ASTROCO: Self-Supervised Conformer-Style Transformers for Light-Curve Embeddings

Antony Tan[1,2*], **Pavlos Protopapas**[1], **M. Cádiz-Leyton**[3,4], **Guillermo Cabrera-Vives**[3,4,5,6], **C. Donoso-Oliva**[4,6], **I. Becker**[5,6,7]

[1] John A. Paulson School of Engineering and Applied Sciences, Harvard University, USA
[2] Department of Biostatistics, Harvard T.H. Chan School of Public Health, USA
[3] Department of Computer Science, Universidad de Concepción, Chile
[4] Center for Data and Artificial Intelligence, Universidad de Concepción, Chile
[5] Millennium Institute of Astrophysics (MAS), Santiago, Chile
[6] Millennium Nucleus on Young Exoplanets and their Moons (YEMS), Chile
[7] Department of Computer Science, Pontificia Universidad Católica de Chile, Santiago, Chile

Abstract

We present AstroCo, a Conformer-style encoder for irregular stellar light curves. By combining attention with depthwise convolutions and gating, AstroCo captures both global dependencies and local features. On MACHO R-band, ASTROCO outperforms Astromer v1/v2, yielding 70%/61% lower error and a relative macro-F_1 gain of $\sim$7%, while producing embeddings that transfer effectively to few-shot classification. These results highlight AstroCo's potential as a strong and label-efficient foundation for time-domain astronomy.

1 Introduction

Self-supervised learning has recently advanced representation learning for astronomical time series. In particular, Donoso-Oliva and collaborators introduced *Astromer* and its successor *Astromer 2* [1, 2], transformer-based models [3] that learn embeddings of stellar light curves through masked reconstruction. These approaches have demonstrated that foundation-style models can provide compact, label-efficient representations for variable star classification and related downstream tasks.

Despite their success, existing transformer encoders often treat every time step equally, limiting their ability to capture short-lived local phenomena such as dips, flares, or bursts. Moreover, they lack explicit mechanisms to regulate how noisy or distant measurements influence the overall representation. This reduces both reconstruction fidelity and downstream generalization.

To address these challenges, we introduce **ASTROCO**, a conformer-style[4] encoder that combines global self-attention with local depthwise convolutions and gating. Our design improves the balance between long-range dependency modeling and local feature extraction, while introducing adaptive mechanisms for information flow across layers. We demonstrate that ASTROCO produces more accurate reconstructions and more label-efficient embeddings than previous Astromer baselines, and transfers effectively to few-shot settings in downstream classification tasks.

2 Architecture and Training

Each encoder block in ASTROCO combines three sublayers: (i) multi-head self-attention to capture long-range dependencies, (ii) a depthwise convolutional sublayer equipped with Gated Linear Units[5]

*Correspondence: youxin_tan@hsph.harvard.edu

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

(GLUs, which use a learned gate to modulate local feature flow), and (iii) a gated feed-forward network that adaptively controls information at the global representation level. These components are tied together with residual connections and layer normalization.

For input, magnitudes and uncertainties are concatenated with time embeddings and fused through a learned projection, avoiding the scale mismatch that can arise in additive fusion (raw magnitudes and times are summed directly). To generate compact sequence embeddings, we mix features across all layers—including the input—via trainable softmax weights, followed by masked mean pooling.

On the MACHO R-band dataset, AstroCo-S (5.9M parameters; trained 11.6h on 4×A100 GPUs) and AstroCo-L (15.2M parameters; trained 1.2d on 4×H200 GPUs) both achieve lower reconstruction error and higher macro-F1 classification scores than Astromer v1/v2(5.4M parameters; trained 3d on 4×A5000 GPUs). See Appendix A for full hyperparameters and training details. Notably, the smaller model surpasses Astromer despite using fewer resources, while the larger variant leverages its capacity to set the strongest overall benchmark.

Contributions. Our design shows that (1) adding locality-aware convolutions and gating mechanisms greatly improves representation quality over purely attention-based encoders, (2) soft feature mixing across layers[6] yields better embeddings than fixed pooling strategies, and (3) the approach is resource-efficient, with the small model outperforming prior work using less training time and hardware.

To realize these improvements, AstroCo integrates attention, convolutions, and gating in a conformer-style encoder, which we outline next.

3 Method

3.1 Inputs, embeddings, and masking

We model each light curve as an irregular sequence $\{(t_i, m_i, \sigma_i)\}_{i=1}^{L}$, where t_i is the time, m_i the magnitude, and σ_i a measurement uncertainty. To encode inputs, photometric values (m_i, σ_i) are projected to a $d/2$-dimensional vector, while times t_i are mapped to $d/2$ dimensions using sinusoidal embeddings. These two representations are concatenated and fused into a d-dimensional embedding by a linear layer, followed by a GeLU activation and LayerNorm.

For self-supervised pretraining, we adopt a masked-reconstruction strategy. At each sequence, we select a target set of positions $\mathcal{M}$ ("probed positions"), covering 50% of the time steps. Among these, $\sim$30% are masked (the true values of m_i are hidden), 10% are replaced with random values, and 10% are left unchanged but still included in the loss (BERT-style) [7]. For masked or padded tokens, we replace the raw (m, t) pair with a zero placeholder before projection, ensuring that the network cannot trivially recover missing values from input leakage.

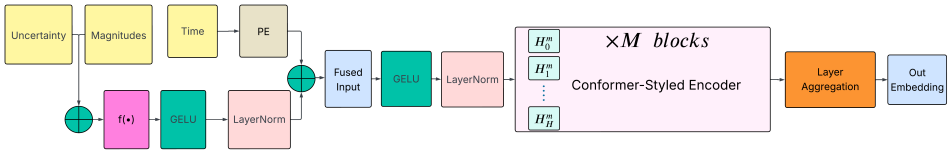

Figure 1: **AstroCo architecture.** Magnitudes and uncertainties are projected and fused with sinusoidal time embeddings, then passed through a Conformer-style encoder. Hidden states from all layers are scalar-mixed to form the final sequence embedding (AstroCo-L: $M = 12$, $H = 4$, $m = 64$, $D = 256$).

3.2 Encoder block architecture

Each encoder block (Figure 2) consists of three sublayers connected with residuals, dropout, and LayerNorm:

(1) Multi-head self-attention. Given $X \in \mathbb{R}^{B\times T\times D}$, we use standard MHSA then applied with dropout and residual followed by post-norm.

(2) Convolutional sublayer. We first normalize the input ($\hat{X} = \mathrm{LN}(X)$), then apply a pointwise 1×1 projection $D \to 2D$ followed by a Gated Linear Unit (GLU), which controls the flow of local features. Next, a depthwise Conv1D with kernel size K mixes information across time within each channel:

$$Y[b, d, t] = \sum_{k=0}^{K-1} W^{\mathrm{dw}}[d, k] \cdot X[b, d,\ t + k - \lfloor K/2 \rfloor],$$

where b indexes the batch, d the channel, and t the time step. This is followed by BatchNorm1d, a SiLU activation, and a final pointwise 1×1 projection back to D [2]. We then add the residual connection and apply LN.

(3) Gated feed-forward network (Gated FFN). With gating expansion, $r = 4$ [3], the FFN computes

$$\mathrm{val} = \mathrm{GeLU}(W_{\mathrm{val}} X), \quad \mathrm{gate} = \sigma(W_{\mathrm{gate}} X), \tag{1}$$

$$Y = X + \mathrm{Dropout}\big(W_{\mathrm{out}}(\mathrm{val} \odot \mathrm{gate})\big), \quad Y \leftarrow \mathrm{LayerNorm}(Y). \tag{2}$$

The gate adaptively modulates which global features are retained in the representation. And a residual addition and LN is applied before the encoder output.

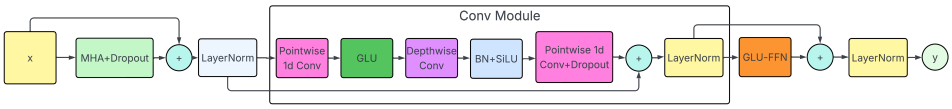

Figure 2: Encoder block consists of a MHSA layer, a depthwise convolutional sublayer with GLU gating, and a gated FFN, with residuals and LayerNorm throughout and BN + SiLU in the convolution.

3.3 Layer-wise aggregation and pooled embeddings

To obtain a single compact representation from the stacked blocks, we aggregate information across layers. Rather than relying only on the final layer, we learn a weighted combination of all intermediate representations, including the input. This allows the model to adaptively decide how much to emphasize shallow vs. deep features.

Let $x^{(0)}$ denote the input sequence and $x^{(\ell)}$ the output of block ℓ=1:M. We learn scalar mixing weights $\{w_\ell\}$, normalized with a softmax $\alpha_\ell = \frac{\exp(w_\ell)}{\sum_{j=0}^{M} \exp(w_j)}$. The final representation at position i is the weighted sum over all layers (including the input): $\tilde{x}_i = \sum_{\ell=0}^{M} \alpha_\ell\, x_i^{(\ell)}$.

3.4 Objective

Our training procedure consists of two distinct objectives, corresponding to the pretraining and downstream phases. During pretraining, the encoder is optimized to reconstruct photometric values by minimizing the root mean squared error (RMSE) between predicted and true magnitudes at the probed positions. For downstream classification, we freeze the pretrained encoder and train a linear head using cross-entropy loss (see Appendix B.3). This setup ensures that the representations learned during pretraining can be leveraged for supervised tasks.

4 Data and Setup

Unlabeled pretraining:

We pretrain on ∼1.5M single-band (R) light curves from the MACHO survey [8]. Following Astromer v2 settings [1], we first segment each series into length-200 windows and then normalize each window

[2] The $D \to 2D$ projection enables splitting into candidate values and gates, which the GLU combines as val $\odot\ \sigma$(gate). The subsequent depthwise convolution applies temporal filters per channel, capturing local patterns without cross-channel mixing.

[3] With expansion ratio r, the input is projected to two rD vectors (via W_{val}, W_{gate}). After applying gating, a linear projection W_{out} reduces back to dimension D.

Table 1: **MACHO R-band: masked reconstruction results.** Lower is better for reconstruction RMSE; higher is better for R^2.

Model	Reconstruction RMSE ↓	R^2 ↑
Astromer v1 (Donoso et al. (2023))	0.148	–
Astromer v2 (Donoso et al. (2025))	0.113	0.73
AstroCo-S	**0.060**	**0.922**
AstroCo-L	**0.044**	**0.956**

(zero-mean magnitude and time). Pretraining uses a BERT-style masking scheme [7]: we select 50% of positions as probed—of all positions, 30% are replaced by masking scalar token, 10% by a randomized value, and 10% left unchanged—and compute the RMSE reconstruction loss on the probed points.

Labeled classification:

For downstream evaluation, we use the MACHO LMC variable star catalog, which provides 20,894 labeled light curves across six broad classes [8]. To ensure label efficiency and comparability, we adopt the same labeled subset protocol as in prior Astromer work: for each class, 20, 100, or 500 instances are sampled without overlap across train, validation, and test splits. To reduce variance, we generate three independent folds for each setting, stratified by class. A linear head is then trained on top of the frozen encoder for each fold, and we report the mean macro-F_1 across folds.

5 Results

5.1 Masked reconstruction

Table 1 reports reconstruction performance on MACHO R-band. Astromer v2 improves upon v1 with an RMSE of 0.113 ($R^2 = 0.73$), but both AstroCo variants substantially surpass these baselines: AstroCo-S achieves 0.060 ($R^2 = 0.922$) and AstroCo-L 0.044 ($R^2 = 0.956$).

5.2 Downstream classification

Our frozen encoder + linear head outperforms Astromer v1/v2 on Alcock (20/100/500 shots)[8]; see Fig. 3 for exact values.

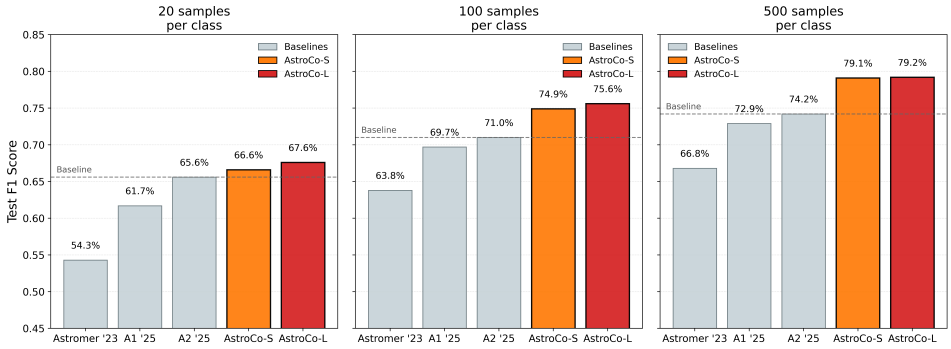

Figure 3: Downstream classification on the Macho-Labeled dataset [8]: macro-F_1 for 20/100/500 labels per class, averaged over 3 folds and 3 seeds. AstroCo-S/L (frozen encoder + linear head) outperform Astromer v1/v2, consistent with the results in Section 5.

5.3 Ablation: Effect of Convolutions and Gated FFNs

To isolate the contributions of the convolutional and gated feed-forward components, we conduct ablations of AstroCo on the MACHO-labeled Alcock benchmark. We compare the full AstroCo-S encoder against two controlled variants: (i) AstroCo-S-No-Conv, which removes the depthwise

Table 2: **Ablation on MACHO-labeled Alcock classification.** Macro-F_1 (in %) for 20/100/500 labels per class (3-fold average). AstroCo-S is the full model. Removing convolutions (AstroCo-S-No-Conv) strongly degrades few-shot performance, while replacing the gated FFN with a vanilla FFN (AstroCo-S-No-Gate) has a smaller effect. Best scores are in **bold**; lowest scores in blue with ↓.

Model Variant	20 samples	100 samples	500 samples
A2 '25 (baseline)	65.6	71.0↓	74.2↓
AstroCo-S (full)	**66.6**	**74.9**	**79.1**
AstroCo-S-No-Conv	63.90↓	73.42	78.42
AstroCo-S-No-Gate	65.64	74.16	77.82

convolutional sublayer entirely, and (ii) AstroCo-S-No-Gate, which replaces the gated FFN with a standard two-layer GeLU FFN while keeping all other components fixed. We also report the strongest prior baseline, A2 '25. Table 2 summarizes macro-F_1 performance for 20/100/500 examples per class.

Discussion. Removing convolutions has the most detrimental effect, particularly in the extreme few-shot setting (20 samples per class), where performance drops from 66.6% to 63.90%. This highlights that depthwise convolutions provide strong locality priors that improve representation quality when labeled supervision is scarce. In contrast, removing the gating mechanism has a smaller impact: AstroCo-S-No-Gate remains close to the full model for 20/100 samples and only lags by $\sim$1.3 points at 500 samples. This suggests that while gating improves robustness and global feature modulation, the convolutional sublayer is the dominant contributor to few-shot generalization.

Finally, we note that A2 '25 is trained in TensorFlow whereas our models use PyTorch, so small cross-framework discrepancies are expected. However, these differences are minor compared to the effects observed when ablating key architectural components.

6 Conclusion and Future Work

AstroCo integrates global self-attention for long-range dependencies with local depthwise convolution to capture short-term temporal patterns, enhanced by GLU-based gating and layer-wise mixing. Pretrained with masked reconstruction, it produces compact embeddings that excel in few-shot classification (20/100/500 labels per class), outperforming Astromer models in both reconstruction and classification under comparable budgets.

While our results demonstrate strong performance, the current evaluation is limited to the MACHO R-band survey and single-band settings. Broader validation on benchmarks such as PLAsTiCC or Pan-STARRS1, as well as multi-band datasets, will help assess the generality and robustness of the learned representations. These directions represent important next steps toward establishing AstroCo as a truly survey-agnostic foundation model.

Future work includes extending to multi-band fusion, scaling across surveys, and exploring tasks such as anomaly detection and period estimation. We will also release pretrained weights and embeddings, whose quality and versatility make them valuable for label-efficient transfer, benchmarking, and broad community use across diverse astronomical applications.

References

[1] Donoso-Oliva, C., et al. (2023). Astromer: A transformer-based embedding for the representation of light curves. *Astronomy & Astrophysics*, 670, A54.

[2] Donoso-Oliva, C., Becker, I., Protopapas, P., Cabrera-Vives, G., Cádiz-Leyton, M., & Moreno-Cartagena, D. (2025). Astromer 2: A foundational model for light-curve embeddings. *arXiv preprint*, arXiv:2502.02717.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. In *Proc. NeurIPS* (pp. 5998–6008).

[4] Gulati, A., et al. (2020). Conformer: Convolution-augmented transformer for speech recognition. In *Proc. Interspeech* (pp. 5036–5040).

[5] Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. In *Proc. ICML* (pp. 933–941).

[6] Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In *Proc. ACL* (pp. 4593–4601).

[7] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL-HLT*, 4171–4186.

[8] Alcock, C., et al. (2003). The MACHO project: LMC variable star inventory (Alcock et al. series, various classes). *Astrophys. J.*, 598, 597–622.

Reproducibility and Open Resources

We provide relevant resources that can be used to reproduce our results at `https://huggingface.co/AntonyT1207/AstroCo`. The repository includes pretrained AstroCo-S/L checkpoints, configuration files, embeddings, our test results, and the exact train/val/test Alcock classification data for the 3-fold MACHO-labeled benchmark. The repository may be updated over time with additional utilities, documentation, or improved checkpoints, but will continue to host the exact artifacts used in this work.

A Implementation Details

We implement AstroCo in PyTorch, where each block combines MHSA, a depthwise Conv1D with K=32 (chosen after a sweep from $K = 5$ to $K = 128$ with no meaningful gains beyond $\sim K = 32$), and a GLU-FFN with residuals and normalization. AstroCo-L uses L=12 layers, 4 heads of dimension 64 (D=256), ∼15.2M parameters, and AdamW with learning rate 7.5×10^{-5}. AstroCo-S uses L=4 layers, 4 heads of dimension 69 (D=276), ∼5.9M parameters, and learning rate 7.5×10^{-4}. Both variants use dropout 0.1, bf16, expansion ratio $r = 4$, window size 200, and batch size 256. We apply early stopping with patience of 20 epochs. And we used zero scalar as masking token.

B Additional Experimental Details

B.1 Learning curves

Figure 4 shows the learning dynamics of AstroCo-L: validation RMSE steadily decreases across epochs and closely follows the test RMSE, indicating stable convergence and good generalization.

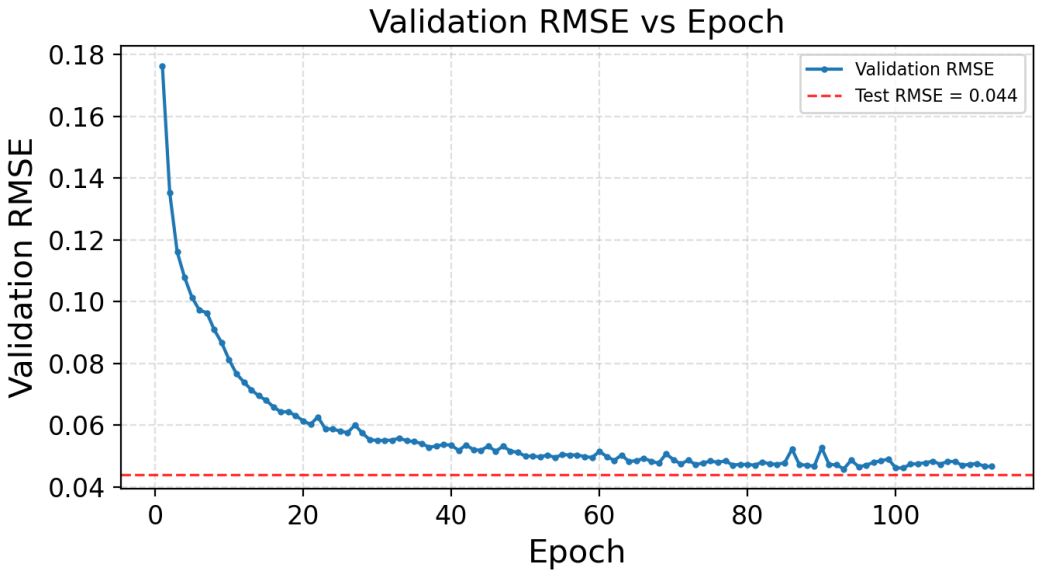

Figure 4: Validation reconstruction error over training epochs for AstroCo-L, with reference to test RMSE.

B.2 Macho-labeled data set distribution

Table 3: Class distribution in the Macho-Label dataset [8].

Tag	Class Name	# of Sources
Cep_0	Cepheid type I	1182
Cep_1	Cepheid type II	683
EC	Eclipsing binary	6824
LPV	Long period variable	3046
RRab	RR Lyrae type ab	7397
RRc	RR Lyrae type c	1762
Total		20,894

B.3 Training Objectives

Given magnitudes $\{m_i\}_{i=1}^{T}$ and probed indices $\mathcal{M} \subseteq \{1, \ldots, T\}$, the reconstruction loss is

$$\mathcal{L}_{\text{RMSE}} = \sqrt{\frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left(\hat{m}_i - m_i\right)^2}, \tag{3}$$

where $\hat{m}_i$ are predictions from the regression head. For classification, we freeze the encoder and train a linear head with cross-entropy loss.