SEASTAR-TS: SELF-SUPERVISED LIGHT-CURVE REPRESENTATIONS WITH SEMANTIC ADAPTATION*

Anonymous authors
Paper under double-blind review

ABSTRACT

Astronomical surveys collect irregular time series of brightness measurements, called light curves, for far more variable stars and transient events than can receive detailed follow-up. Hand-crafted features remain strong baselines for classifying these light curves: they summarize periodicity, amplitude, and shape even when observations are irregular and noisy. We present SEASTAR-TS, which learns representations directly from photometry by predicting masked observations while accounting for measurement uncertainty. Across ELAsTiCC, MACHO, StarEmbed, and LEAVES, we evaluate the approach under different label budgets and observing conditions. It surpasses hand-crafted-feature baselines on ELAsTiCC and LEAVES. On StarEmbed, its linear and MLP readouts exceed the corresponding feature-based readouts, and on MACHO it has the highest listed scores at two label budgets. As an additional use of the learned representation, alignment with class names and descriptions enables top-k recognition of classes omitted from supervised training. These results show that raw-photometry representations can compete with engineered features and place held-out classes among the top-ranked labels.

1 INTRODUCTION

Time-domain surveys produce far more variable and transient sources than spectroscopic follow-up can classify. ZTF already generates large alert streams (Graham et al., 2019), and Rubin-like surveys represented by ELAsTiCC will increase this scale (LSST Dark Energy Science Collaboration, 2023). A photometric light curve records a source's brightness over time, often through multiple passbands, and therefore carries much of the information available for classification. Light curves are difficult inputs for machine learning: observations arrive at irregular times and with unequal uncertainties, and may capture only part of a transient's rise and decline or only a few cycles of a variable star. Surveys also differ in filters, calibration offsets, and temporal coverage, so the same type of variability can appear differently across datasets. The challenge is to extract a useful description of the underlying variability from these incomplete measurements and to reuse it when labels are limited or the observing survey changes.

These constraints help explain the reliance on hand-crafted light-curve features. Gaussian-process interpolation can model irregular observations (Boone, 2019), but fitting a model to every source adds computational cost at survey scale. Compact, physically interpretable descriptive statistics offer a practical alternative. Periods help identify pulsating variables including RR Lyrae stars. Eclipse depths and shapes characterize binaries. Rise, decline, and color evolution help distinguish transients. Hand-crafted features compress light curves into these astronomically motivated quantities. In the original StarEmbed evaluation, its hand-crafted-feature random forest outperforms the general-purpose time-series foundation models tested there (Li et al., 2026). A model trained on raw observations must learn summaries despite irregular cadence and noisy measurements, without being given the expert-designed quantities that a feature-based classifier receives. This makes hand-crafted features a meaningful test of learned astronomical representations, not merely another baseline.

We use masked reconstruction as a self-supervised pretraining objective to learn representations from raw photometry. In SEASTAR-TS, each observation contributes its time, band, measured value, and uncertainty as one token. A Transformer predicts withheld measurements with an uncertainty-aware

*Code will be released upon acceptance.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Gaussian objective. The resulting encoder is adapted to classification tasks without precomputed periods, colors, or morphology statistics. We evaluate whether these representations can compete with engineered features as label budgets and observing conditions change. Across the experiments, we adapt the encoder with either categorical targets or class descriptions. Class names and descriptions also allow top-k recognition of classes with no supervised examples, without retraining.

We examine these uses on four datasets with different roles. ELAsTiCC provides a controlled comparison with an XGBoost classifier trained on hand-crafted light-curve features under the same split and evaluation protocol. The learned representation with conventional one-hot adaptation surpasses this feature-based baseline. MACHO tests whether the representation remains useful with few labels, where our larger backbone has the highest listed scores at 20 and 500 samples per class. StarEmbed tests adaptation to a different survey, where the adapted representation exceeds the feature-based baseline with linear and MLP readouts. LEAVES combines observations from several surveys. ELAsTiCC also supports a leave-one-class-out test of top-k recognition with class names and descriptions. Together, these settings probe classification from raw photometry as cadence, passbands, and label availability change.

We summarize our **contributions** as follows:

- **Versatility.** We propose SEASTAR-TS, a self-supervised framework for learning representations from raw, irregular light curves with measurement uncertainties. Across four datasets, it supports few-shot classification, adaptation to a new survey, and cross-survey readout transfer.
- **Effectiveness.** We show that learned light-curve representations exceed hand-crafted-feature baselines on astronomical time series datasets. On StarEmbed, the adapted model exceeds the corresponding feature-based scores. These results support learning from raw photometry as an effective alternative to engineered feature tables.
- **Extensibility.** Class names and descriptions extend the framework beyond closed-set classification. In a leave-one-class-out experiment, they enable top-k recognition of classes without supervised examples and outperform random unit-vector controls.

2 RELATED WORK

Feature-Based and Raw-Light-Curve Classifiers. Feature-based pipelines summarize variability through quantities such as periods, amplitudes, and light-curve shape (Cabral et al., 2018). Catalog-scale classifiers built on such summaries have been developed for Gaia DR3 and ASAS-SN (Rimoldini et al., 2023; Christy et al., 2023). Models that learn from observations take a different route. ATAT encodes irregular photometry and tabular information with separate Transformers and studies their combination on ELAsTiCC (Cabrera-Vives et al., 2024). ORACLE uses a hierarchical architecture for LSST-like alert streams (Shah et al., 2025), while ATCAT uses causal attention to support classification as observations arrive (Tung, 2025). These systems establish strong supervised classifiers. Our focus is on whether a pretrained encoder can produce useful light-curve representations across several classification settings while remaining competitive with engineered features.

Self-Supervised Representations for Irregular Light Curves. ASTROMER pretrains on single-band light curves by predicting masked observations, then uses the learned embeddings in downstream models (Donoso-Oliva et al., 2023). Astromer 2 studies few-shot transfer across surveys and combines intermediate encoder states in its light-curve embeddings (Donoso-Oliva et al., 2026). Other work changes how irregular time enters the model. RoMAE extends masked autoencoding with rotary embeddings for continuous positions (Zivanovic et al., 2025), while positional-encoding studies compare alternatives for light curves (Moreno-Cartagena et al., 2023). AstroCo combines attention with convolution and gating to capture local and long-range structure (Tan et al., 2025). Multiband mixtures of experts (Cádiz-Leyton et al., 2025) and pretrained vision backbones applied to light-curve images (Moreno-Cartagena et al., 2025) offer further architectural choices. General-purpose time-series models such as Chronos and Moirai provide a different source of pretrained representations (Ansari et al., 2024; Woo et al., 2024; Liu et al., 2025; Shi et al., 2025). StarEmbed compares such representations with astronomy-specific encoders and engineered features on variable-star tasks (Li et al., 2026). An astronomy-specific multi-view JEPA approach supplies periodograms and phase-folded curves alongside the raw sequence (Rui, 2026). Our encoder instead learns from

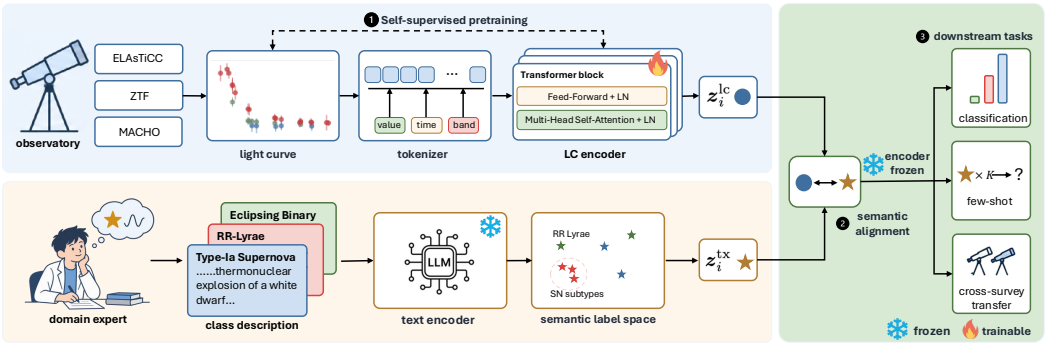


Figure 1: Overview of the SEASTAR-TS framework. A light-curve encoder is pretrained on unlabeled observations through masked reconstruction and then adapted using labeled data. The diagram illustrates class-description targets encoded by a frozen text encoder. The adapted encoder supplies representations for downstream classification and transfer.

observation-level photometry and its measurement uncertainties without requiring those derived views.

Astronomical Information in Training Signals. Cross-modal contrastive learning already has several astronomical precedents. AstroCLIP aligns galaxy images with optical spectra (Parker et al., 2024). AstroM^3 aligns time-series photometry, spectra, and metadata (Rizhko & Bloom, 2025), while Maven aligns supernova light curves with spectra (Zhang et al., 2024). Natural language has also been paired with astronomical observations. PAPERCLIP aligns Hubble images with observing-proposal abstracts (Mishra-Sharma et al., 2024), and recent work aligns X-ray spectra with summaries drawn from scientific papers (Pinciroli Vago et al., 2026). These studies establish cross-modal alignment in astronomy, though their paired inputs and downstream tasks differ. Text-derived class targets have a separate precedent in DeViSE, which scores visual categories absent from labeled image training (Frome et al., 2013). When seen and unseen classes compete at evaluation, scores can favor the seen classes, motivating calibrated stacking (Chao et al., 2016). We use descriptions shared by all objects in an astrophysical class to adapt a photometry-pretrained encoder and rank classes without supervised examples.

3 METHOD

We learn a light-curve encoder in two stages: self-supervised reconstruction of masked observations, followed by supervised adaptation for classification. Class descriptions provide an optional adaptation target that also allows the model to score classes without labeled examples (Figure 1).

Light Curves and Encoder. We represent object i by its irregular sequence

$$x_i = \{(t_j, b_j, v_j, \sigma_j)\}_{j=1}^{n_i},$$

where each observation has a time t_j, passband b_j, photometric measurement v_j, and its uncertainty σ_j. The measurement may be expressed as flux or magnitude, depending on the dataset, with uncertainty in the corresponding unit. Its class label is $y_i \in \{1, \ldots, K\}$. Given labeled light curves $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, the goal is to learn a fixed-dimensional representation that supports classification even when the number and timing of observations vary across objects. Each observation becomes one token rather than a fixed-width time patch (Li et al., 2026; Donoso-Oliva et al., 2023; Moreno-Cartagena et al., 2023). Shorter curves are zero-padded for batching, with a validity mask that excludes padding from attention and the reconstruction loss. Learned multiscale measurement and uncertainty embeddings are combined with a band embedding to form the base token. The band embedding is either a learned lookup or a function of effective wavelength plus a learned band-specific residual. For a new filter, a zero residual lets its wavelength define the initial embedding.

We encode the observation time with fixed Fourier features and use learned FiLM-style scale and shift projections, shared across bands (Perez et al., 2018; Cabrera-Vives et al., 2024):

$$\boldsymbol{e}_j = E_v(v_j) + E_\sigma(\sigma_j) + E_b(b_j), \tag{1}$$

$$\boldsymbol{\phi}(t_j) = [\sin(\omega_1 t_j), \ldots, \sin(\omega_I t_j), \cos(\omega_1 t_j), \ldots, \cos(\omega_I t_j)], \tag{2}$$

$$\tilde{\boldsymbol{e}}_j = \boldsymbol{e}_j \odot \left(\mathbf{1} + \boldsymbol{W}_{\boldsymbol{\alpha}}\boldsymbol{\phi}(t_j) + \boldsymbol{b}_{\boldsymbol{\alpha}}\right) + \boldsymbol{W}_{\boldsymbol{\beta}}\boldsymbol{\phi}(t_j) + \boldsymbol{b}_{\boldsymbol{\beta}}. \tag{3}$$

The frequency grid can be linear or logarithmic to cover the time scales of a dataset. The scale and shift inject the actual observation times into the tokens without first placing measurements on a regular time grid. A learned `[CLS]` token is prepended to the modulated observations, and the sequence passes through L pre-norm Transformer blocks (Vaswani et al., 2017) with LayerNorm and GELU feed-forward layers. The final `[CLS]` state is the fixed-size representation $h = f_\theta(x_i) \in \mathbb{R}^d$. Attention over observation tokens lets this state summarize variability across irregular visits and, for multi-band curves, across filters.

Self-Supervised Pretraining. Following masked reconstruction for irregular light curves (Donoso-Oliva et al., 2023; Zivanovic et al., 2025), we sample a mask M containing 50% of the valid observations in each curve. The encoder sees the remaining measurements and uses a learned query token at each masked position to predict the withheld value from its temporal and passband context. A Gaussian head maps each query's hidden state to a predicted distribution $\mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j^2)$. We represent the measured value and its reported uncertainty by $\mathcal{N}(v_j, \sigma_j^2)$ and minimize its KL divergence to the prediction. Up to terms independent of the model, the masked average loss is

$$\mathcal{L}_{\text{pre-train}} = \frac{1}{|M|} \sum_{j \in M} \left[\tfrac{1}{2} \log \hat{\sigma}_j^2 + \frac{(v_j - \hat{\mu}_j)^2 + \sigma_j^2}{2\hat{\sigma}_j^2} \right]. \tag{4}$$

Unlike RoMAE's unweighted flux-and-variance MSE (Zivanovic et al., 2025), this objective uses the reported measurement uncertainty. The residual and measurement variance appear together in the numerator, while the log-variance term discourages the prediction head from increasing $\hat{\sigma}_j^2$ without bound. Pretraining therefore asks the encoder to recover temporal structure while accounting for the precision of individual measurements. The Gaussian head is discarded after pretraining. Downstream evaluation uses complete, unmasked light curves.

Supervised Adaptation. After pretraining, we fine-tune the encoder using class labels. A conventional adaptation target identifies the class categorically. The class-description variant instead uses a written class definition as its target. For each class k, we write a description d_k covering its physical origin and photometric behavior, including timescale, light-curve shape, periodicity, color evolution, and distinctions from similar classes. We embed each description once with the frozen Qwen3-Embedding-8B text encoder (Zhang et al., 2025) to obtain $\boldsymbol{t}_k \in \mathbb{R}^{4096}$, and every object of class k is assigned the same target $\boldsymbol{t}_k$. A description can express properties shared by related classes, whereas a categorical target supplies only class identity.

We fine-tune f_θ (initialized from the pretrained backbone) together with two projection heads $g_{\text{lc}}, g_{\text{tx}}$ that map the light-curve and text representations into a shared space of dimension $s = 256$. The text projection g_{tx} is a single linear layer from 4,096 to 256 dimensions:

$$\boldsymbol{z}_i^{\text{lc}} = \text{normalize}\big(g_{\text{lc}}(f_\theta(x_i))\big) \in \mathbb{R}^s, \tag{5}$$

$$\boldsymbol{z}_i^{\text{tx}} = \text{normalize}\big(g_{\text{tx}}(\boldsymbol{t}_{y_i})\big) \in \mathbb{R}^s. \tag{6}$$

Here normalize denotes unit-length normalization before cosine-similarity matching. The class descriptions supply the text embeddings, while the learned g_{tx} determines the geometry of their prototypes in the shared space.

An instance-level contrastive loss would treat other objects of the same class as negatives. We use a class-aware contrastive loss (Khosla et al., 2020), in which *all* same-class objects in a batch of size B are mutual positives,

$$p_{ij} = \frac{\mathbb{I}(y_i = y_j)}{\sum_{k=1}^{B} \mathbb{I}(y_i = y_k)}, \tag{7}$$

$$\mathcal{L}_{\text{align}} = -\frac{1}{2B} \sum_{i=1}^{B} \sum_{j=1}^{B} p_{ij} \left[\log \frac{\exp(\tau\, \boldsymbol{z}_i^{\text{lc}} \cdot \boldsymbol{z}_j^{\text{tx}})}{\sum_{k=1}^{B} \exp(\tau\, \boldsymbol{z}_i^{\text{lc}} \cdot \boldsymbol{z}_k^{\text{tx}})} + \log \frac{\exp(\tau\, \boldsymbol{z}_i^{\text{tx}} \cdot \boldsymbol{z}_j^{\text{lc}})}{\sum_{k=1}^{B} \exp(\tau\, \boldsymbol{z}_i^{\text{tx}} \cdot \boldsymbol{z}_k^{\text{lc}})} \right], \tag{8}$$

where τ is a learned temperature, following CLIP (Radford et al., 2021). The two terms align light curves to class prototypes and prototypes to light curves.

Representations and Prediction. For closed-set classification from light curves alone, a task-specific readout is fitted to the adapted encoder state h. The alignment projections shape h during fine-tuning and are discarded for this readout. When metadata are available, their embedding can be combined with h and projected into the shared space for classification. For a class u omitted from supervised fine-tuning but supplied with a description d_u, we compute its prototype from the frozen text encoder and trained g_{tx}, then compare it with the projected input by cosine similarity alongside the existing class prototypes. This extends the candidate set without labeled examples of u or another encoder fine-tuning run.

4 Experiments

We evaluate the representation on four astronomical light-curve datasets. ELAsTiCC tests classification against hand-crafted features on simulated Rubin-like transients and variables (LSST Dark Energy Science Collaboration, 2023). MACHO tests few-shot variable-star classification (Alcock et al., 2000; Donoso-Oliva et al., 2023). StarEmbed tests adaptation to ZTF observations (Graham et al., 2019; Li et al., 2026). LEAVES tests multi-survey classification using ASAS-SN, ZTF, and Gaia (Fei et al., 2024). Dataset provenance and statistics are given in Appendix A. Across these experiments, architectures, checkpoints, training budgets, and readouts follow fixed protocols or validation selection. Test scores are used for final reporting only.

4.1 ELAsTiCC

ELAsTiCC first provides a controlled ablation of five axes: pretraining objective, time encoding, generation, pooling, and backbone. We then compare the learned raw-light-curve representation with an engineered-feature classifier on the same data split and test top-k recognition of held-out classes using class names and descriptions.

Pretraining Backbone Ablation. Across 96 configurations and four label budgets, we used the same fine-tuning and readout protocol to compare design choices. The selected combination uses Gaussian KL, Fourier time features, masked reconstruction with `[CLS]` pooling, and LayerNorm-GELU blocks. The complete design and results are in Appendix C.1.

Classification Protocol and Results. Following ATAT (Cabrera-Vives et al., 2024), we use approximately 1.46M light curves for training, 365K for validation, and a balanced test set of 1,000 objects per class (20K total). The ELAsTiCC backbone uses 64 linearly spaced Fourier frequencies $\omega_k = 2\pi k/(1500\,\text{days})$, a learned lookup for its six bands, and six-scale value and uncertainty embeddings. The 1.8M training and validation curves are used without labels for pretraining. All label-target variants share that checkpoint. Each variant then receives eight independent fine-tunings, each followed by its own inverse-frequency-weighted linear readout on `[CLS]`. Results using both light curves and metadata appear in Appendix C.3.

For the label-target comparison, we use a K-dimensional one-hot vector, a class-description embedding $\boldsymbol{t}_k \in \mathbb{R}^D$, or a random unit vector $\boldsymbol{r}_k \in \mathbb{R}^D$ for each class, where $K = 20$ and $D = 4096$. We draw each random unit vector once as $\boldsymbol{u}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D)$ and set $\boldsymbol{r}_k = \boldsymbol{u}_k/\|\boldsymbol{u}_k\|_2$. All three variants use the same backbone, alignment loss, projection-head design, and optimization schedule.

Table 1 compares light-curve-only classification. Published neural baseline scores come from the cited papers. Our XGBoost baseline uses the same split and metric with 427 ATAT-released ALeRCE features after removing two catalog quantities. Those features summarize each full curve up to 2,048 days after its first alert, whereas our encoder reads at most the first 243 alert-level observations from the same objects. Thus, the split and sources match, but temporal coverage differs. Our one-hot model reaches 0.759 macro-F1 versus 0.739 for XGBoost and exceeds the other listed raw-light-curve methods. Class descriptions reach 0.770 versus 0.759 for both one-hot and random unit-vector targets. Backbone and XGBoost details are in Appendices B.1 and C.2.

Table 1: ELAsTiCC test macro-F1 using light curves only. Our values are mean ± standard deviation over eight independent fine-tunings, each with its own linear readout, from one shared pretrain checkpoint. Baseline uncertainties follow the cited papers. XGBoost was trained once. **Bold** and Underlined mark the best and second-best scores.

Method	Input	Parameters	macro-F1
ATAT (Cabrera-Vives et al., 2024)	raw LC	1.9M	$0.627 \pm 0.004^*$
RoMAE-tiny (Zivanovic et al., 2025)	raw LC	4.7M	0.721
ATCAT (supervised) (Tung, 2025)	raw LC	4.1M	0.718 ± 0.003
ATCAT (pt + ft) (Tung, 2025)	raw LC	4.1M	0.655 ± 0.001
SWINv2 (grid) (Moreno-Cartagena et al., 2025)	LC image	29M	0.646 ± 0.0022
SWINv2 (overlap) (Moreno-Cartagena et al., 2025)	LC image	29M	0.655 ± 0.0028
XGBoost (Chen & Guestrin, 2016)	427 hand-crafted features (Cabrera-Vives et al., 2024)	–	0.739
Ours, one-hot label	raw LC	3.62M	$\underline{0.759 \pm 0.0023}$
Ours, random unit vectors	raw LC	3.62M	$\underline{0.759 \pm 0.0032}$
Ours, class description	raw LC	3.62M	$\mathbf{0.770 \pm 0.0017}$

*: We use the number ATAT reports in its own paper, not the value re-derived by the SWINv2 paper.

Top-k Recognition of Held-Out Classes. Time-domain surveys may uncover rare transients before enough labeled examples exist to train a classifier. Kilonovae illustrate this setting: theoretical light-curve predictions predated the first well-observed event, GW170817/AT2017gfo. We test whether a class name or description can place a held-out class among the highest-ranked labels without retraining the encoder. Here, a held-out class has no labeled examples in supervised fine-tuning, although its unlabeled light curves are included in the shared self-supervised pretraining pool.

We test this using light curves and metadata in a leave-one-class-out protocol. All folds start from the same checkpoint, pretrained without labels on the full ELAsTiCC training and validation pool. For each of the 20 classes in turn, we remove every object of that class from supervised fine-tuning and its validation set, then fine-tune on the remaining 19 classes. Its class prototype is absent from supervised alignment and checkpoint selection. At test time we embed its label with the same text encoder and projection, add it to the 19 seen labels, and rank all 20 by cosine similarity to each object, with no retraining. We report AUROC, average rank, and top-3 and top-5 accuracy on the objects of the held-out class, averaged over the 20 runs. Class-name and random-unit-vector controls follow the same leave-one-class-out protocol. Appendix C.4 gives their label definitions and a control that removes explicit mentions of the held-out class from the descriptions of seen classes.

Table 2 shows that class descriptions reach 0.780 AUROC, 0.559 top-3 accuracy, and 0.756 top-5 accuracy. Class names also substantially outperform random unit vectors. Uncalibrated top-1 accuracy on the held-out class is 0.035, below the 0.05 uniform-choice rate, consistent with the seen-class bias diagnosed in Table 17: 73% of errors go to a seen class in the same astrophysical family, and mean own-label similarity is 0.35 for held-out objects versus 0.72 for seen objects. Post-hoc calibrated stacking raises held-out top-1 accuracy to 0.815 while reducing seen-class accuracy from 0.903 to 0.640. The uncalibrated results therefore support top-k recognition, with calibration analyzed separately as a diagnostic. Per-class results and the description-cleaning control are in Appendix C.4.

Table 2: Top-k recognition of classes held out from supervised fine-tuning on ELAsTiCC using light curves and metadata. Each class is scored among 20 labels. The class-description row uses the original texts. AUROC ranks held-out-class objects against other objects. Average rank and top-3/top-5 accuracy locate the correct label among candidates for its objects. Values are averaged over 20 held-out classes.

Label	AUROC ↑	Avg. rank ↓	Acc@3 ↑	Acc@5 ↑
Random unit vectors	0.460	12.39	0.055	0.147
Class name	0.771	4.92	0.465	0.683
Class description	**0.780**	**4.49**	**0.559**	**0.756**

Table 3: Balanced macro-F1 on Alcock at three label budgets (samples per class). For our rows, values are mean ± standard deviation over three folds. The encoder is fine-tuned and a linear readout is fitted separately in each fold. AstroCo evaluates a frozen encoder with a linear readout (Tan et al., 2025). Other published baseline values are reproduced as reported.

Method	Parameters	20 SPC	100 SPC	500 SPC
SWINv2 (Moreno-Cartagena et al., 2025)	29M	0.601	-	0.774
Astromer 1 (Donoso-Oliva et al., 2023)	0.66M	0.617	0.697	0.729
Astromer 2 (Donoso-Oliva et al., 2026)	3.95M	0.656	0.710	0.742
AstroCo-S (Tan et al., 2025)	5.9M	0.666	0.749	0.791
AstroCo-L (Tan et al., 2025)	15.2M	<u>0.676</u>	**0.759**	<u>0.792</u>
Ours (S-size)	3.68M	0.634 ± 0.033	0.729 ± 0.007	0.766 ± 0.006
Ours (L-size)	21.53M	$\mathbf{0.697 \pm 0.024}$	<u>0.753 ± 0.015</u>	$\mathbf{0.804 \pm 0.008}$

4.2 MACHO

MACHO tests how well the representation adapts when only a few labels are available for each variable-star class. We compare two backbone sizes across three label budgets on the Alcock subset.

Experimental Setting. We pretrain two backbone sizes on MACHO's long-baseline R-band light curves: S-size has width 192, eight Transformer blocks, and three attention heads, while L-size has width 384, twelve blocks, and six heads. We evaluate the six-class Alcock subset with 20, 100, and 500 labeled samples per class (SPC). Both retain the Gaussian KL objective and Transformer block design selected on ELAsTiCC. Their input modules use 128 logarithmically spaced Fourier frequencies, wavelength-conditioned band embeddings with learned residuals, and two-scale measurement and uncertainty embeddings. Using the label budgets and folds of Astromer 2 (Donoso-Oliva et al., 2026), each of three disjoint folds receives class-description fine-tuning and its own linear readout on the adapted `[CLS]` output. The 1,000-step fine-tuning budget and linear readout were fixed across sizes, label budgets, and folds. We report balanced macro-F1 over folds. Table 3 uses the published Astromer 2 results and its re-evaluation of Astromer 1. The pretraining changes, model sizes, and sampling procedure are detailed in Appendix D.1.

Results. In Table 3, our L-size model has the highest listed scores at 20 and 500 SPC and is close to AstroCo-L at 100 SPC. MACHO uses independently pretrained weights on observed, long-baseline, single-band variable-star light curves. Its results therefore show that the representation-learning framework also works beyond ELAsTiCC's simulated multiband light curves. Increasing backbone size improves our score at every label budget, with the largest gain at 20 SPC (0.697 versus 0.634), where supervision is most limited. From 20 to 500 SPC, the S-size score rises from 0.634 to 0.766 and the L-size score from 0.697 to 0.804. The adapted representations remain useful as more labeled examples become available within the tested range.

4.3 StarEmbed

StarEmbed (Li et al., 2026) tests whether the MACHO-pretrained encoder adapts to ZTF periodic variables observed through different filters and sampling patterns. We examine the effect of continued self-supervised pretraining and compare downstream readouts with published feature-based results.

Experimental Setting and Evaluation Protocol. We start from the S-size backbone pretrained on MACHO and follow StarEmbed's train/validation/test partitions and macro-F1 metric. We merge ZTF g- and r-band observations into one time-ordered sequence, with wavelength-conditioned embeddings for the new filters, and exclude the sparsely observed i band. The table compares the unadapted encoder with one-hot and class-description fine-tuning, each with or without continued self-supervised pretraining on StarEmbed curves. For each condition, we freeze the resulting encoder and fit readouts on its `[CLS]` output. Logistic regression gives one deterministic linear fit on fixed embeddings. The MLP results summarize eight readout seeds on those same embeddings. The test evaluation excludes 13 curves without g- or r-band observations. Preprocessing and readout training are detailed in Appendix D.2.

Table 4: StarEmbed macro-F1 with linear and MLP readouts. For each of our rows, the linear score comes from one deterministic fit on a fixed encoder, while MLP values are mean ± standard deviation over eight readout seeds on that encoder. The ± values do not include variation across encoder training runs. Published baseline ± values follow the benchmark definitions. Our parameter counts include the readout and are shown as linear/MLP. Astromer 2 is listed at 3.95M parameters following its original paper (Donoso-Oliva et al., 2026). The StarEmbed benchmark lists the same model at 5.4M (Li et al., 2026). Best and second-best scores within each readout are **bold** and underlined.

Method	Parameters	Linear head	MLP head
Astromer 1 (Donoso-Oliva et al., 2023)	0.66M	0.072	0.138 ± 0.020
Astromer 2 (Donoso-Oliva et al., 2026)	3.95M	0.521	0.470 ± 0.023
Moirai-small (Woo et al., 2024)	14M	0.579	0.594 ± 0.019
Moirai-2-small (Liu et al., 2025)	11.4M	0.577	0.603 ± 0.015
Chronos-tiny (Ansari et al., 2024)	8M	0.617	0.643 ± 0.023
Chronos-Bolt-tiny (Ansari et al., 2024)	9M	0.580	0.589 ± 0.015
Time-MoE (Shi et al., 2025)	113M*	0.597	0.627 ± 0.019
Chronos, full fine-tuning (Li et al., 2026)	8M	0.661	0.659 ± 0.027
Hand-crafted features (Li et al., 2026)	–	0.714	0.723 ± 0.027
Ours, no encoder adaptation	3.68M/4.53M	0.410	0.478 ± 0.016
Ours, one-hot fine-tuning	3.68M/4.53M	0.709	0.717 ± 0.010
Ours, continued pretraining + one-hot fine-tuning	3.68M/4.53M	<u>0.724</u>	<u>0.733 ± 0.014</u>
Ours, class-description fine-tuning	3.68M/4.53M	0.714	0.722 ± 0.005
Ours, continued pretraining + class-description fine-tuning	3.68M/4.53M	**0.740**	**0.744 ± 0.003**

*: 50M activated, 113M total

Results. The frozen MACHO-pretrained encoder transfers weakly to StarEmbed: it scores 0.410 with a linear readout and 0.478 with an MLP. Its linear score is below Astromer 2 (0.521) and Chronos-tiny (0.617). This gap is consistent with the change from MACHO R-band pretraining to ZTF g/r observations with different filters and sampling. Supervised adaptation substantially improves both readouts. Adding continued self-supervised pretraining before fine-tuning further raises the one-hot and class-description MLP scores to 0.733 and 0.744, respectively. After adaptation, our best S-size model has the highest scores among the listed methods with both linear (0.740) and MLP (0.744) readouts, exceeding the corresponding hand-crafted-feature scores of 0.714 and 0.723. These results show that the representation can be adapted effectively from MACHO to ZTF. With a random-forest readout, the L-size model reaches 0.780, which is 0.024 below the original StarEmbed hand-crafted-feature score of 0.804 (Li et al., 2026). JEPA separately reports 0.744 for its own implementation of that feature baseline (Rui, 2026), showing that the RF reference depends on the evaluation pipeline. The full comparison is in Table 19.

4.4 LEAVES

LEAVES tests classification of variable stars observed by ASAS-SN, ZTF, and Gaia, whose filters and cadences differ. We compare with published feature-based classifiers and examine whether a readout fitted on one survey transfers to another after joint backbone adaptation.

Experimental Setting. We use the released light curves and follow the published LEAVES ten-class task and class-stratified 80:20 train–test splitting scheme. We pretrain the S-size and L-size backbones from scratch on unlabeled curves from the full LEAVES catalog, using the logarithmic time grid and wavelength-conditioned band embeddings of the S-size and L-size designs. The five independent class-description fine-tunings and their MLP readouts use only the labeled training partition. Each adapted backbone is frozen before its readout is fitted on the `[CLS]` output. We report weighted F1 because the published feature-based baselines report only this metric.

Table 5: LEAVES ten-class weighted F1. HC denotes hand-crafted features. Baselines are from Fei et al. (2024). Our values are mean ± standard deviation over five independent fine-tunings, each with its own MLP readout. **Bold** and Underlined mark the best and second-best scores.

Method	weighted F1
HC + SVM	0.82
HC + MLP	0.87
HC + hierarchical RF	0.93
Ours (S-size)	<u>0.9441 ± 0.0018</u>
Ours (L-size)	**0.9491 ± 0.0009**

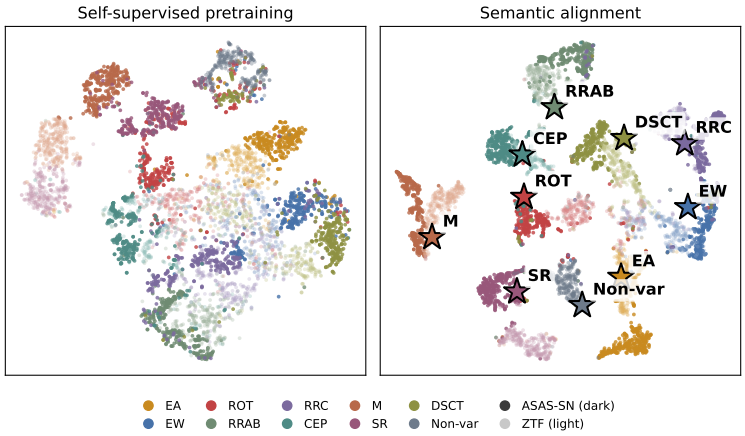


Figure 2: t-SNE views of test-set representations from the S-size model. Left: pretrained `[CLS]` embeddings. Right: projected embeddings after class-description fine-tuning, with stars marking the class-description anchors. The panels use different representation spaces and separate t-SNE fits. The right panel illustrates the alignment space, where curves of the same class from ASAS-SN and ZTF appear near the same anchor, with survey-dependent substructure.

Results. Table 5 shows that the L-size model reaches 0.9491 mean weighted F1 and the S-size model reaches 0.9441. Both exceed the published hand-crafted-feature baseline of 0.93. Figure 2 provides a qualitative view of the projected alignment space: class organization persists across surveys with different cadences and passbands, even though survey-dependent substructure remains. This suggests that the learned representation retains variability-type information across survey conditions. Appendix E reports a separate readout-transfer experiment. After alignment with labels from both ASAS-SN and ZTF, a readout trained on one survey scores within 0.0055 weighted F1 of a readout trained on the evaluation survey. This shows readout transfer within the jointly aligned representation.

5 Conclusion

We introduced SeaStar-TS to learn representations from raw, irregular astronomical light curves using masked pretraining and measurement uncertainties. Across simulated transients and observed variable stars, the learned representations surpass hand-crafted-feature baselines on ELAsTiCC and LEAVES and perform strongly with limited labels on MACHO. After adaptation from MACHO to ZTF, the model exceeds the corresponding feature-based scores under linear and MLP readouts on StarEmbed. Class names and descriptions add top-k recognition of classes without supervised examples. Together, these results show that representations learned from photometry can support classification across different sources, label budgets, and survey conditions.

The broader challenge for AI for Science is how to bring a field's accumulated knowledge into models trained end to end. Astronomical classification draws on extensive domain knowledge, much of it encoded in hand-crafted summaries. Learning useful counterparts from sparse, noisy observations remains difficult. Our study offers one approach: pretraining uses the structure and uncertainties of the measurements, while adaptation can use either categorical labels or astrophysical class descriptions. The comparisons with engineered features show that this approach can learn useful representations without supplying those summaries as input.

Limitations and Future Work. The remaining challenge is to make learned representations consistently effective across observing domains and downstream readouts. Our cross-survey evidence comes from jointly supervised alignment, while class-description adaptation relies on expert-authored texts. Transfer to a survey without labeled examples and adaptation of poorly characterized classes warrant further study. Class names and descriptions support top-k recognition of held-out classes, while reliable first-choice decisions still need better calibration. Future work should separate intrinsic variability from the survey observation process, including passbands, cadence, and noise, and test whether this improves transfer to new instruments. Spectra and metadata could provide complementary constraints on the same source, while comparisons with independently measured physical properties could reveal what the representation captures.

AI USAGE DISCLOSURE

Generative AI tools assisted with manuscript drafting and editing, literature identification, feedback on experimental framing, interpretation of reported results, and software development and debugging. The authors reviewed and verified AI-assisted content and take full responsibility for the final paper.

REPRODUCIBILITY STATEMENT

Section 3 specifies the encoder, training objectives, and prediction paths, while Section 4 describes the data splits, evaluation metrics, and readout protocols used for each benchmark. Appendix A records dataset provenance and statistics, and Appendix B details the model configurations and parameter-counting convention. The ELAsTiCC ablations, feature baseline, and held-out-class protocol are documented in Appendix C. Appendices D.1, D.2, and E provide the remaining dataset-specific training and evaluation details. Table captions identify the source of each reported mean and variance. Code will be released upon acceptance.

REFERENCES

Charles Alcock, RA Allsman, David R Alves, TS Axelrod, Andrew C Becker, DP Bennett, Kem H Cook, N Dalal, Andrew John Drake, KC Freeman, et al. The macho project: microlensing results from 5.7 years of large magellanic cloud observations. *The Astrophysical Journal*, 542(1):281–307, 2000.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Syndar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024.

Kyle Boone. Avocado: Photometric classification of astronomical transients with gaussian process augmentation. *The Astronomical Journal*, 158(6):257, 2019.

J. B. Cabral, B. Sánchez, F. Ramos, S. Gurovich, P. Granitto, and J. Vanderplas. From fats to feets: Further improvements to an astronomical feature extraction tool based on machine learning. *Astronomy and Computing*, 25:213–220, 2018.

G Cabrera-Vives, Daniel Moreno-Cartagena, Nicolás Astorga, I Reyes-Jainaga, F Förster, P Huijse, J Arredondo, AM Muñoz Arancibia, A Bayo, M Catelan, et al. Atat: Astronomical transformer for time series and tabular data. *Astronomy & Astrophysics*, 689:A289, 2024.

Martina Cádiz-Leyton, Guillermo Cabrera-Vives, Pavlos Protopapas, Daniel Moreno-Cartagena, and Ignacio Becker. Astro-moe: Mixture of experts for multiband astronomical time series. *arXiv preprint arXiv:2507.12611*, 2025.

Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *Computer Vision – ECCV 2016*, pp. 52–68, 2016. doi: 10.1007/978-3-319-46475-6_4.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Xiaodian Chen, Shu Wang, Licai Deng, Richard De Grijs, Ming Yang, and Hao Tian. The zwicky transient facility catalog of periodic variable stars. *The Astrophysical Journal Supplement Series*, 249(1):18, 2020.

CT Christy, T Jayasinghe, KZ Stanek, CS Kochanek, TA Thompson, BJ Shappee, T WS Holoien, JL Prieto, Subo Dong, and W Giles. The asas-sn catalogue of variable stars x: discovery of 116 000 new variable stars using g-band photometry. *Monthly Notices of the Royal Astronomical Society*, 519(4):5271–5287, 2023.

C. Donoso-Oliva, I. Becker, P. Protopapas, G. Cabrera-Vives, M. Cádiz-Leyton, and D. Moreno-Cartagena. Generalizing across astronomical surveys: Few-shot light curve classification with astromer 2. *Astronomy & Astrophysics*, 2026. In press.

Cristobal Donoso-Oliva, Ignacio Becker, Pavlos Protopapas, Guillermo Cabrera-Vives, M Vishnu, and Harsh Vardhan. Astromer: A transformer-based embedding for the representation of light curves. *Astronomy & Astrophysics*, 670:A54, 2023.

Ya Fei, Ce Yu, Kun Li, Xiaodian Chen, Yajie Zhang, Chenzhou Cui, Jian Xiao, Yunfei Xu, and Yihan Tao. Leaves: an expandable light-curve data set for automatic classification of variable stars. *The Astrophysical Journal Supplement Series*, 275(1):10, 2024.

Andrea Frome, Greg S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Matthew J Graham, Shrinivas R Kulkarni, Eric C Bellm, Scott M Adams, Cristina Barbarino, Nadejda Blagorodnova, Dennis Bodewits, Bryce Bolin, Patrick R Brady, S Bradley Cenko, et al. The zwicky transient facility: science objectives. *Publications of the Astronomical Society of the Pacific*, 131 (1001):078001, 2019.

T Jayasinghe, CS Kochanek, KZ Stanek, BJ Shappee, T WS Holoien, Toda A Thompson, JL Prieto, Subo Dong, M Pawlak, JV Shields, et al. The asas-sn catalogue of variable stars i: The serendipitous survey. *Monthly Notices of the Royal Astronomical Society*, 477(3):3145–3163, 2018.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Weijian Li, Hong-Yu Chen, Nabeel Rehemtulla, Ved G. Shah, Dongho Kim, Dennis Wu, Qinjie Lin, Adam A. Miller, and Han Liu. Starembed: Benchmarking time series foundation models on astronomical observations of variable stars. In *Forty-third International Conference on Machine Learning*, 2026.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025.

LSST Dark Energy Science Collaboration. The desc elasticc challenge. https://portal.nersc.gov/cfs/lsst/DESC_TD_PUBLIC/ELASTICC/, 2023.

Siddharth Mishra-Sharma, Yiding Song, and Jesse Thaler. Paperclip: Associating astronomical observations and natural language with multi-modal models. *arXiv preprint arXiv:2403.08851*, 2024.

Daniel Moreno-Cartagena, Guillermo Cabrera-Vives, Pavlos Protopapas, Cristobal Donoso-Oliva, Manuel Pérez-Carrasco, and Martina Cádiz-Leyton. Positional encodings for light curve transformers: Playing with positions and attention. *arXiv preprint arXiv:2308.06404*, 2023.

Daniel Moreno-Cartagena, Pavlos Protopapas, Guillermo Cabrera-Vives, Martina Cádiz-Leyton, Ignacio Becker, and Cristóbal Donoso-Oliva. Leveraging pre-trained vision transformers for multi-band photometric light curve classification. *Astronomy & Astrophysics*, 703:A41, 2025.

Liam Parker, François Lanusse, Siavash Golkar, Leopoldo Sarra, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Geraud Krawezik, Michael McCabe, et al. Astroclip: a cross-modal foundation model for galaxies. *Monthly Notices of the Royal Astronomical Society*, 531(4):4990–5011, 2024. doi: 10.1093/mnras/stae1450.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Nicolò Oreste Pinciroli Vago, Rocco Di Tella, Carolina Cuesta-Lázaro, Michael J. Smith, Cecilia Garraffo, and Rafael Martínez-Galarza. Augmenting representations with scientific papers. *arXiv preprint arXiv:2603.04516*, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

L. Rimoldini, B. Holl, P. Gavras, M. Audard, J. De Ridder, E. Distefano, L. Eyer, P. Garcia-Lario, L. P. Guy, G. Jevardat de Fombelle, et al. Gaia data release 3: All-sky classification of 12.4 million variable sources into 25 classes. *Astronomy & Astrophysics*, 674:A14, 2023.

Mariia Rizhko and Joshua S. Bloom. Astrom^3: A self-supervised multimodal model for astronomy. *The Astronomical Journal*, 170(1):28, 2025. doi: 10.3847/1538-3881/adcbad.

Yicheng Rui. Domain-informed multi-view self-distillation for astronomical light-curve representation learning with jepa. *arXiv preprint arXiv:2606.28446*, 2026.

Ved G Shah, Alex Gagliano, Konstantin Malanchev, Gautham Narayan, Alex I Malz, and LSST Dark Energy Science Collaboration. Oracle: A real-time, hierarchical, deep learning photometric classifier for the lsst. *The Astrophysical Journal*, 995(1):4, 2025.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *International conference on learning representations*, volume 2025, pp. 34635–34667, 2025.

Antony Tan, Pavlos Protopapas, Martina Cádiz-Leyton, Guillermo Cabrera-Vives, Cristobal Donoso-Oliva, and Ignacio Becker. ASTROCO: Self-supervised conformer-style transformers for light-curve embeddings. In *NeurIPS Workshop on Machine Learning and the Physical Sciences (ML4PS)*, 2025.

Zora Tung. Atcat: Astronomical timeseries causal transformer. *arXiv preprint arXiv:2511.00614*, 2025.

Antonella Vallenari, Anthony GA Brown, Timo Prusti, Jos HJ De Bruijne, F Arenou, Carine Babusiaux, Michael Biermann, Orlagh L Creevey, Christine Ducourant, Dafydd Wyn Evans, et al. Gaia data release 3: Summary of the content and survey properties. *Astronomy & Astrophysics*, 674:A1, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 2017.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53140–53164. PMLR, 21–27 Jul 2024.

Gemma Zhang, Thomas Helfer, Alexander T. Gagliano, Siddharth Mishra-Sharma, and V. Ashley Villar. Maven: A multimodal foundation model for supernova science. *Machine Learning: Science and Technology*, 5:045069, 2024. doi: 10.1088/2632-2153/ad990d.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Uros Zivanovic, Serafina Di Gioia, Andre Scaffidi, Martín de los Rios, Gabriella Contardo, and Roberto Trotta. Rotary masked autoencoders are versatile learners. *Advances in Neural Information Processing Systems*, 38:133952–133987, 2025.

APPENDIX

A SUMMARY OF THE DATASETS

Table 6: Summary of the datasets.

Dimension	ELAsTiCC	MACHO	ZTF	LEAVES
No. of light curves	1,821,875	1,529,385 (Alcock 21,444)	42,585*	1,112,453***
Physical content	transient & periodic	periodic	periodic	periodic & non-variables
Labeled	labeled	unlabeled (Alcock labeled)	labeled	labeled
No. of classes	20	6 (Alcock)	7	10
Median observations per light curve	21	711	738	175
Bands	6 (LSST u, g, r, i, z, Y)	1 (MACHO r)	2 (StarEmbed g, r)	4**
Origin	LSST simulation	real survey	real survey	real multi-survey compilation

$*$: This is a curated periodic variable subset of ZTF DR2/DR3 from the StarEmbed benchmark.
$**$: ZTF g / ASAS-SN g / ASAS-SN v / Gaia G.
$***$: This LEAVES count is 92 below the total implied by the published variable and non-variable counts, after excluding unusable light curves.

Dataset Provenance.

- **ELAsTiCC** is a multiband simulation of transient and variable sources designed to emulate the Vera C. Rubin Observatory's Legacy Survey of Space and Time alert stream (LSST Dark Energy Science Collaboration, 2023). Its photometric values are fluxes.
- **MACHO** is a long-baseline photometric survey of stellar variability. We use R-band magnitudes from the curated MACHO release accompanying Astromer and the labeled Alcock subset for few-shot evaluation (Alcock et al., 2000; Donoso-Oliva et al., 2023).
- **StarEmbed** supplies a curated ZTF periodic-variable benchmark, for which we use g- and r-band observations (Graham et al., 2019; Li et al., 2026).
- **LEAVES** combines labeled magnitude curves from ASAS-SN, Gaia DR3, and ZTF DR2 under a common variable-star taxonomy (Jayasinghe et al., 2018; Vallenari et al., 2023; Chen et al., 2020; Fei et al., 2024).

B MODEL SIZES

B.1 BACKBONE

Our models encode each observation as a token containing a photometric measurement, its uncertainty, time, and passband, then use a Transformer to produce a pooled encoder output at the `[CLS]` token. The encoder is first trained by masked reconstruction with a Gaussian prediction head. The ELAsTiCC configuration has width $d = 192$, eight pre-norm Transformer blocks, three attention heads of dimension 64, and a feed-forward width of 768. Table 7 specifies every layer. The input embedding contributes 53,760 parameters and the blocks contribute 3,558,912, giving 3,612,672 distinct backbone parameters. The 386-parameter Gaussian head raises the pretraining total to 3,613,058 and is discarded afterward.

Table 7: Layer-by-layer composition and parameter count of the self-supervised pretraining backbone f_θ used for the ELAsTiCC results. d is the embedding dimension, L the encoder depth, and H the number of attention heads. Parameter counts use the configuration d=192, L=8, H=3 ($d/H = 64$). ELAsTiCC photometric measurements are fluxes.

Module	Operation	Output dim.	#Params
Input embedding (per observation j)			
Measurement embed.	tanh-feature (6 scales) $\to$ `Linear`$(6, d)$	d	$7d$
Uncertainty embed.	tanh-feature (6 scales) $\to$ `Linear`$(6, d)$	d	$7d$
Band embed.	lookup, 6 LSST bands	d	$6d$
Fourier features $\phi(t_j)$	[sin, cos], 64 frequencies (fixed, not learned)	128	0
Time projection (scale) $\alpha(t_j)$	`Linear`$(128, d)$	d	$129d$
Time projection (shift) $\beta(t_j)$	`Linear`$(128, d)$	d	$129d$
`[CLS]` token	learned constant	d	d
query token (masked pos.)	learned constant	d	d
Transformer encoder block (repeated $\times L$)			
Pre-attn. norm	LayerNorm(d)	d	$2d$
QKV proj.	`Linear`$(d, 3d)$, bias	$3d$	$3d^2 + 3d$
Self-attention	H-head scaled dot-product	d	0
Output proj.	`Linear`(d, d), bias	d	$d^2 + d$
Pre-FFN norm	LayerNorm(d)	d	$2d$
Feed-forward	`Linear`$(d, 4d)$-GELU-`Linear`$(4d, d)$	d	$8d^2 + 5d$
block subtotal			$12d^2 + 13d$
Output heads			
Pooling (`[CLS]` gather)	index gather, no parameters	d	0
Gaussian head (pretrain only)†	`Linear`$(d, 2)$, softplus on $\hat{\sigma}$	2	$2d + 2$
Backbone total	$280d + L(12d^2 + 13d)$		**3,612,672**
Pretraining total	$282d + 2 + L(12d^2 + 13d)$		**3,613,058**

†Discarded after pretraining. Only the pooled `[CLS]` encoder output $h = h_{\text{CLS}} \in \mathbb{R}^d$ is carried into fine-tuning ($\approx$3.61M total parameters at d=192, L=8).

The ELAsTiCC backbone defines the smaller Transformer configuration used in our experiments. For MACHO and LEAVES, we retain that Transformer design in the S-size model and also train an L-size model to test whether the representation improves with capacity. S-size has width 192, eight blocks, and three heads. L-size has width 384, twelve blocks, six heads, and a feed-forward width of 1,536. Both use 64 dimensions per head. We pretrain the two sizes independently on MACHO and LEAVES. StarEmbed starts from MACHO-pretrained backbones, using S-size in the main table and both sizes in the random-forest comparison.

The S-size backbone retains the ELAsTiCC Transformer and changes its input encoding as summarized in Table 8. It uses 128 logarithmically spaced Fourier frequencies instead of 64 linearly spaced ones. A $1 \to 64 \to d$ MLP embeds each band's effective wavelength, with a learned residual for each allocated filter ID. The MACHO pretraining checkpoint allocates 15 residual rows, including rows for filters absent from MACHO. An unused row remains zero, so the wavelength MLP supplies that filter's initial embedding. A zero row can also be appended for a new filter ID. Measurement and

uncertainty embeddings use two scales instead of six. These input changes bring the S-size backbone to 3,674,624 parameters, 61,952 more than ELAsTiCC. The corresponding L-size backbone has 21,524,864 parameters.

Table 8: Input-encoding differences between the ELAsTiCC and S-size backbones. Both use the same $d = 192$ Transformer with eight blocks and three attention heads.

Module	ELAsTiCC backbone	S-size backbone
Time encoding	64 Fourier frequencies. Two $\texttt{Linear}(128, d)$ projections	128 log-spaced Fourier frequencies. Two $\texttt{Linear}(256, d)$ projections
Band embedding	Learned lookup for 6 bands	Wavelength MLP $1 \to 64 \to d$ plus 15 allocated residual rows
Measurement embedding	6 scales. Two $\texttt{Linear}(6, d)$ projections	2 scales. Two $\texttt{Linear}(2, d)$ projections

B.2 Dataset-Specific Inference Models

Each dataset uses the backbone with its own input modules and classification readout. The totals below count the modules applied to each light curve at inference.

ELAsTiCC. With light curves alone, the 192-dimensional backbone output is classified by a 20-class linear layer with 3,860 parameters. All three label-target variants therefore have 3,616,532 inference parameters (3.62M). With metadata, a 170-to-192-dimensional metadata embedder adds 776,448 parameters. The concatenated 384-dimensional light-curve and metadata representation passes through a $384 \to 384 \to 256$ projection with 246,400 parameters and a 20-class linear readout with 5,140 parameters. The one-hot, random-unit-vector, and class-description variants use this same inference architecture, totaling 4,640,660 parameters (4.64M).

MACHO. The two backbones are pretrained on MACHO R-band light curves. The evaluation model includes one additional zero-initialized residual row for a filter ID unused in MACHO. It adds 192 parameters to S-size or 384 to L-size without changing MACHO predictions. The six-class linear readout on the adapted `[CLS]` output adds 1,158 or 2,310 parameters. The inference models used in Table 3 therefore contain 3,675,974 (3.68M) and 21,527,558 (21.53M) parameters, respectively.

LEAVES. We pretrain the S-size and L-size backbones from scratch on LEAVES before supervised adaptation. A ten-class MLP readout on the adapted `[CLS]` output has widths 1,024, 512, and 256 and adds 856,330 or 1,052,938 parameters, giving totals of 4,530,954 (4.53M) and 22,577,802 (22.58M). The nine-class readout in the cross-survey experiment has 257 fewer parameters at either size. The rounded totals are unchanged.

StarEmbed. We use the MACHO-pretrained S-size architecture on a merged, time-ordered sequence of ZTF g- and r-band observations, excluding i. Joint robust normalization retains the relative offset between g and r. The 15-row checkpoint already has a residual row assigned to ZTF g, but MACHO pretraining never uses that filter, so its residual stays zero. ZTF r is a different filter ID from MACHO R and has no row in the checkpoint. We append one zero-initialized row for ZTF r, adding 192 parameters. The wavelength network therefore supplies the initial embedding for both ZTF filters, and their residuals can be updated during StarEmbed adaptation. We average the `[CLS]` outputs of eight deterministic windows of at most 1,024 observations per star, which adds no parameters. The evaluation set contains 8,276 objects. The S-size backbone has 3,674,816 parameters. A seven-class logistic readout gives 3,676,167 (3.68M) total, and an MLP readout with widths 1,024, 512, and 256 gives 4,530,375 (4.53M). Table 19 also evaluates an L-size backbone with a random-forest readout.

B.3 Parameter-Counting Convention

For our models, the reported size includes each distinct weight used to map a light curve to its prediction, including the readout. The Gaussian prediction head and learned alignment temperature are training-only. On MACHO, LEAVES, and StarEmbed, the light-curve and text alignment

projections are also absent from the inference path because classification reads the backbone's [CLS] representation. The S-size and L-size light-curve projections contain 115,200 and 246,400 parameters. Across all datasets, the text projection is one linear layer from 4,096 to 256 dimensions, with $4096 \times 256 + 256 = 1{,}048{,}832$ trainable parameters. ELAsTiCC with metadata is different: its 246,400-parameter light-curve projection remains in the inference path and is included above. The text projection is training-only in each setting because class-label embeddings can be computed and stored once. Adding a new candidate class requires one additional pass through the text encoder and projection. Published baseline counts follow their cited sources and may use different conventions.

Table 9 reconciles the 4.47M S-size and 23.9M L-size pretraining state-dictionary sums with the inference parameter counts by separating duplicated entries for the shared time encoding, the training-only Gaussian head, and non-trainable buffers.

Table 9: Decomposition of the MACHO-pretrained state-dictionary sums. Duplicated time-encoding entries include a 128-element frequency buffer per extra copy. Non-trainable buffers are listed separately from distinct trainable parameters.

	S-size	L-size
Backbone, distinct trainable parameters	3,674,624	21,524,864
Gaussian pretraining head	386	770
Duplicated time-encoding entries	790,528	2,370,048
Non-trainable buffers	145	145
State-dictionary sum	4,465,683	23,895,827

C More Experimental Results on ELAsTiCC

The ELAsTiCC experiments were conducted on 8 NVIDIA A100 GPUs.

C.1 Full backbone ablation results

In this ablation, *pooling* is the rule that turns encoder states into one light-curve representation: `[CLS]`, mean, or last-token pooling. We reserve *readout* for the downstream classification head.

Table 10: The five ablation axes and their candidate options for the self-supervised pretraining backbone search.

Axis	Options
Pretraining objective	Gaussian KL, simple MSE, normalized MSE, RoMAE MSE (joint flux and flux-error MSE) (Zivanovic et al., 2025)
Time encoding	RoPE (Zivanovic et al., 2025) (time only or time and channel), Fourier time (Cabrera-Vives et al., 2024)
Generation	Masked reconstruction, auto-regression
Pooling	`[CLS]` token pooling, last token pooling, mean pooling
Backbone	LayerNorm-GELU Transformer, RMSNorm-SiLU Transformer

This appendix reports the complete backbone ablation summarized in Section 4.1 and Table 10: 96 pretraining configurations (4 pretraining objectives $\times$ 3 time encodings $\times$ 4 valid generation-pooling combinations $\times$ 2 backbones), each fine-tuned and evaluated at four label budgets (1%, 5%, 10%, and 98% of the labeled training set), for 384 fine-tuning runs in total. Checkpoints and the downstream architecture were selected on validation results. The tables below report test macro-F1 at those checkpoints. The 10 degenerate runs discussed in Table 13 are excluded from the performance summaries. Not every generation and pooling option can be combined: a learned `[CLS]` summary token requires bidirectional attention and so is only defined for masked reconstruction, while last-token pooling relies on causal attention and so is only defined for autoregressive prediction. Mean pooling is defined for both. This leaves four valid combinations, masked reconstruction with `[CLS]` or mean pooling, and autoregressive prediction with last-token or mean pooling, rather than the full $2 \times 3 = 6$ cross product of the Generation and Pooling options in Table 10.

To compare backbone design choices, every ablation configuration uses the same inexpensive ATCAT-style fine-tuning protocol (Tung, 2025): the fine-tuned network predicts through its own classification head. This fixed protocol ranks the design choices under identical downstream conditions. The main ELAsTiCC results use the fine-tuning procedure in Section 3 followed by the separate readout described in Section 4.1, trained with inverse-frequency class weights to address the skewed training distribution and balanced test set. Because the downstream protocols differ, comparisons of backbone design choices should be made within the ablation.

Table 11 breaks the test-set label-efficiency results down by generation and pooling. The reported values are post-hoc maxima over the other ablation axes. Masked reconstruction with `[CLS]` pooling has the highest or tied-highest reported test maximum at every label budget for both backbones, except at 98% labels for the RMSNorm-SiLU Transformer, where mean pooling leads by 0.003. The test summaries suggest that generation and pooling have a more consistent association with performance than backbone choice.

Table 12 reports test macro-F1 averaged over configurations of the other axes. Table 11 reports the maximum within each generation-pooling combination. In these results, Gaussian KL has the highest marginal mean at every label budget, and Fourier time features have the highest marginal mean among time encodings. The marginal means for Generation and Pooling combine different partner options and should be interpreted with the joint generation-pooling table.

All 10 collapsed runs (2.6% of the 384 total) occur under normalized MSE or RoMAE MSE combined with a RoPE-based time encoding, and span both backbones and multiple generation-pooling combinations and label budgets. No collapse occurs under Gaussian KL or simple MSE, or under Fourier time features. In these runs, collapse is associated with the combination of pretraining objective and time encoding rather than with a single backbone or label budget.

Table 11: Post-hoc maximum test macro-F1 within each generation/pooling combination and label budget, for each backbone (degenerate runs excluded). These maxima summarize the completed ablation. **Bold** marks the largest test value within each backbone and label-budget column.

LayerNorm-GELU Transformer				
Generation + pooling	1%	5%	10%	98%
masked recon. + `[CLS]`	**0.605**	**0.676**	**0.684**	**0.729**
masked recon. + mean	0.593	0.664	**0.684**	0.722
autoregressive + last-token	0.578	0.640	0.662	0.711
autoregressive + mean	0.570	0.639	0.662	0.717
RMSNorm-SiLU Transformer				
Generation + pooling	1%	5%	10%	98%
masked recon. + `[CLS]`	**0.630**	**0.676**	**0.688**	0.725
masked recon. + mean	0.604	0.658	0.685	**0.728**
autoregressive + last-token	0.582	0.638	0.658	0.710
autoregressive + mean	0.569	0.644	0.665	0.713

Table 12: Post-hoc marginal test macro-F1 of each axis option, averaged over all other axes, at each label budget (degenerate runs excluded). Figures in parentheses are the standard deviation across the other axes' combinations. Components used in the downstream recipe are underlined. **Bold** marks the largest descriptive test mean within an axis and label-budget column.

Axis	Option	1%	5%	10%	98%
Pretraining objective	Gaussian KL	**0.576 (0.025)**	**0.642 (0.017)**	**0.665 (0.012)**	**0.707 (0.014)**
	simple MSE	0.556 (0.024)	0.637 (0.015)	0.661 (0.011)	0.701 (0.012)
	RoMAE MSE	0.529 (0.091)	0.619 (0.040)	0.645 (0.045)	0.701 (0.012)
	normalized MSE	0.510 (0.098)	0.606 (0.096)	0.648 (0.019)	0.682 (0.075)
Time encoding	Fourier time	**0.556 (0.024)**	**0.636 (0.013)**	**0.665 (0.011)**	**0.713 (0.008)**
	RoPE (time only)	0.528 (0.106)	0.625 (0.036)	0.647 (0.040)	0.696 (0.013)
	RoPE (time and channel)	0.546 (0.061)	0.618 (0.086)	0.653 (0.017)	0.684 (0.063)
Generation	autoregressive	**0.552 (0.017)**	**0.629 (0.006)**	0.645 (0.005)	0.697 (0.009)
	masked reconstruction	0.535 (0.101)	0.623 (0.077)	**0.655 (0.038)**	**0.699 (0.056)**
Pooling	`[CLS]`	0.542 (0.104)	0.622 (0.099)	**0.664 (0.021)**	**0.711 (0.015)**
	last-token	**0.558 (0.014)**	**0.630 (0.006)**	0.654 (0.005)	0.696 (0.009)
	mean	0.536 (0.070)	0.627 (0.032)	0.651 (0.034)	0.692 (0.054)
Backbone	RMSNorm-SiLU	**0.550 (0.060)**	**0.633 (0.020)**	0.654 (0.034)	0.696 (0.054)
	LayerNorm-GELU	0.537 (0.082)	0.620 (0.073)	**0.656 (0.017)**	**0.699 (0.015)**

The selected downstream recipe uses Gaussian KL, Fourier time features, masked reconstruction with `[CLS]` pooling, and a LayerNorm-GELU Transformer.

C.2 Hand-crafted-feature XGBoost baselines

The light-curve-only baseline uses the ATAT-released ALeRCE features (Cabrera-Vives et al., 2024), computed over each complete light curve. We use these features as released rather than recomputing them. Removing Milky Way extinction and heliocentric redshift, which are catalog quantities, leaves 427 light-curve features. Missing values are handled natively by XGBoost.

We train the 20-class softmax model with sample weights inversely proportional to class frequency, as for the neural readouts. The selected XGBoost configuration uses the histogram method, a learning rate of 0.1, maximum depth 9, row and column subsampling of 0.8, minimum child weight 50, and L2 penalty 1.0. We grow at most 3,000 trees, with early stopping after 100 rounds on a class-balanced subset of the validation split capped at 3,000 objects per class. We evaluated eight

Table 13: Runs that collapse to a degenerate single-class prediction (test macro-F1 = 0.0048), by pretraining objective and time encoding, out of 32 runs per (pretraining objective, time encoding) pair (2 backbones $\times$ 4 generation-pooling combinations $\times$ 4 label budgets).

Pretraining objective	Time encoding	Collapsed / 32
normalized MSE	RoPE (time and channel)	3
normalized MSE	RoPE (time only)	2
RoMAE MSE	RoPE (time only)	5

candidate configurations and selected the reported model by validation macro-F1. It obtains 0.739 test macro-F1. This baseline was trained with one seed, so no variance estimate is available.

For XGBoost with metadata, each of the 427 light-curve features and 85 catalog features is represented by a quantile-transformed value and a missing-value indicator, giving 1,024 input dimensions. We search a grid of 24 configurations with the same inverse-frequency class weighting and select by validation macro-F1. Each configuration uses one seed. This baseline obtains 0.843 test macro-F1.

C.3 CLASSIFICATION FROM LIGHT CURVES AND METADATA.

Table 14 reports ELAsTiCC classification when every method also uses static catalog metadata: host-galaxy photometry, photometric and spectroscopic redshift, and extinction. To incorporate this information, we map the metadata to a fixed-size vector with a small embedding network, concatenate it with the pooled light-curve representation, and train the embedding network jointly with the backbone during fine-tuning. The rest of the procedure is unchanged. All methods benefit substantially from metadata. ATAT rises from 0.627 to 0.826 and reaches 0.835 with additional engineered features. ATCAT rises from 0.718 to 0.897. The XGBoost baseline reaches 0.843 under the configuration described in Appendix C.2, again outperforming both ATAT variants. The class-description variant reaches 0.907, above fine-tuning with one-hot labels at 0.901 and with random unit vectors at 0.900. All three label-target variants use the same inference architecture, and the class-description variant surpasses every baseline in Table 14.

Table 14: Results on ELAsTiCC using both light curves and metadata. Random unit vectors are fixed targets obtained by normalizing independently sampled D-dimensional Gaussian vectors, one per class. All three label-target variants use the same light-curve and metadata projection followed by a linear readout. For our rows, values are mean $\pm$ standard deviation over eight independently seeded fine-tuning runs, each with its own fitted linear readout, starting from one shared pretrained checkpoint. Baseline $\pm$ values follow the cited papers and may summarize different sources of variation. Our parameter counts include all weights used per light curve at inference. Baseline counts follow their cited sources. The best result is shown in **bold**, and the second-best result is underlined.

Method	Input	Parameters	macro-F1
ATAT (Cabrera-Vives et al., 2024)	raw LC + metadata	1.9M	0.826 ± 0.005
ATAT (Cabrera-Vives et al., 2024)	raw LC + metadata + features	1.9M	0.835 ± 0.006
MoE-Astro (Cádiz-Leyton et al., 2025)	raw LC + metadata	4.9M	0.860 ± 0.003
ORACLE (Shah et al., 2025)	raw LC + metadata	–	0.84
ATCAT (supervised) (Tung, 2025)	raw LC + metadata	4.1M	0.897 ± 0.0004
ATCAT (pt+ft) (Tung, 2025)	raw LC + metadata	4.1M	0.897 ± 0.0004
XGBoost (Chen & Guestrin, 2016)	427 hand-crafted features + metadata	–	0.843
Ours, one-hot label	raw LC + metadata	4.64M	$\underline{0.901 \pm 0.0014}$
Ours, random unit vectors	raw LC + metadata	4.64M	0.900 ± 0.0020
Ours, class description	raw LC + metadata	4.64M	$\mathbf{0.907 \pm 0.0034}$

C.4 TOP-k RECOGNITION OF HELD-OUT CLASSES

We provide the full protocol of the held-out-class top-k recognition experiment summarized in Table 2, the per-class results, and an analysis of why uncalibrated top-1 accuracy on a held-out class is low. Each label variant was evaluated with one supervised fine-tuning seed per held-out class. The 20 classes are the units of the per-class comparisons below.

Protocol. The experiment tests whether the label embedding of a class with no labeled training examples can rank its test objects among candidate classes. All 20 folds use the same backbone checkpoint, pretrained without labels on the full ELAsTiCC training and validation pool. This pool includes unlabeled light curves from the class later held out in each fold. For each fold, we remove every object of that class from the supervised fine-tuning and validation partitions, then fine-tune on the remaining 19 classes using light curves and metadata. The held-out class has no supervised examples or prototype in the alignment batches, fine-tuning validation set, or checkpoint selection. The held-out class therefore has no class-specific supervision, while its unlabeled curves are included in self-supervised pretraining. At test time no readout is trained. Each test object is mapped to the shared space by concatenating its pooled `[CLS]` output with the metadata embedder output, then applying the light-curve projection and normalization. Each candidate label is projected into the same space, and the 20 candidates are ranked by cosine similarity. The held-out and seen classes use the same scoring rule, although their score distributions need not be calibrated to one another.

We compare class descriptions, class names, and fixed random unit vectors under this protocol. The original descriptions are the same fixed class-level texts used in the closed-set experiment. They were not revised in response to held-out-class validation or test results, and the held-out class's own text is introduced only at evaluation. The random-unit-vector arm uses one 20-dimensional Gaussian vector normalized to unit length per class, including the held-out class. This dimensionality is specific to the held-out-class experiment. The test set contains 1,000 objects per class, and each held-out class is evaluated in a single run. We also evaluate descriptions after deleting, in each fold, sentences in the 19 seen-class descriptions that explicitly name the held-out class. This control preserves the held-out label for test-time ranking while removing direct cross-mentions from the seen-class text used during alignment.

The main table reports AUROC, average rank, top-3 accuracy, and top-5 accuracy. AUROC measures how well the embedding of the held-out class ranks its own test objects above all other test objects, and an uninformative embedding gives 0.5. Average rank is the position of the held-out class among the 20 classes for its own objects, where 1 is best. Top-3 and top-5 accuracy are the fractions of held-out objects for which the held-out class appears among the first three or five candidates. All results in this subsection use light curves together with metadata, so they describe the combined input rather than the contribution of either input alone.

Description Cross-Mention Control. In 17 of the 20 folds, at least one seen-class description explicitly named the held-out class. We reran those folds after removing the sentences containing that name from the seen-class descriptions. The other three folds were unchanged. Table 15 compares the original and cleaned descriptions. The cleaned descriptions retain a strong ranking signal: AUROC 0.769 and top-5 accuracy 0.722, compared with 0.460 and 0.147 for the random-unit-vector control in Table 2. The original descriptions have higher point estimates, but this control shows that some of their advantage came from information about the held-out class in the descriptions used during alignment.

Table 15: Effect of removing explicit mentions of the held-out class from the 19 seen-class descriptions before fine-tuning. Results are averaged over 20 held-out classes, with one fine-tuning seed per fold. The original class-description row reproduces Table 2. That table also reports the class-name and random-unit-vector controls.

Label target	AUROC ↑	Average rank ↓	Acc@3 ↑	Acc@5 ↑
Original class description	0.780	4.49	0.559	0.756
Cleaned class description	0.769	4.71	0.513	0.722

The class-name and class-description variants in Table 2 use the same alignment method and differ only in the label text. Across the 20 held-out classes, full descriptions yield higher AUROC than names for 12 classes (60%), a better average rank for 13 (65%), and higher top-3 accuracy for 13 (65%). The mean paired difference between descriptions and names is $+0.009 \pm 0.023$ AUROC and $+0.094 \pm 0.063$ top-3 accuracy, where $\pm$ is the standard error across held-out classes. After cleaning, the corresponding differences are -0.002 ± 0.025 and $+0.047 \pm 0.067$. Among the 17 changed folds, cleaning reduces top-3 by 0.055 ± 0.028 and top-5 by 0.040 ± 0.020. The per-class gains show where descriptions help, while these mean differences do not establish a uniform advantage over names. Both class names and cleaned descriptions rank candidates much better than random unit vectors.

Table 16: Per-class top-k recognition of classes held out from supervised fine-tuning on ELAsTiCC. Each row is a separate run in which the held-out class is scored using its label embedding.

	Random unit vectors			Class name			Class description		
Held-out class	AUROC	Rank	Acc@3	AUROC	Rank	Acc@3	AUROC	Rank	Acc@3
AGN	0.705	11.68	0.006	0.855	5.47	0.460	0.887	3.82	0.607
CART	0.865	6.05	0.173	0.833	5.20	0.391	0.862	3.38	0.583
Cepheid	0.053	17.29	0.000	0.938	5.49	0.091	0.785	6.42	0.018
δ Scuti	0.417	13.68	0.000	0.960	2.36	0.987	0.948	2.29	0.998
Dwarf nova	0.735	7.68	0.029	0.607	7.04	0.114	0.732	6.72	0.010
EB	0.361	15.61	0.000	0.943	2.73	0.934	0.948	4.02	0.468
ILOT	0.659	10.69	0.000	0.664	7.26	0.071	0.859	3.05	0.735
KN	0.608	11.24	0.014	0.838	5.50	0.204	0.941	3.09	0.698
M-dwarf flare	0.195	18.02	0.000	0.906	3.83	0.649	0.766	4.66	0.376
PISN	0.630	4.19	0.413	0.695	2.68	0.843	0.721	2.52	0.878
RR Lyrae	0.405	10.80	0.187	0.957	2.23	0.939	0.915	2.59	0.998
SLSN	0.353	9.41	0.108	0.523	5.52	0.169	0.697	3.00	0.890
91bg	0.409	16.15	0.000	0.926	3.00	0.723	0.872	2.22	0.951
Ia	0.209	17.43	0.000	0.836	3.21	0.528	0.816	3.09	0.584
Iax	0.572	12.20	0.010	0.866	4.79	0.355	0.878	4.00	0.494
Ib/c	0.686	6.16	0.096	0.830	3.04	0.781	0.850	3.06	0.755
II	0.491	13.46	0.062	0.838	3.13	0.757	0.820	3.31	0.711
SN-like/other	0.122	18.64	0.000	0.599	4.56	0.281	0.634	3.70	0.418
TDE	0.452	12.43	0.000	0.525	7.01	0.030	0.302	13.09	0.007
μLens	0.263	14.90	0.000	0.282	14.36	0.000	0.365	11.81	0.002
Mean	0.460	12.39	0.055	0.771	4.92	0.465	0.780	4.49	0.559

Per-Class Results Table 16 shows a text-based ranking signal for many classes. With class descriptions, 16 of the 20 held-out classes reach an AUROC of at least 0.7, and the held-out class is ranked among the top three for more than half of its objects on average. Random unit vectors give substantially weaker rankings: their AUROC is below 0.5 for 12 classes and their average rank of 12.4 is worse than the 10.5 of an uninformative ordering.

Class descriptions and class names are close on average, and the per-class results show heterogeneous effects. The full descriptions have higher AUROC for ILOT (0.859 versus 0.664), KN (0.941 versus 0.838), and SLSN (0.697 versus 0.523). ILOT is not named in any other class description, so its gain cannot be explained by the cross-mention removed in Table 15. Class names have higher AUROC for Cepheids (0.938 versus 0.785) and M-dwarf flares (0.906 versus 0.766). The Cepheid description devotes substantial space to distinctions from other periodic classes, and its mean similarity to its own held-out objects is 0.196, versus 0.337 for the class name. TDE and μLens remain difficult: the full-description AUROCs are 0.302 and 0.365. These cases show that more description text does not consistently improve the ranking.

Seen-Class Bias and Calibrated-Stacking Diagnostic. To investigate the low uncalibrated top-1 accuracy, we examine whether the score distributions favor seen classes. For every leave-one-class-out run we store the cosine similarity between each test object and each of the 20 class embeddings. For objects of the held-out class whose top-1 prediction is wrong, we measure the fraction assigned to

a seen class of the same astrophysical family, using the five coarse ELAsTiCC families of supernova-like, fast, long, periodic, and non-periodic sources. We compare it with the share of same-family classes among the 19 seen classes, which is the fraction expected if errors were spread evenly. We compare the mean similarity of held-out objects to their own label with the corresponding mean for seen objects, and report the median margin by which the held-out class loses top-1. As a post-hoc calibrated-stacking diagnostic (Chao et al., 2016), we subtract a constant γ from every seen-class score, $s'_k(x) = s_k(x) - \gamma \mathbb{1}[k \in \mathcal{S}]$, where $\mathcal{S}$ is the set of 19 seen classes, and recompute held-out and seen-class top-1 accuracy. For the evaluated class c, γ is chosen using validation scores from the other 19 held-out runs after excluding class-c objects from those validation sets. Neither the class-c test set nor its validation objects select the offset. The other models may nevertheless have trained on class c, so this cross-run selection remains a diagnostic rather than a deployable calibration protocol for classes without labeled examples.

Table 17: Diagnosis of low uncalibrated top-1 accuracy on held-out classes. Values are averaged over the 20 held-out runs. Same-family rate is the fraction of misassigned held-out objects assigned to a seen class of the same astrophysical family. The expected rate is the share of such classes among the 19 seen classes. Post-hoc calibrated stacking (Chao et al., 2016) subtracts one constant from all seen-class scores. The harmonic mean is computed within each run before averaging, so it need not equal the harmonic mean of the displayed mean accuracies.

	Random unit vectors	Class name	Class description
Same-family rate of misassigned objects	0.725	0.718	0.732
Expected rate	0.205	0.205	0.205
Similarity of held-out objects to own label	0.017	0.331	0.346
Similarity of seen objects to own label	0.670	0.717	0.721
Median top-1 margin lost	0.617	0.333	0.319
Top-1 on held-out class, uncalibrated	0.000	0.031	0.035
Top-1 on held-out class, calibrated	0.591	0.838	0.815
Top-1 on seen classes, uncalibrated	0.906	0.906	0.903
Top-1 on seen classes, calibrated	0.455	0.653	0.640
Harmonic mean, uncalibrated	0.000	0.056	0.061
Harmonic mean, calibrated	0.431	0.694	0.660

Table 17 is consistent with a bias toward seen classes. Of the misassigned objects of a held-out class, 73% go to a seen class of the same family, about 3.6 times the expected rate under uniformly distributed errors. Typical cases are Cepheids assigned to δ Scuti, pair-instability supernovae to superluminous supernovae, and Type Ia supernovae to Type Ib/c. This pattern is similar across the three label types. The seen classes also score their own objects much higher than the held-out class scores its objects, with a mean similarity of 0.72 against 0.35. The held-out class loses the top-1 position by a median of about 0.32 in cosine similarity and is ranked fourth or fifth on average.

The post-hoc calibrated-stacking diagnostic raises held-out top-1 accuracy from 0.035 to 0.815 with class descriptions. It also lowers seen-class accuracy from 0.903 to 0.640, so the gain depends on a substantial seen/held-out trade-off. Class names reach 0.838 held-out top-1 and a harmonic mean of 0.694, versus 0.815 and 0.660 for full descriptions. Even so, descriptions have higher calibrated held-out top-1 accuracy for 12 of the 20 classes (60%). The paired per-class differences (description minus name) are -0.023 ± 0.037 for held-out top-1 and -0.034 ± 0.024 for the harmonic mean, where $\pm$ is the standard error. Class names carry astronomical meaning for the pretrained text encoder and therefore provide a meaningful text-derived baseline. Random unit vectors remain weaker, at 0.591 calibrated held-out top-1. The selected offsets average 0.53. We report uncalibrated top-k recognition as the primary held-out-class result.

C.5 CONTRIBUTION OF PRETRAINING UNDER CLASS-DESCRIPTION FINE-TUNING

Table 18 compares class-description fine-tuning from randomly initialized backbones and self-supervised pretrained backbones of the same architecture. Within each row, the alignment loss, readout, training budget, and data split are identical. Pretrained initialization is the only experimental difference. The pretrained values reproduce Table 1 and Table 14.

Table 18: Matched ELAsTiCC class-description fine-tuning with random initialization and self-supervised pretraining. Within each row, only the initialization differs. The readout uses the `[CLS]` output h for light curves alone and the projected joint embedding z with metadata. Values are mean $\pm$ standard deviation over eight fine-tuning and readout seeds. Pretrained values reproduce the main classification tables.

Input	Fine-tuning target	Readout	Random init.	Pretrained
LC only	Class description	linear on h	0.754 ± 0.0018	0.770 ± 0.0017
LC + metadata	Class description	linear on z	0.898 ± 0.0032	0.907 ± 0.0034

Under this matched class-description protocol, pretraining raises mean macro-F1 from 0.754 to 0.770 with light curves only and from 0.898 to 0.907 with metadata.

C.6 Visualization of Latent Representations

Figure 3 visualizes the learned representations on the ELAsTiCC test set with t-SNE. All encoders start from the same masked-reconstruction pretraining checkpoint, and each panel is an independent t-SNE fit on the same 6,000 test objects, 300 per class, colored by the 20 true classes. The top row uses light curves only and the bottom row adds metadata. The left column shows the pretrained encoder, the middle column the encoder after fine-tuning with one-hot labels, and the right column the shared embedding space of our framework after fine-tuning with class descriptions. Stars mark the label positions, which are the class directions of the one-hot classifier in the middle column and the embeddings of the class descriptions in the right column.

In the displayed t-SNE projections, classes appear more separated after supervised fine-tuning than after self-supervised pretraining. The semantic-alignment panels place many objects near the projected anchor of their own class. With metadata, several classes appear more compact and separated.

Because each panel is fitted independently, their global layouts and apparent inter-cluster distances cannot be compared directly. t-SNE can also exaggerate gaps. The figure is therefore a qualitative visualization of the learned embeddings. The classification results provide the quantitative evaluation.

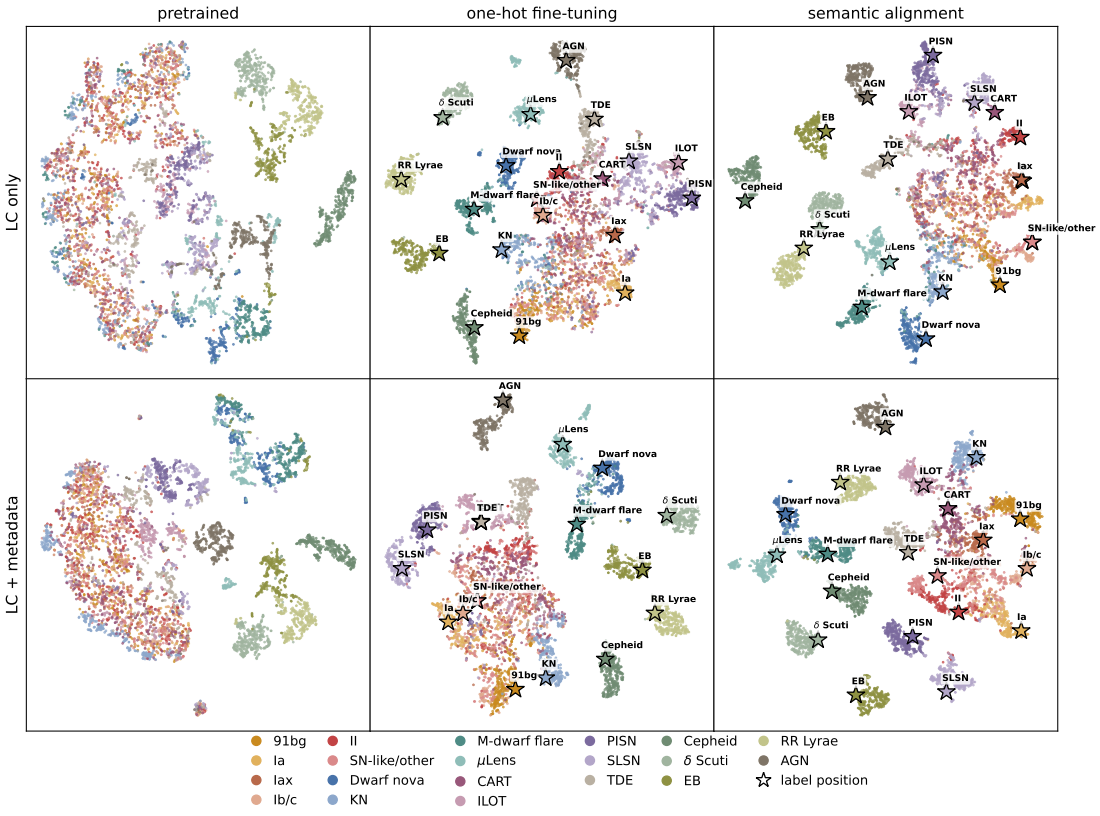


Figure 3: t-SNE visualization of light-curve representations on the ELAsTiCC test set.

D ADDITIONAL MACHO AND STAREMBED DETAILS

D.1 MACHO

Experimental Setting. We test few-shot classification on MACHO using the masked-reconstruction objective and Transformer block design selected in the ELAsTiCC ablation. The backbones start from random initialization and are pretrained on unlabeled MACHO light curves. To accommodate MACHO's longer baseline and support filters beyond pretraining, we change three input modules. The time encoding uses 128 logarithmically spaced Fourier frequencies covering periods from 0.01 to 3,000 days instead of 64 linearly spaced frequencies. A wavelength-conditioned MLP with learned band residuals replaces the band lookup, and the measurement and uncertainty embeddings use two scales instead of six. The pretraining learning rate is 2e-3 rather than 3e-4. Table 8 compares these modules with the ELAsTiCC backbone. We train S-size (192 dimensions, 8 layers, 3 heads, 3.67M backbone parameters) and L-size (384 dimensions, 12 layers, 6 heads, 21.52M backbone parameters). With the six-class linear readout used below, their inference models have 3.68M and 21.53M parameters, respectively.

We evaluate label efficiency on the labeled Alcock subset (6 usable classes after excluding UNK), using the 20, 100, and 500 labeled samples per class (SPC) and three disjoint folds of Astromer 2 (Donoso-Oliva et al., 2026). At each budget, we adapt the pretrained encoder for a fixed 1,000 steps with no validation-based early stopping, freeze it, extract its `[CLS]` outputs, and fit a linear readout on the same labeled samples. This schedule and readout type were fixed across both sizes, all label budgets, and all folds. The readout uses only the SPC-limited training samples, with no additional labels. The test set is resampled to a class-balanced 1,000 samples per class. We report the resulting exact balanced macro-F1 (an analytic expectation over this resampling, rather than a single noisy draw) as mean $\pm$ standard deviation over folds. The adaptation step is class-description fine-tuning: following the same class-aware contrastive alignment as Section 3, we fine-tune the full encoder so that each light curve's projected embedding is pulled toward the projected textual embedding of its class description, using the labeled samples available at each budget, rather than a one-hot target. Astromer 2's numbers come directly from its published results. Astromer 1's numbers come from the Astromer 2 authors' re-evaluation under their protocol (their Figure 11), since Astromer 1's own paper does not report this benchmark.

D.2 STAREMBED

Experimental Setting and Evaluation Protocol. We use the S-size backbone pretrained on MACHO, with 192 dimensions, 8 layers, 3 heads, and 3.67M backbone parameters, for the main comparison. Table 19 also includes an L-size model. We follow the benchmark's train/validation/test partitions and metric, with the preprocessing and readout details in Appendix B.2. We evaluate a linear logistic-regression probe and an MLP in the main table, with an additional random-forest readout comparison in Table 19. The linear head uses balanced class weights and is fitted on the concatenated training and validation sets. The MLP head has three hidden layers of 1,024, 512, and 256 units and is trained with inverse-frequency weighted cross-entropy on the training set. The validation set is used for model selection. Each condition uses one encoder, frozen after the specified adaptation steps. We fit logistic regression once on its `[CLS]` outputs. Given fixed embeddings, logistic regression is deterministic. We fit the MLP and random-forest readouts with eight independent seeds each and report their mean and standard deviation. These deviations measure readout-seed variation on a fixed encoder, not variation across pretraining or fine-tuning runs. We use class-balanced sampling during supervised fine-tuning to counteract StarEmbed's class imbalance. We use the benchmark's held-out test partition, excluding 13 of its 8,289 light curves that have no g- or r-band observations and therefore cannot be embedded by our preprocessing. We extract a pooled `[CLS]` output by averaging over eight temporal windows per curve, which differs from the benchmark's own per-band extraction. We report five rows for our method. The unadapted-encoder row uses the encoder as pretrained on MACHO with no encoder adaptation. Two rows fine-tune the encoder against one-hot class labels with cross-entropy, and two rows use class descriptions with the class-aware contrastive alignment from Section 3. For each target, one row fine-tunes directly and the other first continues self-supervised pretraining on StarEmbed's unlabeled training and validation curves. Unlike the ELAsTiCC target ablation, this comparison changes both the target representation and the fine-tuning loss.

Table 19: StarEmbed macro-F1 with random-forest readouts. Published baseline values and their ± estimates follow Li et al. (2026). The JEPA score follows Rui (2026) and is included as a contextual literature result under that paper's random-forest setup. Our rows use S-size unless marked L-size. Our values are mean ± standard deviation over eight random-forest seeds on one fixed encoder per row, excluding variation across encoder training runs. **Bold** and Underlined mark the best and second-best scores.

Method	RF head
Astromer 1 (Donoso-Oliva et al., 2023)	0.115 ± 0.000
Astromer 2 (Donoso-Oliva et al., 2026)	0.580 ± 0.002
Moirai-small (Woo et al., 2024)	0.557 ± 0.002
Moirai-2-small (Liu et al., 2025)	0.556 ± 0.003
Chronos-tiny (Ansari et al., 2024)	0.638 ± 0.002
Chronos-Bolt-tiny (Ansari et al., 2024)	0.582 ± 0.001
Time-MoE (Shi et al., 2025)	0.598 ± 0.003
Multi-view JEPA (Rui, 2026)	0.774
Hand-crafted features (Li et al., 2026)	$\mathbf{0.804 \pm 0.003}$
Ours, no encoder adaptation	0.390 ± 0.003
Ours, one-hot fine-tuning	0.729 ± 0.002
Ours, continued pretraining + one-hot fine-tuning	0.758 ± 0.001
Ours, class-description fine-tuning	0.739 ± 0.002
Ours, continued pretraining + class-description fine-tuning	0.770 ± 0.001
Ours (L-size), continued pretraining + class-description fine-tuning	$\underline{0.780 \pm 0.001}$

The unadapted MACHO encoder scores 0.390 ± 0.003 with a random-forest readout, below Astromer 2's 0.580 ± 0.002, reinforcing the weak direct-transfer result in Table 4. After adaptation, the L-size model reaches 0.780 ± 0.001 macro-F1, compared with the original StarEmbed hand-crafted-feature score of 0.804 ± 0.003. Multi-view JEPA reports 0.774 using its own feature extraction and random-forest setup. Its hand-crafted-feature baseline scores 0.744 rather than the original 0.804 (Rui, 2026; Li et al., 2026). The table retains the original StarEmbed feature baseline and includes JEPA's score for context.

E MORE EXPERIMENTAL RESULTS ON LEAVES

This experiment tests whether a readout fitted on one survey transfers to the other after the backbone has been aligned with class-description targets using labeled training curves from both surveys. We restrict the task to the 9 classes shared by ASAS-SN and ZTF, excluding Non-var, which is present only in the ASAS-SN portion of LEAVES. Grouping the LEAVES training and test partitions by survey yields 389,668 and 382,442 training curves and 97,524 and 95,505 test curves for ASAS-SN and ZTF, respectively. Three readout heads are fitted to the same frozen `[CLS]` output: one using ASAS-SN labels alone, one using ZTF labels alone, and one using both. Each head is then evaluated on each survey's test subset. The readout is an MLP with hidden widths of 1,024, 512, and 256, trained with unweighted cross-entropy at natural class frequencies. Its training epoch is selected by weighted F1 on a validation subset of the readout's training data. Every entry is the mean over three head seeds. We run the experiment for both reported backbone sizes.

A readout trained without labels from the survey on which it is evaluated loses between 0.0016 and 0.0055 weighted F1 relative to a readout trained on that survey, for both backbone sizes. In weighted F1, a readout trained on both surveys matches the ASAS-SN same-survey reference to within 0.0001 and exceeds the ZTF reference by 0.0003–0.0006. The comparison should be made within each test-survey column of Table 20. The ZTF subset has roughly half the median number of observations per light curve as ASAS-SN. These results describe readout transfer across the two included surveys.

Table 20: Cross-survey readout transfer on LEAVES over the 9 classes shared by ASAS-SN and ZTF. The backbone was aligned using labeled training curves from both surveys before it was frozen. Within a block, rows differ only in the survey labels used to train the readout head. Underlined numbers are same-survey references. The other entries in the two single-survey rows use a readout trained without labels from the test survey. Entries are means over 3 head seeds.

		ASAS-SN		ZTF	
Model	Readout training labels	weighted F1	macro-F1	weighted F1	macro-F1
L-size (22.58M)	ASAS-SN only	0.9826	0.9529	0.9083	0.8693
	ZTF only	0.9810	0.9467	0.9138	0.8793
	both	0.9825	0.9523	0.9144	0.8792
S-size (4.53M)	ASAS-SN only	0.9797	0.9451	0.9053	0.8575
	ZTF only	0.9776	0.9403	0.9077	0.8635
	both	0.9797	0.9449	0.9080	0.8661

Table 21 gives a separate per-class analysis for one fixed adapted encoder of each size, averaged over three readout seeds. Both models reach F1 above 0.97 for several classes, including SR, RRAB, M, and Non-var. The hardest classes are the rarer pulsators CEP, RRC, and DSCT, for which recall is lower than precision. The S-size model trails the L-size model by about 0.01 in macro-F1, reaching 0.910 against 0.920, with roughly five times fewer parameters.

Table 21: Per-class diagnostic on the LEAVES 10-class task using one fixed adapted encoder of each size. The summary rows apply macro or support-weighted averaging to precision, recall, and F1. Values with $\pm$ are mean $\pm$ standard deviation over three independently fitted MLP readouts for that encoder. These are separate from the five independent fine-tuning runs summarized in Table 5.

Model	Class	No. of samples	Precision	Recall	F1
L-size (22.58M)	EA	15,471	0.9327	0.9135	0.9230 ± 0.0008
	EW	67,375	0.9345	0.9482	0.9413 ± 0.0001
	ROT	23,822	0.8856	0.8657	0.8755 ± 0.0007
	RRAB	8,365	0.9694	0.9779	0.9736 ± 0.0002
	RRC	3,741	0.8843	0.8289	0.8557 ± 0.0021
	CEP	491	0.8707	0.8140	0.8412 ± 0.0032
	M	3,834	0.9632	0.9808	0.9719 ± 0.0004
	SR	66,210	0.9841	0.9872	0.9857 ± 0.0001
	DSCT	3,720	0.8743	0.8390	0.8561 ± 0.0008
	Non-var	26,918	0.9752	0.9727	0.9739 ± 0.0002
	macro avg.	219,947	0.9274 ± 0.0028	0.9128 ± 0.0023	0.9198 ± 0.0004
	weighted avg.	219,947	0.9488 ± 0.0002	0.9491 ± 0.0001	0.9489 ± 0.0002
S-size (4.53M)	EA	15,471	0.9264	0.9098	0.9180 ± 0.0017
	EW	67,375	0.9299	0.9449	0.9374 ± 0.0005
	ROT	23,822	0.8772	0.8529	0.8648 ± 0.0006
	RRAB	8,365	0.9693	0.9698	0.9695 ± 0.0006
	RRC	3,741	0.8771	0.8010	0.8373 ± 0.0011
	CEP	491	0.8645	0.7637	0.8104 ± 0.0058
	M	3,834	0.9629	0.9784	0.9705 ± 0.0005
	SR	66,210	0.9820	0.9863	0.9841 ± 0.0002
	DSCT	3,720	0.8421	0.8218	0.8313 ± 0.0061
	Non-var	26,918	0.9721	0.9705	0.9713 ± 0.0006
	macro avg.	219,947	0.9203 ± 0.0016	0.8999 ± 0.0012	0.9095 ± 0.0016
	weighted avg.	219,947	0.9443 ± 0.0004	0.9446 ± 0.0005	0.9444 ± 0.0005

F REPRESENTATIVE LIGHT CURVES

F.1 ELASTICC

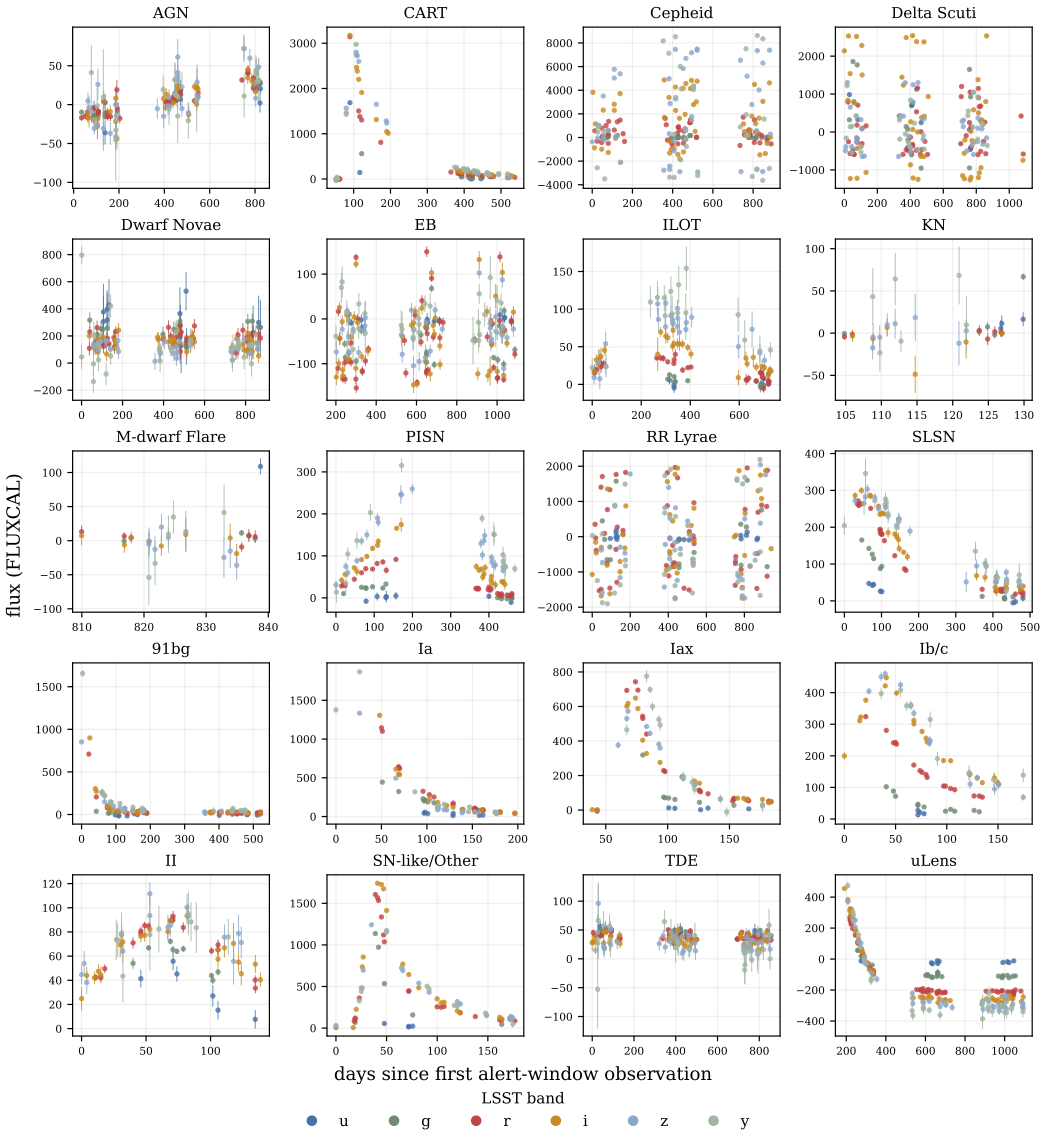


Figure 4: Representative light curves on ELAsTiCC dataset

F.2 MACHO

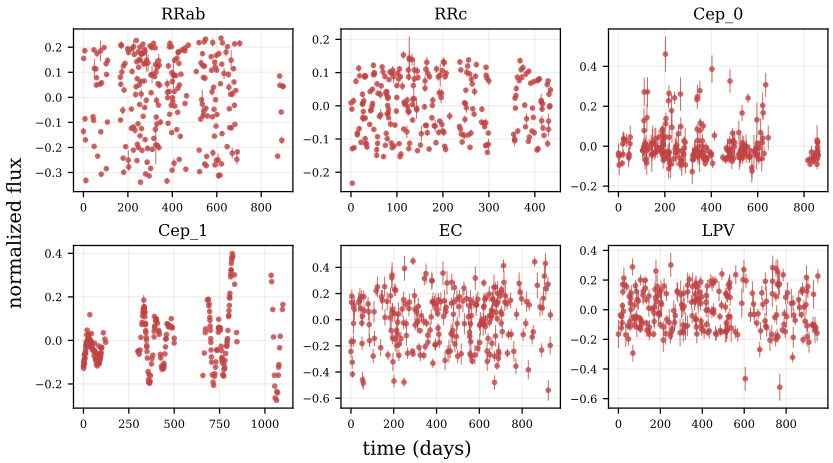


Figure 5: Representative light curves on MACHO dataset

F.3 ZTF

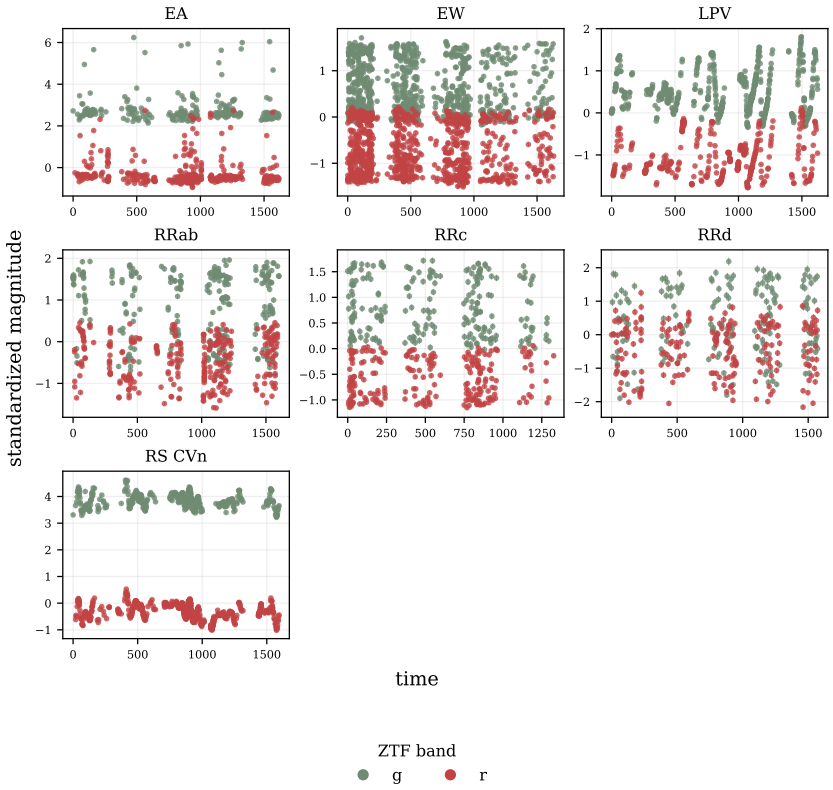


Figure 6: Representative light curves on ZTF dataset

F.4 LEAVES

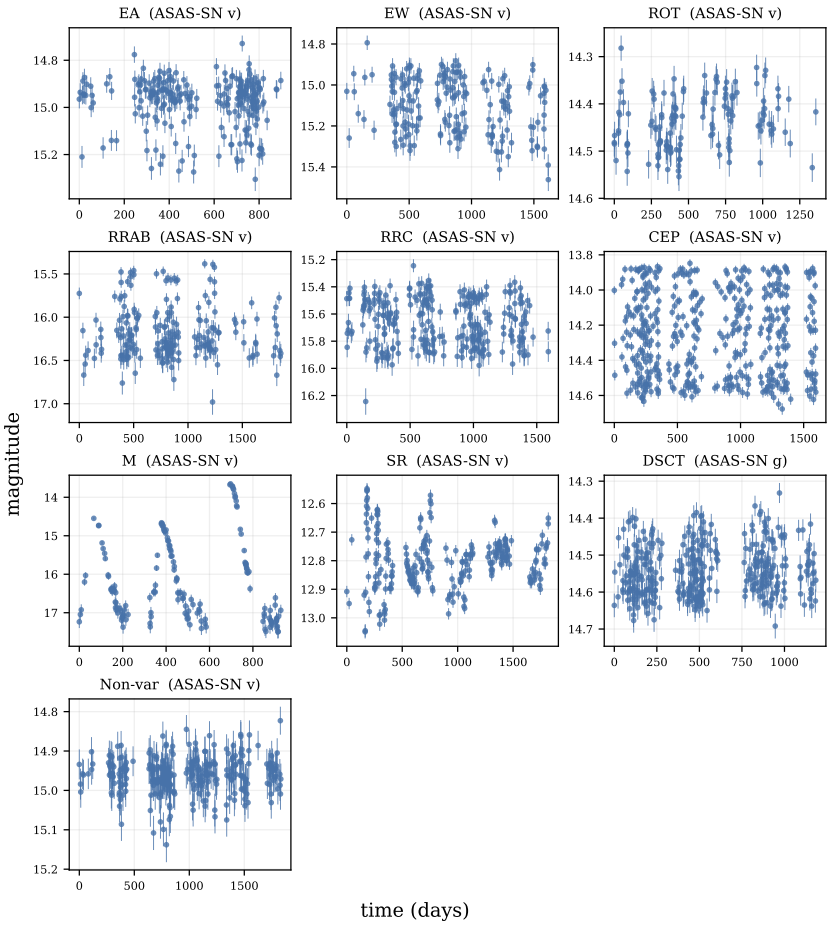


Figure 7: Representative light curves on LEAVES dataset

G EXAMPLES OF CLASS DESCRIPTIONS

The descriptions below are reproduced as model inputs, with their original wording and punctuation retained.

EA Detached Algol-type eclipsing binary [EA]

Description: An EA object is a binary system containing two stars orbiting one another, with the orbital plane oriented so that one periodically passes in front of the other from our viewpoint. In an EA system, the stars are detached, i.e. they are physically separate and do not share their outer envelopes.

Properties: Light-curve shape: Mostly flat with distinct eclipse dips. Primary eclipse: Usually deeper. Secondary eclipse: Often shallower, sometimes weak or absent. Between eclipses: Nearly constant brightness. Period: Depends on orbital period. Key feature: Flat sections + isolated sharp dips.

Because the components are detached, the brightness is typically nearly constant outside the eclipses, with distinct dips when one star passes in front of the other. Compared with contact binaries such as EW, EA light curves generally have flatter intervals between eclipses and more clearly defined eclipse boundaries.

EW W Ursae Majoris-type contact eclipsing binary [EW]

Description: An EW object is an eclipsing binary where its two stars orbit extremely close together and are in, or very close to, physical contact. Their shapes are distorted by their mutual gravity, so the amount of emitting surface visible to us changes continuously as the system revolves.

Properties: Light-curve shape: Continuous wave-like variation. Eclipses: Usually two minima per orbital cycle. Minima: Often similar in depth. Between eclipses: No long flat sections. Period: Usually short, often less than one day. Key feature: Smooth double-wave pattern with continuous variation.

They are typically mistaken for RRC. For EW vs EA, EW does not have a flat baseline in its light curve, instead it looks more like a continuous rounded wave with alternating minima. EW usually has double-wave morphology associated with two eclipses per orbit, differentiating it from other sinusoidal pulsators like RRC.

ROT Rotational variable [ROT]

Description: A rotational variable changes brightness because the star's surface is not uniformly bright. The usual cause is starspots or other magnetically active regions. As the star rotates, the spotted and unspotted portions repeatedly move into and out of view, modulating the amount of light reaching us.

Properties: Light-curve shape: Broad sinusoidal or quasi-sinusoidal variation. Period: Related to rotation period. Amplitude: Usually fairly low. Regularity: Often quasi-periodic rather than perfectly periodic. Shape evolution: Common because spots change with time. Key feature: Sinusoidal-looking modulation whose shape/amplitude slowly changes.

They are typically mistaken for RRC or DSCT. To differentiate itself, the amplitude, phase or shape of ROT would change over time, differentiating it from species like RRC and DSCT.

RRAB Fundamental-mode RR Lyrae [RRAB]

Description: An RRab is an RR Lyrae star pulsating in its fundamental radial mode, which is the simplest and lowest-frequency pattern of spherical expansion and contraction in an oscillating system. RR Lyrae stars are old, relatively low-mass horizontal-branch stars whose envelopes repeatedly expand and contract.

Properties: Light-curve shape: Strongly asymmetric sawtooth. Rise: Very rapid. Decline: Much slower. Period: Usually roughly 0.3-1 day. Amplitude: Relatively large. Regularity: Very regular, though some show longer-term modulation. Key feature: Sharp rise + slow decline.

They are typically mistaken for RRC, CEP or DSCT. However, a period of RRab is sharply asymmetric while a period of RRC is smoother and more sinusoidal. They may look like CEP or DSCT but they all have different periods.

RRC First-overtone RR Lyrae [RRC]

Description: An RRC star is an RR Lyrae that pulsates predominantly in the first overtone (second lowest frequency pattern of spherical oscillation directly above the fundamental mode). RR Lyrae stars are old, relatively low-mass horizontal-branch stars whose envelopes repeatedly expand and contract.

Properties: Light-curve shape: Smooth and almost sinusoidal. Period: 0.16-0.5 days Amplitude: Lower than RRAB. Symmetry: Much more symmetric than RRAB. Key feature: Short-period, smooth sinusoid.

They are typically mistaken for DSCT, ROT, EW and Cepheids. See above for how to differentiate.

CEP Cepheid [CEP]

Description: A Cepheid is an intrinsically variable star whose brightness changes because the star expands and contracts repeatedly. As it pulsates, its radius and temperature change, producing a very regular light curve. Cepheids are very important in astronomy because their pulsation period is related to their intrinsic luminosity.

Properties: Light-curve shape: Usually smooth and periodic. It is often asymmetric. Typical appearance: Faster rise to maximum brightness, slower decline. Period: Generally days to tens of days. Amplitude: Moderate to large. Regularity: Very regular. Key feature: Smooth, coherent pulsation with a period usually longer than RR Lyrae or DSCT.

They are typically mistaken for RRAB, RRC or DSCT. The main thing separating CEP from RRAB is that Cepheids generally operate on longer timescales and are physically much more luminous stars. CEP and RRAB can both display sawtooth shaped light curves. Cepheids will also usually have a longer period and often a less perfectly sinusoidal curve compared with RRC. Delta Scuti oscillations are extremely rapid compared to that of Cepheids.

M Mira variable [M]

Description: A Mira variable is an evolved, cool giant star undergoing strong radial pulsations. These stars are generally on the asymptotic giant branch, near a late stage of stellar evolution, and their enormous extended envelopes expand and contract over very long timescales.

Properties: Light-curve shape: Large smooth rises and falls. Period: Hundreds of days. Amplitude: Very large. Regularity: Fairly regular Cycle-to-cycle variation: Negligible (Dominant period is still obvious) Key feature: Large-amplitude, long-period, obvious repeating cycles

They are typically mistaken for SR. M normally has a larger amplitude, clearer dominant period and more coherent repeating pattern than SR.

SR Semiregular variable [SR]

Description: A semiregular variable is usually an evolved giant or supergiant whose outer layers pulsate, but not regularly, unlike Cepheids or RR Lyrae. They are generally cool, evolved stars, and their pulsation behaviour can involve changing amplitudes, changing dominant periods and several competing pulsation modes.

Properties: Light-curve shape: Slow rises and falls with imperfect repetition. Period: Tens to hundreds of days, sometimes longer. Amplitude: Usually lower than Mira variables. Regularity: Only partly regular. Cycle-to-cycle variation: Common. Key feature: Long-period variability that does not repeat perfectly.

They are typically mistaken for M. Compared to M, SR usually has smaller amplitudes and noticeably poorer cycle-to-cycle regularity. Mira usually has stronger and much more obvious repeated cycles.

DSCT Delta Scuti [DSCT]

Description: A Delta Scuti star is a pulsating star, usually an A- or F-type star, whose outer layers oscillate in radial, non-radial, or multiple simultaneous modes through an opacity valve, hence it could fluctuate differently at different areas in the star. The variability comes from the star itself rather than from an eclipse or rotation.

Properties: Light-curve shape: Often sinusoidal. High-amplitude examples can be asymmetric. Period: Extremely short, ranging from less than an hour to just a few hours Amplitude: Usually small, but HADS objects can have large amplitudes. Regularity: Generally periodic, though multiple pulsation modes can make it look complicated. Key feature: Very fast variability.

They are typically mistaken for RRAB and RRC. Both DSCT and RRC look like rapid sinusoidal oscillations, but DSCT is generally considerably faster. High-amplitude Delta Scuti stars look like miniature RRab stars, but they have a much faster period.

Non-var Non-variable source [Non-var]

Description: Non-var contains objects that have little to no significant intrinsic stellar variability.

Properties: Light-curve shape: Roughly flat. Variation present: Mostly measurement noise. Periodicity: None. Amplitude: Very low. Key feature: No coherent repeating structure.