SELDON: Supernova Explosions Learned by Deep ODE Networks

Jiezhong Wu[1,2*], **Jack O'Brien**[1,3*‡], **Jennifer Li**[1,3,4], **M. S. Krafczyk**[1,4], **Ved G. Shah**[1,5,6], **Amanda R. Wasserman**[1,3,4], **Daniel W. Apley**[1,2], **Gautham Narayan**[1,3,4†‡], **Noelle I. Samia**[1,7†‡]

[1] NSF-Simons AI Institute for the Sky (SkAI), Chicago, IL, USA
[2] Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, USA
[3] Department of Astronomy, University of Illinois Urbana-Champaign, Urbana, IL, USA
[4] National Center for Supercomputing Applications (NCSA), Urbana, IL, USA
[5] Department of Physics and Astronomy, Northwestern University, Evanston, IL, USA
[6] Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA), Northwestern University, Evanston, IL, USA
[7] Department of Statistics and Data Science, Northwestern University, Evanston, IL, USA

jiezhongwu2021@u.northwestern.edu, jackob@illinois.edu, jli184@illinois.edu, mkrafcz2@illinois.edu, vedshah2029@u.northwestern.edu, amandaw8@illinois.edu, apley@northwestern.edu, gsn@illinois.edu, n-samia@northwestern.edu

Abstract

The discovery rate of optical transients will explode to 10 million public alerts per night once the Vera C. Rubin Observatory's Legacy Survey of Space and Time comes online, overwhelming the traditional physics-based inference pipelines. A continuous-time forecasting AI model is of interest because it can deliver millisecond-scale inference for thousands of objects per day, whereas legacy MCMC codes need hours per object. In this paper, we propose SELDON, a new continuous-time variational autoencoder for panels of sparse and irregularly time-sampled (gappy) astrophysical light curves that are nonstationary, heteroscedastic, and inherently dependent. SELDON combines a masked GRU-ODE encoder with a latent neural ODE propagator and an interpretable Gaussian-basis decoder. The encoder learns to summarize panels of imbalanced and correlated data even when only a handful of points are observed. The neural ODE then integrates this hidden state forward in continuous time, extrapolating to future unseen epochs. This extrapolated time series is further encoded by deep sets to a latent distribution that is decoded to a weighted sum of Gaussian basis functions, the parameters of which are physically meaningful. Such parameters (e.g., rise time, decay rate, peak flux) directly drive downstream prioritization of spectroscopic follow-up for astrophysical surveys. Beyond astronomy, the architecture of SELDON offers a generic recipe for interpretable and continuous-time sequence modeling in any time domain where data are multivariate, sparse, heteroscedastic, and irregularly spaced.

SELDON code — https://github.com/skai-institute/seldon

[*] Jiezhong Wu and Jack O'Brien contributed equally.
[†] Noelle I. Samia and Gautham Narayan contributed equally.
[‡] Corresponding authors: Noelle I. Samia, Gautham Narayan, Jack O'Brien

Introduction

The arrival of the Rubin Observatory's Legacy Survey of Space and Time (LSST) will transform time-domain astronomy into a data-deluge era as it is expected to issue $\sim$10 million public alerts per night (Ivezić et al. 2019), overwhelming the capacity of traditional physics-driven analysis codes that rely on hours-long Markov-chain Monte Carlo (MCMC) runs per source. To support scientific discovery and maximize the return on limited spectroscopic resources, a new continuous-time forecasting AI model that can deliver millisecond-scale inference for hundreds of objects per day and extrapolate light curves from early, partial observations is urgently needed.

Modeling astronomical time-domain light curves presents fundamental challenges. These nonstationary time series are sparse (often containing few measurements per object), heteroscedastic (exhibiting varying uncertainties across epochs), limited (covering only part of the event evolution), and irregularly spaced in time (Bianco et al. 2022). Classic ARMA/ARIMA families are designed for evenly spaced and inherently stationary time series, and treating Rubin's highly irregular cadence as missing data degrades both statistical power and interoperability. Also, their i.i.d. Gaussian-error assumption ignores the band-dependent photometric uncertainties, violating homoscedasticity assumptions and biasing parameter estimates (Feigelson, Babu, and Caceres 2018). Even when continuous-time generalizations (e.g., CARMA/CARFIMA) are used, inference scales cubically in the number of points, so a single supernova light curve can still take minutes to hours – orders of magnitude slower than the millisecond budget required for $\sim$10 million alerts-per-night (Kelly et al. 2014). There has been a growing body of work applying deep learning to transient light curves, but most existing models target either classification or coarse parameter regression rather than full multi-

band flux forecasting. For example, SUPERNNOVA employs recurrent neural networks to assign supernova subtypes from partial photometry (Möller and de Boissière 2020), while RAPID leverages a recurrent neural network architecture to deliver near-real-time supernova type and rough peak-epoch estimates (Muthukrishna et al. 2019). PELICAN augments sparse LSST-like sequences with domain adaptation to improve classification under distribution shift (Pasquet et al. 2019), and ORACLE extends this line to a hierarchical, broker-scale classifier aimed at Rubin alert streams (Shah et al. 2025).

Autoencoders offer a powerful framework for learning compact task-agnostic representations for complex data, and the autoencoder families that operate on a fixed temporal grid have long been the default solution for sequence modeling. Early examples such as STORN (Bayer and Osendorfer 2014), VRNN (Chung et al. 2015), SRNN (Fraccaro et al. 2016), and the Deep Kalman Filter (Krishnan, Shalit, and Sontag 2017) combine a recurrent (or state–space) encoder with a generative decoder, delivering strong likelihoods on densely sampled videos and speech. More sophisticated variants embed a Gaussian-process prior in latent space (Fortuin et al. 2019) or add score-based time extrapolation (Tóth et al. 2020), but all of these methods assume an equispaced input grid. When confronted with gappy, irregularly-spaced, and highly heteroscedastic Rubin-like light curves, they fail to make accurate predictions on critical quantities such as peak time and flux.

Continuous-time latent models offer a principled route around the fixed-grid limitation. The seminal Neural ODE of Chen et al. (2018) replaces the discrete RNN update with a differential flow $\dot{z} = f_\theta(z, t)$, allowing predictions at arbitrary time stamps. From the encoder side, variants such as ODE–RNN (Rubanova, Chen, and Duvenaud 2019) and GRU-ODE-Bayes (Li et al. 2020) integrate this flow only between arrivals and therefore cope well with sparsity, but they decode with an unconstrained multilayer perceptron. In contrast, Latent ODE (Chen et al. 2018) and ODE2VAE (Yildiz et al. 2019) generate directly in continuous time, yet still lack band-specific, physically interpretable outputs. We combine the strength of previous work and provide SELDON, a novel architecture that marries a gap-aware encoder with a continuous-time latent propagator, condenses the resulting trajectory into a fixed-length summary, and channels that summary through an analytic decoder whose parameters are directly interpretable. More specifically, our architecture has a masked GRU-ODE encoder that processes the sparse, irregularly-spaced, and heteroscedastic time series. At each observation, it performs a GRU update, then continuously propagates the hidden state between observations with a neural ODE, advancing the hidden state to the time of the next observation in the time series. The encoder's final latent vector serves as the initial condition for a downstream hidden ODE. Then, a continuous-time flow $\dot{z} = f_\theta(z, t)$ evolves an initial latent vector forward to a static regularly sampled grid, producing a dense trajectory $\{h(t_j)\}$. The trajectory is passed through a network ϕ and aggregated by a permutation-invariant pooling ρ, leading to a fixed-length representation $z = \rho(\{\phi(h(t_j))\})$. A RESNET takes z, together with a learnable embedding and returns for each band amplitudes, centers, and (inverse) widths of K Gaussian basis functions. These Gaussian bases can deliver both pointwise predictions and inference about physically interpretable light curve attributes such as rise time, decay rate, peak time and flux that downstream schedulers use to prioritize scarce spectroscopic resources for followup in real time.

Methods

Data

Our work is driven by a new, publicly released data source, the `ELAsTiCC` astronomical multivariate time-series dataset (Narayan and ELAsTiCC Team 2023), hosted in `Astro-ORACLE` (Shah et al. 2025). The library realistically simulated transients into a single HDF5 file with the canonical schema {`MJD`, `band`, `FLUXCAL`, `FLUXCALERR`}, representing the time of observation, the photometric filter with which the observation was taken, the calibrated flux in a given filter band, and the associated error with that flux measurement respectively. Focusing on the Type Ia supernova (SN Ia) class of transient events, every event is a multi-band $(\mathrm{u}, \mathrm{g}, \mathrm{r}, \mathrm{i}, \mathrm{z}, \mathrm{y})$ photometric time series. For an overview of supernovae physics and evolution, see Alsabti and Murdin (2017).

We represent these data as panels of dependent time-domain light curves LC_j, such that each light curve contains a total of N_j flux observations denoted by f_{j,t_i,b_i} where b_i is the photometric filter band at time point t_i, where i indicates the i^{th} observation in the light curve.

Due to the fact that filters are changed between observations, we can only observe the flux in a single filter band, at any given time for a single supernova. The data are irregularly spaced (gappy), sparse with few observations, and limited in that they often cover only part of the evolution of the light curve. Figure 2 is an illustration of a light curve observed over time across six bands, where the total number of observations is in the 99^{th} percentile of all light curves. Typically, the total number of observations per light curves across all bands have a mean of 18 and a median of 14. Each point in a light curve has an observed flux error indicated as an error bar in Figure 2.

LSST is more sensitive in some filter bands (e.g., bands g and r) than others (e.g., u and y) (Olivier, Seppala, and Gilmore 2008), resulting in an imbalance of the data – biased towards the more sensitive bands – and varying signal-to-noise ratio between bands. In addition, data corresponding to each light curve are correlated within and between bands (Filippenko 1997).

These irregularly-sampled, inherently-dependent light curves are nonstationary, heteroscedastic, and nonlinear (Figure 2), thus posing substantial challenges in performing real-time prediction of light curves per band, in addition to recovering information across bands.

Preprocessing For each light curve, we set the temporal origin of every light curve as follows. If an observed flux is below the acceptable signal-to-noise threshold for the survey, this observation is labeled as a non-detection point. Some non-detection points may have a negative flux

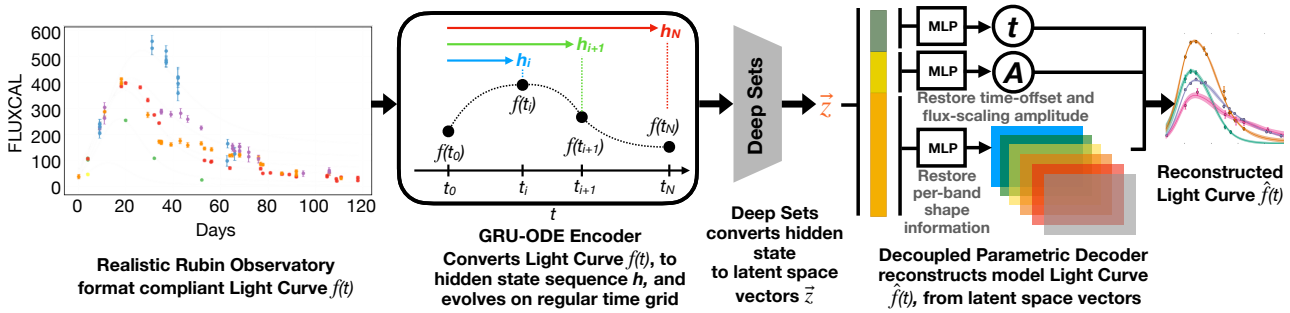

Figure 1: Architecture of our proposed SELDON, a customized VAE with band-aware GRU-ODE encoder and interpretable Gaussian-basis decoder. A light curve described by a series of flux observations in various filter bands is encoded to an initial hidden state with the GRU-ODE. The hidden state is evolved with the neural ODE forward in time to form a trajectory on a regularly-sampled grid. This trajectory is then interpreted by a Deep Sets layer to an approximate posterior latent vector. The latent vector is then decoded into a series of basis function parameters representing the history and future evolution of the light curve at all times in all filter bands.

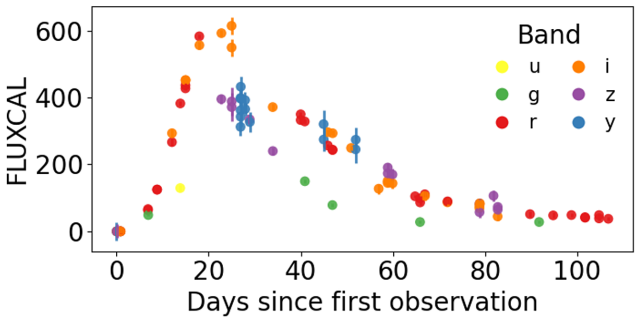

Figure 2: An illustration of a light curve observed over time across six bands indicated in distinct colors, where the total number of observations is in the 99$^{\text{th}}$ percentile of all light curves. The error bars for each observation represent the observed flux errors.

due to systematic errors from background subtraction, and hence are noise dominated. The flux of every non-detection point is then fixed at zero and its observed flux error remains unchanged. For each light curve, all non-detection points are omitted from our data except for up to 8 non-detection points immediately preceding the first detection point. This is needed to gain information about the point in time at which the light curve begins to rise. The upper limit of 8 non-detection points is set at the median number of such points across all light curves. We take the first of those non-detections as t_0. If there are no non-detections prior to the detection point, we set t_0 to be the time of the first detection point which coincides with the discovery time at which we observe the first detection point. Then, we rescale time by $t_{\text{norm}} = 2\sigma$, where σ is the standard deviation of full-curve durations over the training set. The resulting $\tilde{t}_i = (t_i - t_0)/t_{\text{norm}}$ keeps gradient magnitudes well scaled. From the training data, we set t_{norm} to be 71.9 days.

Raw flux measurements span several orders of magnitude and carry heteroscedastic errors. To bring all measurements onto a numerically stable scale for training purposes, we apply a signed logarithmic compression followed by symmetric scaling. Specifically, we use the log-modulus transformation

$$g(f) = \operatorname{sgn}(f)\,\log_{10}\big(|f|+1\big), \qquad \sigma_g = \frac{\sigma_f}{(|f|+1)\,\ln 10}. \tag{1}$$

This transformation makes the flux distribution approximately symmetric and keeps magnitudes $\mathcal{O}(1)$ for stable optimization. We then scale (g, σ_g) by the dataset-wide maximum absolute value $g_{\max} = \max_{\text{train}} |g|$ and recenter

$$\tilde{g} = \frac{g}{g_{\max}} - 0.5. \tag{2}$$

In practice this maps $\tilde{g}$ close to $[-0.5, 0.5]$ while preserving relative magnitudes. The inverse transform uses $g = (\tilde{g} + 0.5)g_{\max}$ followed by $f = \operatorname{sgn}(g)\,(10^{|g|} - 1)$.

Augmentation At each training step, the model is trained on a partial light curve, which is a freshly generated, incomplete version of full light curve. More precisely, we keep only the first K measurements, where K is drawn uniformly between a minimum of 10 points and the full length of the light curve. Because K is almost always smaller than the phase of maximum flux, these cut-offs mainly contain the rising part of the light curve, which is exactly the scenario faced by survey schedulers who must decide follow-up strategy before the peak. By generating these partial curves on the fly, the model encounters a new mix of truncated curves in every mini-batch and learns to make reliable predictions from whatever segment of the light curve it happens to receive.

Architecture

We develop a customized variational autoencoder (VAE) in which a band-aware GRU-ODE encoder with deep sets maps each partial light curve to a latent Gaussian $\mathcal{N}(\boldsymbol{\mu}, \operatorname{diag}\boldsymbol{\sigma}^2)$ of dimension D_z (we use $D_z = 64$), and a parametric basis decoder reconstructs flux values on any provided continuous time grid from a linear combination of

Gaussian basis functions. Figure 1 provides an overview of our developed architecture pipeline. Below we describe the various components of our proposed VAE in detail. Hyperparameter values for the models described below are shown in Table 1.

Embedding Every measurement is represented as a fixed-width six-channel vector $[\tilde{t}_i,\, \tilde{g}_i,\, \mathbf{e}_i^\top]^\top$, where $\tilde{t}_i$ is the normalized time, $\tilde{g}_i$ is the log-scaled flux after rescaling, and $\mathbf{e}_i \in \mathbb{R}^4$ is a learnable band embedding. The band information is essential for capturing the color evolution of the supernovae. Each photometric band $\mathrm{u, g, r, i, z, y}$ is assigned an integer index that retrieves $\mathbf{e}_i$ from an embedding matrix $\mathbf{E} \in \mathbb{R}^{6\times 4}$ initialized randomly and learned jointly with the network. To mitigate band-frequency imbalance, gradients of each embedding vector are scaled by the inverse occurrence frequency of its band by dividing the gradients in each embedding by the number of corresponding samples in each mini-batch. Thus every observation is encoded as

$$\mathbf{z}_i = [\tilde{t}_i,\, \tilde{g}_i,\, \mathbf{e}_i^\top]^\top \in \mathbb{R}^6,$$

providing a compact representation that combines time, flux, and band information for input to the encoder.

Encoder The encoder takes a sparse irregular light curve and returns a Gaussian posterior over an initial latent state z_0. We implement three competing variants: a traditional GRU encoder (Cho et al. 2014), a permutation-invariant Deep Sets network model (Zaheer et al. 2017), and a GRU-ODE network model (Rubanova, Chen, and Duvenaud 2019) with Deep Sets.

Gated Recurrent Unit The GRU is a lightweight gated-RNN architecture proposed to mitigate the vanishing or exploding gradient problems that plague vanilla RNNs. We consider a single-layer GRU with `hidden_dim` hidden units that processes the six-dimensional input $[\tilde{t}_i, \tilde{g}_i, \mathbf{e}_i^\top]^\top$ and outputs a `hidden_dim`-dimensional hidden state that memorizes the entire light curve for the downstream decoder.

Deep Sets Deep Sets provides a simple way to encode a variable-length order-agnostic collection of items. To implement it, we use an element-wise network ϕ that maps each observation to a `hidden_dim`-dimensional space, followed by a sum pooling, $\sum_t \phi(\mathbf{x}_t)$, and a second MLP ρ outputting $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}^2$. This encoder is permutation invariant and preserves the association between flux observations and time as they are initially passed as pairs to this encoder.

GRU-ODE Encoder with Deep Sets A single-layer, unidirectional GRU with `hidden_dim` hidden units takes five-dimensional inputs $[\tilde{g}_i,\, \mathbf{e}_i^\top]^\top$ in reverse chronological order, omitting time since temporal evolution is handled explicitly by the continuous ODE between observations. Between observation times, the hidden state $h(\tilde{t})$ is propagated by an autonomous latent ODE

$$\frac{\mathrm{d}h_{\tilde{t}}}{\mathrm{d}\tilde{t}} = f_\theta(h_{\tilde{t}}), \tag{3}$$

and updated by the GRU at each measurement, ensuring smooth trajectories for irregularly spaced light curves. The final hidden state serves as the initial condition for a forward ODE, which evolves this state on a regularly sampled time grid of 50 points for $\tilde{t} \in [0, 1]$, producing a `hidden_dim`$\times 50$ hidden-state trajectory. This corresponds to a $\approx$72 day evolution which adequately encompasses the evolution of SN Ia optical light curves. This trajectory is concatenated with the corresponding time grid and encoded by a `Deep Sets` module with sum aggregation to yield the approximate posterior

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}\, \boldsymbol{\sigma}^2).$$

Latent Neural-ODE Solver We integrate the latent flow $\dot{z} = f_\theta(z)$ with the adaptive TSIT5 solver from TORCHODE (Lienen and Günnemann 2023) with a maximum step size dt of 0.01 in normalized time space. The entire solver is wrapped in an ODESOLVER module that is `torch.compile`-optimized, giving a $\sim 2\times$ speed-up over the default eager-mode PyTorch execution. At run time, we can switch between the *AutoDiffAdjoint* path, which stores the forward trajectory and is fastest when memory allows, and the memory-lean *BacksolveAdjoint* path, which recomputes states during the backward sweep. Both adjoint choices deliver gradients accurate up to solver tolerances and discretization error, and the solver naturally parallelizes over a batch of latent trajectories, allowing us to propagate hundreds of light curves per second on a single GPU.

Decoder During training we use the re-parameterization trick, drawing $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ (Kingma and Welling 2014). At test time, we set $\mathbf{z} = \boldsymbol{\mu}$ to obtain a single deterministic reconstruction. We use a parametric Gaussian basis decoder, for which the flux in band b is modeled as a sum of K Gaussian basis functions

$$\hat{f}_b(t) = \sum_{k=1}^{K} w_{bk} \exp\left[-\left((t - \mu_{bk})\sigma_{bk}\right)^2\right]. \tag{4}$$

In (4), for each band b and basis component k, the decoder predicts three parameters: the amplitude $w_{b,k}$, the center time $\mu_{b,k}$, and the rate $\sigma_{b,k}$ that controls its spread. Together, these parameters let the sum of K Gaussians flexibly trace the rise and fall of the light curve in band b.

The Gaussian basis decoder takes $[\mathbf{z}]$, passing it through a four-layer RESNET with hidden size `hidden_dim`, and outputs the parameters of a band–specific analytic light-curve model that can be evaluated on any query grid $\{t_j\}$. We multiply the latent vector of basis function parameters element-wise with the band embedding and sum them to a set of $K = 8$ basis parameters for each band, reducing the input dimensionality without halting gradient flow. The MLP weights are shared across bands, while band-to-band variation enters through another learned embedding $\mathbf{e}_{b,decoder}$ with 16 output dimensions.

Amplitudes and centroids are mapped through a learnable exponential or normal inverse transform mapping, respectively. These enforce the parameter distributions to be well behaved during early training epochs. By initializing the learnable parameters of these mappings, we tune the network to make initial predictions close to reasonable values for these parameters. We scale the amplitudes by dividing

each amplitude by the mean of the set of K amplitudes. We center the centroids by subtracting the mean of the set of K centroids from each centroid.

A global amplitude and centroid are then decoded from a decoupled segment of the latent vector, each from their own independent RESNET, and projected onto their own respective inverse transform sampler. The individual basis amplitudes are then multiplied by the global amplitude, and the global centroid is added to the individual centroids. Of the 64 components of the latent vector, the individual basis parameters are decoded from the first 48 components. The global centroid is decoded from the next 8 components of the latent vector, and the global amplitude is decoded from the final 8 components of the latent vector. The independence of the global parameters allows for a decoupling between the overall amplitude of the light curve and its time offset, providing scale and time invariance to the individual components.

Learnable Inverse Transform Mapping We use a learnable distribution mapping for certain basis parameters in order to improve training performance by enforcing a restricted initial distribution for early epochs which is then relaxed during training. Particularly, an inverse transform is used to learn the parameters μ and σ of a normal distribution applied to the individual and global centroids with initial respective values of 0.0 and 2.0 for μ and an initial value of 0.1 for σ. For the case of individual and global amplitudes, an exponential distribution is applied with the initial values for λ set to 0.25 and 8.0, respectively.

Loss For each light curve LC_j observed at time t_i, let $\hat{f}_{j,t_i,b_i}$ be the reconstructed flux, f_{j,t_i,b_i} the observed flux, and σ_{j,t_i,b_i} the reported flux error. A binary mask $m_i \in \{0,1\}$ identifies valid (unpadded) samples at the i^{th} observation in the light curve. With numerical safeguard $\varepsilon = 10^{-6}$ and Huber scale $\delta = 1$, we apply Huber loss ℓ_{j,t_i,b_i} to the standardized residual $r_{j,t_i,b_i} = \frac{f_{j,t_i,b_i} - \hat{f}_{j,t_i,b_i}}{\max(\sigma_{j,t_i,b_i}, \varepsilon)}$.

Masking out padded entries and averaging over the valid points of each sequence yields the per-curve loss

$$\mathcal{L}_{\text{rec},j} = \frac{\sum_i m_{j,i}\, \ell_{j,t_i,b_i}}{\sum_i m_{j,i} + \varepsilon}. \tag{5}$$

To regularize the latent space, we add the Kullback–Leibler divergence between the approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}, \operatorname{diag} \boldsymbol{\sigma}^2)$ and the unit Gaussian prior $p(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ given by

$$D_{\text{KL}}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\big) = \tfrac{1}{2} \sum_{d=1}^{D_z} \big(\mu_d^2 + \sigma_d^2 - \log \sigma_d^2 - 1\big), \tag{6}$$

where D_z is the latent dimensionality (set to 64 in our experiments). The total per-curve objective is therefore

$$\mathcal{L}_j = \mathcal{L}_{\text{rec},j} + \beta\, D_{\text{KL}}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})\big), \tag{7}$$

with $\beta = 10^{-4}$ used as a fixed constant throughout training. The scalar training objective is the mean over a batch of size B, given by

$$\mathcal{L} = \frac{1}{B} \sum_{j=1}^{B} \mathcal{L}_j. \tag{8}$$

This formulation combines a robust, uncertainty-aware reconstruction term with a variational regularizer, yielding stable training and a well-behaved latent prior.

Model	Masked-GRU	Deep Sets	SELDON
`hidden_dim`	256	256	128
Learning Rate	0.0001	0.002	0.002
Batch Size	2048	2048	2048

Table 1: Hyperparameters for the three encoder models compared within this work. Hyperparameters were optimized through grid search.

Training SELDON was trained for 180 epochs on a Nvidia H100 GPU using the `Adam` optimizer with β_1 = 0.9, β_2 = 0.999, and $\epsilon = 10^{-8}$. A single training step took 7.5 seconds with 1.1 seconds for inference. Training used batch accumulation over 4 mini-batches of size 512 and required 45GB of GPU memory. The model was trained until the validation loss stopped improving.

Results

Performance metrics We adopt the following three metrics to evaluate the model forecasting performance of a panel of inherently-dependent time-domain light curves LC_j, such that each light curve contains a total of N_j flux predictions given by $\hat{f}_{j,t_i,b_i}$, with observed flux f_{j,t_i,b_i} and observed flux error σ_{j,t_i,b_i}. For simplicity, we denote the i^{th} flux observation of LC_j by $f_{j,i}$ with observed flux error $\sigma_{j,i}$ and the associated predicted flux as $\hat{f}_{j,i}$. While we are using simpler notations for this section, it is important to note that the i^{th} flux observation of a light curve LC_j implicitly refers to a filter band b_i and time point t_i.

Define the standardized Z-score for the i^{th} flux observation of light curve LC_j as follows

$$|Z_{j,i}| := \frac{\left| f_{j,i} - \hat{f}_{j,i} \right|}{\sigma_{j,i}}. \tag{9}$$

Then, the mean absolute Z-score and the maximum absolute Z-score are defined as

$$\overline{|Z|}_j := \frac{1}{N_j} \sum_{i=1}^{N_j} |Z_{j,i}|\,, \quad \operatorname{Max} |Z|_j := \max_{1 \le i \le N_j} |Z_{j,i}|\,. \tag{10}$$

Assuming the test set has M curves, then the overall metrics are defined as

$$\text{Mean}|Z| := \frac{1}{M} \sum_{j=1}^{M} \overline{|Z|}_j, \quad \operatorname{Max} |Z| := \max_{1 \le j \le M} \operatorname{Max} |Z|_j, \tag{11}$$

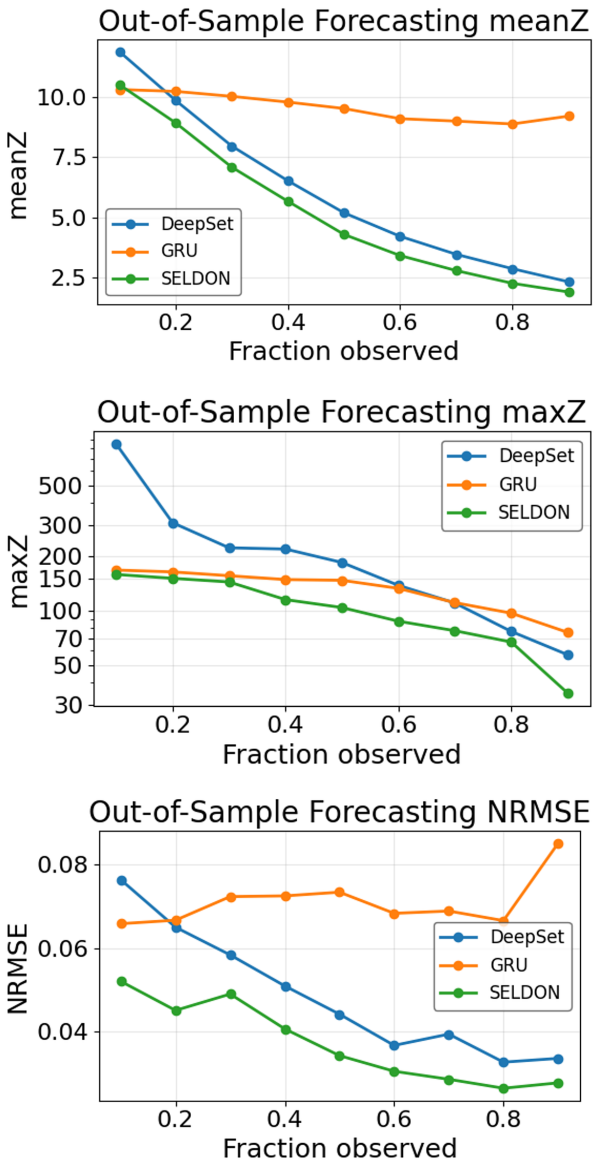

Figure 3: Out-of-sample forecasting performance as a function of the fraction of the light curve that has been observed. **Top**: mean absolute Z-score ($\mathrm{mean}\,|Z|$). **Middle**: worst-case absolute Z-score ($\max|Z|$, log-scale). **Bottom**: normalised RMSE. Lower is better in all panels. SELDON (i.e., GRU-ODE) (green) consistently produces the lowest tail and aggregate errors. A plain masked-GRU (orange) has the best median at 10% observed but is outperformed by SELDON afterward. Deep Sets (blue) shows competitive medians, yet the heaviest tails.

The normalized root mean-square error NRMSE is defined as

$$\mathrm{NRMSE} := \frac{1}{M}\sum_{j=1}^{M}\frac{\sqrt{\frac{1}{N_j}\sum_{i=1}^{N_j}(f_{j,i}-\hat{f}_{j,i})^2}}{\max_i f_{j,i}}, \qquad (12)$$

which is the average of the root-mean-square reconstruction errors normalized by the curves' maximum observed fluxes, yielding a dimensionless quantity.

Forecasting Performance

We evaluate the performance of our proposed SELDON against the two other models with the masked-GRU encoder and the Deep Sets encoder, on out-of-sample predictions cutoff at a given percentage of sequential data points that are assumed to be observed. Table 2 reports two sets of aggregate metrics divided into two categories, absolute and relative metrics. The results show that SELDON with Deep Sets model achieves the best performance of the three we have tested across all columns of Table 2 (except at 10% observed fraction where the Mean $|Z|$ of GRU is slightly better) and with increasing performance over other models at earlier times in the sequence. This is also illustrated in Figure 3, which displays the out-of-sample forecasting performance as a function of the percentage observed which is consistently favoring our SELDON model against the masked-GRU model and the Deep Sets model.

Figure 4 reports the violin plots of the signed standardized residuals for out-of-sample forecasts per percentage observed, in each of the three models. The violin plots demonstrate that the masked-GRU encoder has consistently poor performance regardless of the percentage of the light curve revealed. While the Deep Sets model has poor performance for low percentages, it does improve as more data is revealed though not as quickly as SELDON. For our applications, it is most crucial to have accurate real-time predictions at the early stages of the time series before the peaks are observed, a task that is best achieved by our SELDON and consistently across all the metrics used.

When only 10% of the light curve is revealed, the task is essentially extrapolation from a handful of early points. In that regime the masked-GRU model attains the smallest mean $|\mathrm{Z}|$ (i.e., 10.3σ), edging SELDON by $< 2\%$. The difference disappears once $\geq$ 20% of the curve is available: SELDON takes the lead and keeps it through 90% coverage, cutting the average error by 10-35% relative to GRU and by 15-45% relative to Deep Sets. The gain grows with the fraction because the latent-ODE decoder can exploit every additional observation to tighten its continuous trajectory, whereas the discrete-time masked-GRU is limited to fixed-step updates.

The Max$|Z|$ highlights a contrast. Deep Sets occasionally produces catastrophic residuals, hundreds of σ at 20% coverage and nearly 900σ at 10%. A vanilla masked-GRU removes the worst spikes but still exhibits excursions above 160σ. SELDON caps the worst error below 160σ in the sparsest slice and below 90σ thereafter, delivering the tightest upper tail throughout.

Every standardized residual in the NRMSE is weighted quadratically, and therefore it penalizes both bias and heavy tails. SELDON achieves the lowest NRMSE at every percentage of observed values, $20-35\%$ lower than masked-GRU and $30-50\%$ lower than Deep Sets, showing that it not only reduces the mean error but also suppresses large deviations effectively.

	Mean$\|Z\| \downarrow$			Max$\|Z\| \downarrow$			NRMSE $\downarrow$		
Fraction	Deep Sets	Masked-GRU	SELDON	Deep Sets	Masked-GRU	SELDON	Deep Sets	Masked-GRU	SELDON
0.1	11.885	**10.315**	10.513	848.772	168.496	**159.292**	0.076	0.066	**0.052**
0.2	9.862	10.237	**8.929**	309.186	164.536	**151.551**	0.065	0.067	**0.045**
0.3	7.966	10.031	**7.089**	224.164	156.688	**144.667**	0.058	0.072	**0.049**
0.4	6.525	9.792	**5.673**	220.531	149.150	**115.359**	0.051	0.072	**0.041**
0.5	5.193	9.526	**4.295**	185.677	147.793	**104.237**	0.044	0.073	**0.034**
0.6	4.218	9.103	**3.418**	138.518	133.256	**87.474**	0.037	0.068	**0.031**
0.7	3.469	9.002	**2.794**	110.469	110.968	**77.439**	0.039	0.069	**0.029**
0.8	2.872	8.884	**2.266**	76.708	96.979	**66.972**	0.033	0.067	**0.027**
0.9	2.329	9.209	**1.906**	56.891	75.786	**34.789**	0.034	0.085	**0.028**

Table 2: Out–of–sample performance per fraction of observed values. Down arrows next to the metrics indicate lower values are better. **Bold** values indicate best overall performance across all models.

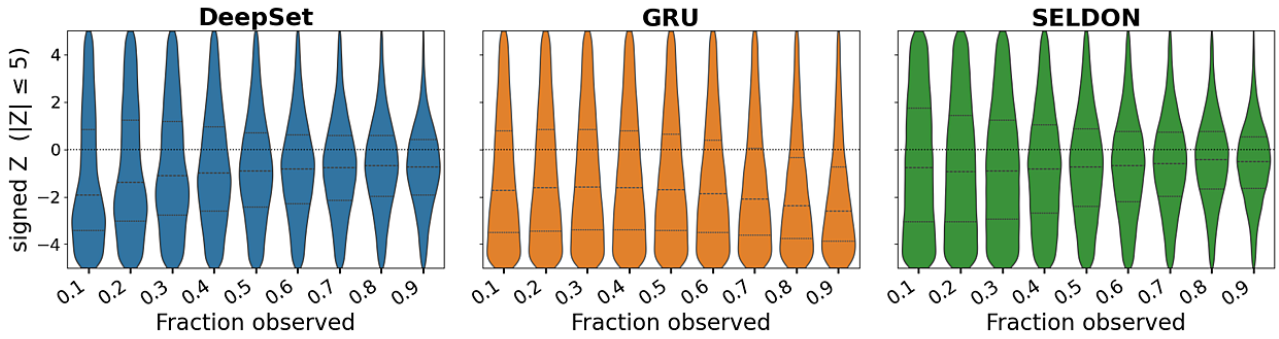

Figure 4: Violin plots of the signed standardized residuals for out-of-sample forecasts per fraction observed, in each of the three models. These residuals are clipped between ± 5 for visual clarity.

Discussion

We developed a forecasting VAE model for SNe Ia optical light curve evolution based on a combination of GRU-ODE encoder and Deep Sets with a Gaussian basis decoder, accounting for band-frequency imbalance. In comparison to a pure masked-GRU encoder and a pure Deep Sets encoder, we find that our model demonstrates superior performance consistently across a variety of forecasting metrics. The Gaussian basis decoder provides flexibility in forecasting by allowing evaluation to be performed at any point in time from a functional representation produced from encoded observations. By decoupling global parameters of the basis function in the latent dimension, we provide local scale- and time-invariant intrinsic basis parameters describing the evolution of the supernova.

SELDON is capable of handling sparse, heteroscedastic, band-frequency-imbalanced multivariate time series with irregular sampling. These panels of gappy time series are inherently dependent and exhibit a nonlinear and nonstationary behavior. We have shown that forecasting such data can be achieved using our GRU-ODE plus Deep Sets encoder model. In particular, SELDON has a unique crucial feature in predicting light curves in the early times before the peak is observed with exceptionally limited amount of data observed. Developing such a framework is useful in other fields of astronomy as well as those outside astronomy.

In the age of the Vera C. Rubin Observatory, SELDON is easily capable of keeping up with the expected LSST alert rate, helping provide predictions for optimal spectroscopic followup scheduling.

Acknowledgments

We gratefully acknowledge the support of the NSF-Simons AI-Institute for the Sky (SkAI) via grants NSF AST-2421845 and Simons Foundation MPS-AI-00010513.

This research used the DeltaAI advanced computing and data resource, which is supported by the National Science Foundation (award OAC 2320345) and the State of Illinois. DeltaAI is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications.

This research was supported in part through the computational resources and staff contributions provided for the Quest high performance computing facility at Northwestern University, which is jointly supported by the Office of the Provost, the Office for Research, and Northwestern University Information Technology.

This research was supported in part by the Illinois Computes project which is supported by the University of Illinois Urbana-Champaign.

We are grateful to Nabeel Rehemtulla for valuable discussions. We thank the anonymous reviewers for their constructive feedback.

References

Alsabti, A. W.; and Murdin, P. 2017. *Handbook of Supernovae*.

Bayer, J.; and Osendorfer, C. 2014. Learning Stochastic Recurrent Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Bianco, F. B.; Ivezić, Ž.; Jones, R. L.; Graham, M. L.; Marshall, P.; Saha, A.; Strauss, M. A.; Yoachim, P.; Ribeiro, T.; Anguita, T.; Bauer, A. E.; Bauer, F. E.; Bellm, E. C.; Blum, R. D.; Brandt, W. N.; Brough, S.; Catelan, M.; Clarkson, W. I.; Connolly, A. J.; Gawiser, E.; Gizis, J. E.; Hložek, R.; Kaviraj, S.; Liu, C. T.; Lochner, M.; Mahabal, A. A.; Mandelbaum, R.; McGehee, P.; Neilsen, E. H., Jr.; Olsen, K. A. G.; Peiris, H. V.; Rhodes, J.; Richards, G. T.; Ridgway, S.; Schwamb, M. E.; Scolnic, D.; Shemmer, O.; Slater, C. T.; Slosar, A.; Smartt, S. J.; Strader, J.; Street, R.; Trilling, D. E.; Verma, A.; Vivas, A. K.; Wechsler, R. H.; and Willman, B. 2022. Optimization of the Observing Cadence for the Rubin Observatory Legacy Survey of Space and Time: A Pioneering Process of Community-focused Experimental Design. *ApJS*, 258(1): 1.

Chen, T. Q.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. 2018. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 6572–6583.

Cho, K.; van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Wu, D.; Carpuat, M.; Carreras, X.; and Vecchi, E. M., eds., *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 103–111. Doha, Qatar: Association for Computational Linguistics.

Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A.; and Bengio, Y. 2015. A Recurrent Latent Variable Model for Sequential Data. In *Advances in Neural Information Processing Systems 28*, 2980–2988.

Feigelson, E. D.; Babu, G. J.; and Caceres, G. A. 2018. Autoregressive Time-Series Methods for Time-Domain Astronomy. *Frontiers in Physics*, 6: 80.

Filippenko, A. V. 1997. Optical Spectra of Supernovae. *ARA&A*, 35: 309–355.

Fortuin, V.; Baranchuk, D.; Rätsch, G.; and Mandt, S. 2019. GP-VAE: Deep Probabilistic Time Series Imputation. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2280–2290.

Fraccaro, M.; Sønderby, S. K.; Paquet, U.; and Winther, O. 2016. Sequential Neural Models with Stochastic Layers. In *Proceedings of the 30th Advances in Neural Information Processing Systems (NeurIPS)*, 2199–2207.

Ivezić, Ž.; Kahn, S. M.; Tyson, J. A.; Abel, B.; Acosta, E.; Allsman, R.; Alonso, D.; AlSayyad, Y.; Anderson, S. F.; Andrew, J.; Angel, J. R. P.; Angeli, G. Z.; Ansari, R.; Antilogus, P.; Araujo, C.; Armstrong, R.; Arndt, K. T.; Astier, P.; Aubourg, É.; Auza, N.; Axelrod, T. S.; Bard, D. J.; Barr, J. D.; Barrau, A.; Bartlett, J. G.; Bauer, A. E.; Bauman, B. J.; Baumont, S.; Bechtol, E.; Bechtol, K.; Becker, A. C.; Becla, J.; Beldica, C.; Bellavia, S.; Bianco, F. B.; Biswas, R.; Blanc, G.; Blazek, J.; Blandford, R. D.; Bloom, J. S.; Bogart, J.; Bond, T. W.; Booth, M. T.; Borgland, A. W.; Borne, K.; Bosch, J. F.; Boutigny, D.; Brackett, C. A.; Bradshaw, A.; Brandt, W. N.; Brown, M. E.; Bullock, J. S.; Burchat, P.; Burke, D. L.; Cagnoli, G.; Calabrese, D.; Callahan, S.; Callen, A. L.; Carlin, J. L.; Carlson, E. L.; Chandrasekharan, S.; Charles-Emerson, G.; Chesley, S.; Cheu, E. C.; Chiang, H.-F.; Chiang, J.; Chirino, C.; Chow, D.; Ciardi, D. R.; Claver, C. F.; Cohen-Tanugi, J.; Cockrum, J. J.; Coles, R.; Connolly, A. J.; Cook, K. H.; Cooray, A.; Covey, K. R.; Cribbs, C.; Cui, W.; Cutri, R.; Daly, P. N.; Daniel, S. F.; Daruich, F.; Daubard, G.; Daues, G.; Dawson, W.; Delgado, F.; Dellapenna, A.; de Peyster, R.; de Val-Borro, M.; Digel, S. W.; Doherty, P.; Dubois, R.; Dubois-Felsmann, G. P.; Durech, J.; Economou, F.; Eifler, T.; Eracleous, M.; Emmons, B. L.; Fausti Neto, A.; Ferguson, H.; Figueroa, E.; Fisher-Levine, M.; Focke, W.; Foss, M. D.; Frank, J.; Freemon, M. D.; Gangler, E.; Gawiser, E.; Geary, J. C.; Gee, P.; Geha, M.; Gessner, C. J. B.; Gibson, R. R.; Gilmore, D. K.; Glanzman, T.; Glick, W.; Goldina, T.; Goldstein, D. A.; Goodenow, I.; Graham, M. L.; Gressler, W. J.; Gris, P.; Guy, L. P.; Guyonnet, A.; Haller, G.; Harris, R.; Hascall, P. A.; Haupt, J.; Hernandez, F.; Herrmann, S.; Hileman, E.; Hoblitt, J.; Hodgson, J. A.; Hogan, C.; Howard, J. D.; Huang, D.; Huffer, M. E.; Ingraham, P.; Innes, W. R.; Jacoby, S. H.; Jain, B.; Jammes, F.; Jee, M. J.; Jenness, T.; Jernigan, G.; Jevremović, D.; Johns, K.; Johnson, A. S.; Johnson, M. W. G.; Jones, R. L.; Juramy-Gilles, C.; Jurić, M.; Kalirai, J. S.; Kallivayalil, N. J.; Kalmbach, B.; Kantor, J. P.; Karst, P.; Kasliwal, M. M.; Kelly, H.; Kessler, R.; Kinnison, V.; Kirkby, D.; Knox, L.; Kotov, I. V.; Krabbendam, V. L.; Krughoff, K. S.; Kubánek, P.; Kuczewski, J.; Kulkarni, S.; Ku, J.; Kurita, N. R.; Lage, C. S.; Lambert, R.; Lange, T.; Langton, J. B.; Le Guillou, L.; Levine, D.; Liang, M.; Lim, K.-T.; Lintott, C. J.; Long, K. E.; Lopez, M.; Lotz, P. J.; Lupton, R. H.; Lust, N. B.; MacArthur, L. A.; Mahabal, A.; Mandelbaum, R.; Markiewicz, T. W.; Marsh, D. S.; Marshall, P. J.; Marshall, S.; May, M.; McKercher, R.; McQueen, M.; Meyers, J.; Migliore, M.; Miller, M.; and Mills, D. J. 2019. LSST: From Science Drivers to Reference Design and Anticipated Data Products. *ApJ*, 873(2): 111.

Kelly, B. C.; Becker, A. C.; Sobolewska, M.; Siemiginowska, A.; and Uttley, P. 2014. Flexible and Scalable Methods for Quantifying Stochastic Variability in the Era of Massive Time-Domain Astronomical Data Sets. *The Astrophysical Journal*, 788(1): 33.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*.

Krishnan, R. G.; Shalit, U.; and Sontag, D. 2017. Structured Inference Networks for Nonlinear State Space Models. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2101–2109.

Li, S.; Zhu, Z.; Ning, L.; Yang, W.; Turner, R. E.; Dupont, E.; and Doucet, A. 2020. GRU-ODE-Bayes: Continuous Modeling of Sporadically-Observed Time Series. In *Proceedings*

of the 37th International Conference on Machine Learning (ICML), 6020–6031.

Lienen, M.; and Günnemann, S. 2023. torchode: A Parallel ODE Solver for PyTorch. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.

Muthukrishna, D.; Narayan, G.; Mandel, K. S.; Biswas, R.; and Hložek, R. 2019. RAPID: Early Classification of Explosive Transients Using Deep Learning. *Publications of the Astronomical Society of the Pacific*, 131(1004): 118002.

Möller, A.; and de Boissière, T. 2020. SuperNNova: An Open-Source Framework for Bayesian, Neural-Network Based Supernova Classification. *Monthly Notices of the Royal Astronomical Society*, 491(3): 4277–4293.

Narayan, G.; and ELAsTiCC Team. 2023. The Extended LSST Astronomical Time-series Classification Challenge (ELAsTiCC). In *American Astronomical Society Meeting Abstracts*, volume 241 of *American Astronomical Society Meeting Abstracts*, 117.01.

Olivier, S. S.; Seppala, L.; and Gilmore, K. 2008. Optical design of the LSST camera. In Atad-Ettedgui, E.; and Lemke, D., eds., *Advanced Optical and Mechanical Technologies in Telescopes and Instrumentation*, volume 7018 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 70182G.

Pasquet, J.; Pasquet, J.; Chaumont, M.; and Fouché, G. 2019. PELICAN: deeP architecturE for the LIght Curve ANalysis. *Astronomy & Astrophysics*, 627: A21.

Rubanova, Y.; Chen, R. T. Q.; and Duvenaud, D. 2019. Latent ODEs for Irregularly-Sampled Time Series. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS)*, 5321–5331.

Rubanova, Y.; Chen, R. T. Q.; and Duvenaud, D. 2019. Latent ODEs for Irregularly-Sampled Time Series. *arXiv e-prints*, arXiv:1907.03907.

Shah, V. G.; Gagliano, A.; Malanchev, K.; Narayan, G.; and Collaboration, L. D. E. S. 2025. ORACLE: A Real-Time, Hierarchical, Deep-Learning Photometric Classifier for the LSST. *arXiv e-prints*, arXiv:2501.01496.

Shah, V. G.; Gagliano, A.; Malanchev, K.; Narayan, G.; and The LSST Dark Energy Science Collaboration. 2025. ORACLE: A Real-Time, Hierarchical, Deep-Learning Photometric Classifier for the LSST. *arXiv e-prints*, arXiv:2501.01496.

Tóth, P.; Cvitkovic, M.; Lozano, I.; Klushyn, A.; and Vogt, M. 2020. TimeGrad: Modeling the Subdistribution of Future Trajectories. In *Proceedings of the 34th Advances in Neural Information Processing Systems (NeurIPS)*.

Yildiz, C.; Heinonen, M.; Rai, P.; and Kaski, S. 2019. ODE2VAE: Deep Generative Second-Order ODEs for Irregular Time Series. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 7104–7113.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R.; and Smola, A. 2017. Deep Sets. *arXiv e-prints*, arXiv:1703.06114.