# RAPID Real-time Automated Photometric IDentification

Early classification of transients

#### **Daniel Muthukrishna**

University of Cambridge

Collaborators: Gautham Narayan, Kaisey Mandel, Rahul Biswas, Renee Hložek



# The known transient universe

- The transient universe remains largely mysterious
- New surveys will observe observe an unprecedented number of transients
- Need to prioritize follow-up based on class and epoch
- Automated, fast, early classifications are required



## Simulated dataset - PLAsTiCC

- > A comprehensive real training dataset isn't available
- PLAsTiCC software used to create simulations with the observing properties of ZTF
- > 48000 ZTF light curves split between 12 transient classes





## **Previous classification attempts**

- > Require full phase coverage of each light curve
- > Often only SNe or SNIa vs non-SNIa classifications
- Often require user-defined feature extraction before classification
- > Template matching (slow)
- > No early classification



## **RAPID: Early and real time classifications**

- Automatically identify transients from within a day of the initial alert to the full life-time of the light curve
- Classifier is trained on 60% of the dataset and is validated on the remaining 40%



# **RAPID Design**

- > Takes multiband photometric information and contextual information as input
- > Two classifiers: with and without known redshift



# **Preprocessing light curves**

- > Exclude galactic objects
- > Correct for extinction
- Correct for time dilation and distance if redshift is known
- Estimate explosion time by modelling early part of the light curve with a quadratic step function
- > Define Pre-explosion  $(t < t_0)$  and transient phase  $(t \ge t_0)$



## **Deep Recurrent Neural Network**



#### **Classification performance**



### **Classification performance**





True label

#### **Confusion matrices**

	-1.00		40  days since trigger										1.00					
	1.00	Р	re-explosion	_	-	-	-	-	-	-	-	-	-	-	-	-		1.00
	-0.75		SNIa-norm	0.00	0.92	0.00	0.02	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00		0.75
			SNIbc	0.00	0.05	0.31	0.02	0.06	0.20	0.03	0.01	0.00	0.01	0.01	0.26	0.05		
	-0.50		SNIL	0.00	0.09	0.07	0.49	0.02	0.05	0.01	0.00	0.04	0.03	0.02	0.10	0.08		0.50
	-0.25	bel	SNIa-91bg	0.00	0.01	0.05	0.01	0.83	0.06	0.02	0.00	0.00	0.00	0.00	0.01	0.00		0.25
			SNIa-x	0.00	0.09	0.03	0.01	0.01	0.74	0.01	0.00	0.00	0.00	0.00	0.07	0.03		
	-0.00	e lal	point-Ia	0.01	0.00	0.00	0.00	0.01	0.00	0.84	0.11	0.00	0.00	0.02	0.00	0.00		0.00
		Tru	Kilonova	0.01	0.00	0.01	0.00	0.00	0.00	0.07	0.90	0.00	0.00	0.01	0.00	0.00		
	-0.25		SLSN-I	0.00	0.02	0.00	0.02	0.00	0.02	0.00	0.00	0.85	0.09	0.00	0.00	0.00		-0.25
	0.50		PISN	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.02	0.93	0.00	0.00	0.02		
			ILOT	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.93	0.01	0.01		-0.50
	-0.75		CART	0.00	0.00	0.09	0.02	0.01	0.11	0.03	0.00	0.00	0.00	0.01	0.70	0.03		-0.75
			TDE	0.00	0.02	0.02	0.01	0.00	0.02	0.00	0.00	0.03	0.03	0.01	0.01	0.86		
	-1.00			ion	rm	Ibc-	·IIN	lbg-	a-X.	-Ia	ova-	-I-N	SN.	ĊŢĊ	RT.	DE-		-1.00
				plos	la-no	SN	S	Ia-91	INS	point	<i>ilon</i>	SLS	Id	IL(	CA	E		
				re-ey	SNI			SN		[	X							
			Predicted label															



True label

#### **Classification Performance**





# **Python interface**

#### > pip install astrorapid

A astrorapid	Docs » Usage Q Edit on GitHub
Search docs	TT
CONTENTS:	Usage
Installation	Classify Light curves
Usage	
Classify Light curves	Use the following example code:
Train your own classifier with your own data	<pre>from astrorapid.classify import Classify</pre>
ample	<pre># Each light curve should be a tuple in this form. Look at the example code for an example of light_curve_info1 = (mjd, flux, fluxerr, passband, zeropoint, photflag, ra, dec, objid, redshi light_curve_list = [light_curve_info1,]</pre>
	# Classify Light curves
	<pre>classification = classify(light_curve_list, known_redshift=)rue) predictions = classification.get_predictions() redshift=ord</pre>
	print(predictions)
	<pre># Plot light curve and classification vs time of the light curves at the specified indexes classification.plot_light_curves_and_classifications(indexes_to_plot=(0,1,4,6)) classification.plot_classification_animation(indexes_to_plot=(0,1,4,6))</pre>
	<pre>classification.plot_classification_animation(indexes_to_plot=(0,1,4,6))</pre>
	Train your own classifier with your own data

## **Integration into Transient Brokers**

- > Currently integrating RAPID into the ANTARES transient broker
- > Aiming to classify objects in the ZTF live data stream



## Conclusions

- RAPID enables prioritized follow-up of new large-scale transient surveys based on transient class and epoch
- Early classification: The use of a Recurrent Neural Network allows us to classify transients as a function of time
- We can identify 12 different transient classes within days of its explosion, despite low S/N data and limited colour information
- It's fast: Can classify tens of thousands of events that will be discovered in LSST and ZTF within a few seconds