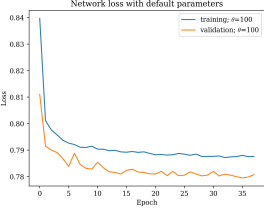
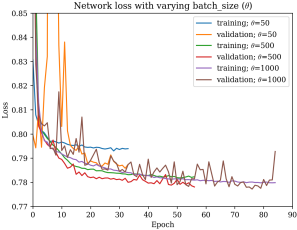
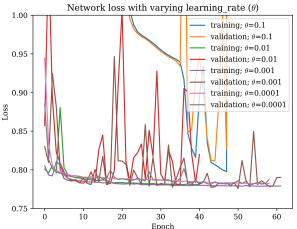
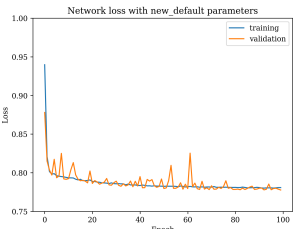
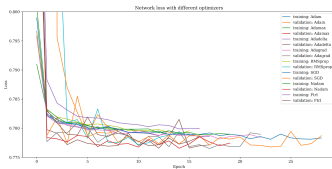
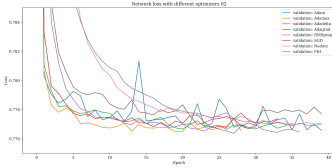
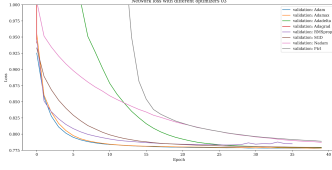
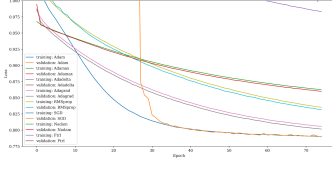
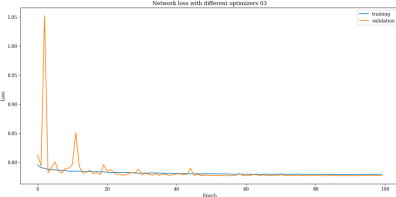
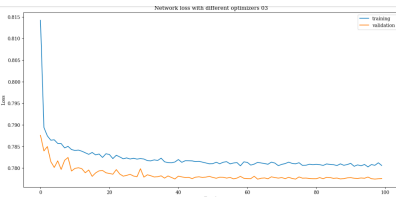
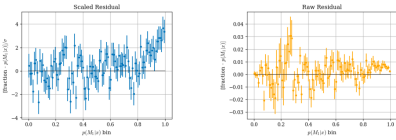
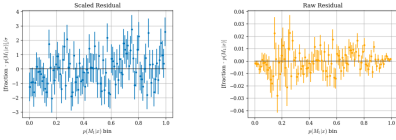


Manual Network Tuning									
Parameters	Values used	Loss	Validation Loss	Early stopping?	Loss plot	Notes	Scaled residual mean	Scaled residual std	
default	default			37					
batch size	[50,100,500,1000]	[0.7939, 0.7887, 0.7823, 0.7798]	[0.7874, 1440098.375, 0.778, 0.7927]	[34, 31, 57, 85]		Batch size 100 diverged. As such, the default has been changed to 500. Also recommended to increase the patience			
learning rate	[0.1, 0.01, 0.001, 0.0001]	[897.8777, 0.7813, 0.7789, 0.7823]	[1.8117, 0.8197, 0.7901, 0.7874]	[53, 41, 62, 59]		Need to really adjust learning rate and batch size at the same time as these two parameters are linked			
new default	I.E, now with patience = 10, L-POP, batch=500	0.7808	0.7777	No		Will lower patience down to 7, 10 won't really do anything. The coverage tests look a lot worse - may be worth incorporating these into your experiments.			

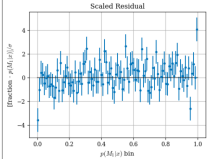
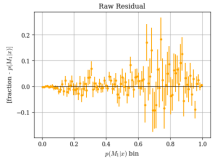
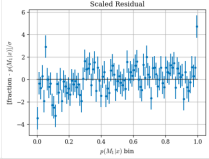
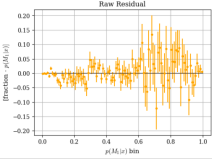
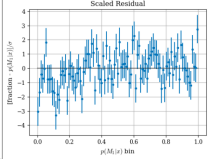
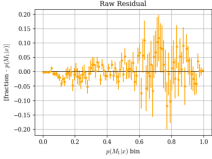
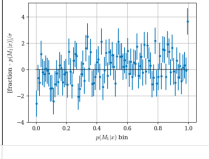
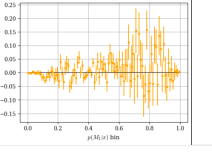
	optimizers, ExpLoss()	['Adam', 'Adamax', 'Adadelta', 'Adagrad', 'RMSprop', 'SGD', 'Nadam', 'Ftrl']	[0.7795, 0.7789, 0.7785, 0.7784, 0.7796, 0.7783, 0.779, 0.78]	[0.7795, 0.7789, 0.7785, 0.7784, 0.7796, 0.7783, 0.779, 0.78]	[9, 14, 18, 23, 16, 29, 20, 17]		Used batch 500, patience 5, width 46, added layers 6. Will try upping the patience as some seemed to have stopped too early.				
	optimizers, ExpLoss()	['Adam', 'Adamax', 'Adadelta', 'Adagrad', 'RMSprop', 'SGD', 'Nadam', 'Ftrl']	[0.7786, 0.779, 0.7784, 0.779, 0.7794, 0.7791, 0.7799, 0.7799]	[0.7766, 0.7765, 0.7771, 0.7774, 0.7765, 0.7777, 0.7771, 0.777]	[40, 29, 36, 34, 37, 40, 40, 40]		Upped the patience from 5 to 9. Allowed val losses to drop marginally.				
	optimizers, POPEXPloss()	['Adam', 'Adamax', 'Adadelta', 'Adagrad', 'RMSprop', 'SGD', 'Nadam', 'Ftrl']	[0.7811, 0.7822, 0.7828, 0.793, 0.785, 0.7834, 0.7918, 0.793]	[0.7782, 0.7784, 0.779, 4800.291, 0.7849, 0.7795, 0.7872, 0.7889]	[40, 40, 40, 10, 36, 40, 40, 40]		Validation losses are marginally higher than that of ExpLoss(). Optimizers generally need more epochs to converge. Try running with more epochs				
	optimizers, POPEXPloss()	['Adam', 'Adamax', 'Adadelta', 'Adagrad', 'RMSprop', 'SGD', 'Nadam', 'Ftrl']	[0.7899, 1.0565, 1.0431, 0.8057, 0.8352, 1.7653, 0.8623, 0.9828]	[0.7896, 905005.1875, 1803869.375, 0.8015, 0.8317, 2702104.75, 0.8599, 1.0032]	[75, 10, 10, 75, 75, 10, 75, 75]		Left for longer number of epochs. Generally showed poor results.				
alpha		[2,1,0.5,1.5,1.25,1.15]	[0.7814, 0.7811, 0.7820,0.7804]	[0.7771,0.7771, 7765,0.7811]	[34, 34, 35,36,39]	In drive	Used batch 150 and best params from test 10 optuna. Patience set to 7.	means = [0.19725, 0.00988, 0.24377,0.21281,0.12763,0.22214]	std = [1.14741, 1.17554,1.28740,1.13950,1.07462,0.99933]		

		alpha=1, lw=45, al=3, lr=1e-3, dr=0.9 (I.E more normal values).	0.7788	0.7773	Let it run to epoch 100, no early stopping		Comparable mean and std to optuna found hyperparameters.	0.18127	0.96711		
		layer_width=45, added_layers=3, learning_rate=1e-4, decay_rate=0.9	0.7806	val_loss: 0.7776	100, no ES		Good values. Again shows can achieve good results with more 'regular' lr and dr	-0.01237	1.07265	What batch size?	
		layer_width=45, added_layers=3, learning_rate=1e-4, decay_rate=0.9: batch 150, alpha=1					Good but not better than PZFlow. Need to reproduce values above.	0.387	1.316		
		layer_width=45, added_layers=3, learning_rate=1e-4, decay_rate=0.9: batch 500, alpha=1					Good! Will also try bs 1000	-0.04386	1.1423		
		layer_width=45, added_layers=3, learning_rate=1e-4, decay_rate=1000, alpha=1					Decent again	-0.2059	1.0095		

		layer_width=45, added_layers=3, learning_rate=1e-4, decay_rate=0.9: batch 500, alpha=2						-0.3604	1.21		
		layer_width=45, added_layers=3, learning_rate=1e-4, decay_rate=0.9: batch 150, alpha=2					Good!	-0.0888	1.1844		
		layer_width=45, added_layers=3, learning_rate=1e-4, decay_rate=0.9: batch 100, alpha=2						poor			

Optuna Network Tuning														
Study number	Parameters	Values	Number of trials	Batch size and learning rate	Layer_width and added_layers	Epochs	Pruning	Early Stopping Patience	Best trial	Best values	Loss	Validation Loss	Notes	Plots
1	Layer width and added layers	5-50, 0-7	100	500, 1e-4	5-50, 0-7	30	Medium pruning	5	76	47, 6	0.7774	0.7747277617	Prefers deeper network - more layers and greater width. Can't really improve on 0.77 in general. Hard to see improvement of added layers from 4+, layer width 30+. Interesting that 74 trials were pruned.	Google drive
2	batch_size and learning rate	50-2500(50), 1e-5, 1e-2 (log)	50	50-2500(50), 1e-5, 1e-2 (log)	16,1	30	Medium pruning	5	32	6.904e-05, 150	0.7871	0.7835963964	Will be interesting to see if similar results are obtained with the layer widths found above. Smaller batch sizes (<100) preferred as expected. Learning rate found to be best around 1e-4	Google drive
3	batch_size and learning rate	50-1000(50), 1e-5, 1e-3 (log)	40	50-1000(50), 1e-5, 1e-3 (log)	40,4	30	Medium pruning	5	4	3.256e-5, 500	0.7799	0.7796990275	With the deeper network, prefers a higher batch size. Note the difference of the validation losses in the range 100-500 is <0.002	Google drive
4	batch_size and learning rate					30	Medium pruning	5	4	0.00006729, 600	0.7783	0.779718101		
5	batch_size, learning rate and decay rate	50-500(50), 1e-5-1e-4(0.2e-5)	50	50-500(50), 1e-5-1e-4(0.2e-5)	40,4	30	Medium pruning	5		500,1e-04,0.5	0.7794	0.7746039033	Compared to the models used above, the lower decay rate finds a marginally lower validation loss	Google drive
6	learning rate, layer width and added layers	1e-5-1e-3, 5-70, 0-9	100	500		50	Medium pruning	8	46	0.00106915259 5, 32, 7		0.7907698154	Held alpha at 2. Looks like needed more epochs to run so rerun with 100 epochs. results now stored as df and saved onto drive	Google drive
7	learning rate, layer width and added layers	1e-5-1e-3, 5-70, 0-9	200	500		100	Medium pruning	8	166	0.00018778670 639054014, 65, 5	0.7802	0.7852050662		
8	learning rate, layer width, decay rate and added layers	1e-5-1e-3, 5-70, 0-9	100	500		100	Medium pruning	8	216	0.00518296640 4054591, 69, 7	0.7798	0.7829719186	Seems you can only add trials to continue tests if parameters stay the same. Decay rate did not change here	Lost
08 (in drive)	learning rate, layer width, decay rate and added layers	1e-5-1e-3, 5-70, 0.5-1, 0-9	100	500		100	Medium pruning	8	3	0.00155084202 40121252, 0.59290721956 80534, 45, 1	0.7785	0.7752603292	Good. Will add on 100 trials to study 2 to continue	Google drive
9	learning rate, layer width, decay rate and added layers	1e-5-1e-3, 5-70, 0.5-1, 0-10	100	500		100	Medium pruning	8 same		same	same	same	Continuation of above. Trial 3 remains the best.	
10	learning rate, layer width, decay rate and added layers	1e-5-1e-3, 5-70, 0.5-1, 0-11	100	150		100	Medium pruning	8		45, 3, 'learning_rate': 0.00121557217 98680723, 'decay_rate': 0.66764945162 13406	0.7811	0.7743891478	Tested above but with batch of 150. Validation loss has been made marginally lower again, although might be overfitted.	

	<b>PZFlow Tuning</b>								
	Parameters	Values used	M1 Losses	M0 Losses	Epochs	Plots	Notes	Mean	Std
	Batch size	500	-4.2919 -4.2960	-4.0426 -4.0457	50		Very bad	6.2687	2.4729
	Batch size	100	-4.2931 -4.2966	-4.0430 -4.0458				6.2503	2.7401
	Data size	Used all data vectors, batch 100							
	Default values		-4.2930 -4.2929	-4.0487 -4.0515	100	Drive	Decent. M0 training loss divereged so ended after epoch 68	-0.2611	1.296

Optuna Network Tuning on Photometry data															
Study number	Parameters	Values	Number of trials	Batch size and learning rate	Layer_width and added_layers	Epochs	Pruning	Early Stopping Patience	Best trial	Best values	Losses	Metric	Mean+Std	Notes	Plots
11	lr,dr,al,lw	1e-4, 1e-3(1e-4); 0.9, 1 (0.01); 0.9; 5, 70	50x2	500			medium pruning (based on validation)	10	14	55.5, 4e-4, 0.9	0.2916, 0.2988	SSR: 853794.1775	0.5972, 1.2821	Residuals lie above 0.5. Some valued 4-6 sigma	
12	lr,dr,al,lw	1e-4, 1e-3(1e-4); 0.9, 1 (0.01); 0.9; 5, 70		Repeat of 50 above with 150					0*	lw:9, al:1, 3e-4, 0.92	0.3097, 0.3149	SSR: 198.061719389 694	-0.1637, 1.7723	All trials found the same value - something strange here. Overall most values look better but has more outliers and higher variance.	Drive
13	lr,dr,al,lw	1e-4, 1e-3(1e-4); 0.9, 1 (0.01); 0.9; 5, 70	50	150				10		layer_width: 34, 'added_layers': 0, 'learning_rate': 0.0004, 'decay_rate': 0 1.0]		AM+SD: 93.5017	1.9855, 2.6440	Only got through 10 trials or so. Very poor outcome. The metric from the objective function does not match that outputted later	
14	lr,dr,al,lw	1e-4, 1e-3(1e-4); 0.9, 1 (0.01); 0.9; 5, 70	50 (got to 13 before disconnect)	500		50		10	0			MSE(on SR): 4775.3971	-1.6591, 1.3193	MSE seems to like to make the residuals negative.	Drive
	lr,dr,al,lw	1e-4, 1e-3(1e-4); 0.9, 1 (0.01); 0.9; 5, 70		150		50	/		0		/		-1.089, 1.2636	Took test above and tried 150 instead of 500. Not awful though still a few at 4 sigma. Perhaps keep running and see what appears?	Drive
15	lr,dr,al,lw	1e-4, 1e-3(1e-4); 0.9, 1 (0.01); 0.9; 5, 70	50	500		30 (reduced to get through more trials)		10				MSE(on R):		Ended early as metric is giving repeat values	
/	best values from unobservables	45, 3, 1e-4, 0.9	/	500		50							-0.2005, 1.335	Repeat with 100 epochs, then try smaller batch size	
/	best values from unobservables	45, 3, 1e-4, 0.9	/	500		100							0.16433, 1.1115	Good but a couple of values that are 4 sigma. Next try 150 batch size	 
/	best values from unobservables	45, 3, 1e-4, 0.9	/	150		100							-0.0139, 1.2237	Still high variance.	 
	Repeat of above but alpha=2	45, 3, 1e-4, 0.9		150		100							-0.06932, 1.14716	The high sigma 4 values are gone which is good - all just about within 3	 
	Alpha=2, bs of 500	45, 3, 1e-4, 0.9		500									0.08035, 1.1944	Not better than above, some 4 sigma values.	//
	Alpha=2, bs of 100	45, 3, 1e-4, 0.9		100									0.15887, 1.11877	Nice! Highest sigma around 3.8. Go lower?	 
	Alpha=2, bs of 50			50									overfitting		