERSITY OF CAMBRIDGE

1) Generative models have the potential to help combat data drift between real and simulated data, improving classifiers

2) Bayesian inference shares many similarities with neural networks which enable order-of-magnitude performance increases using GPUs, making it scalable for LSST-size data sets

3) Simulation-based inference will be an increasingly important, machine learning-based approach to statistical inference going forward

Postdoc opportunities at Cambridge in supernovae / astrostatistics / data science... apply by January 2, 2024 (GMT)! https://www.jobs.cam.ac.uk/job/43803/