Optimisation of LEO satellites Doppler Shift Compensation by Reducing TLE Errors and Improving Trajectory Prediction Algorithms

Authors: Rajvrat Thapliyal[1], Kushagra Shrivastava[1] **Staff Supervisor:** Assistant Professor Abhishek Rai[2] **Lab Mentors:** Ma Yuling[2], Yuan Fangxing[2], Fu Naigeng[3] **SPS Mentors:** Lim Yi Zhen[1], Hillson Hung[1]

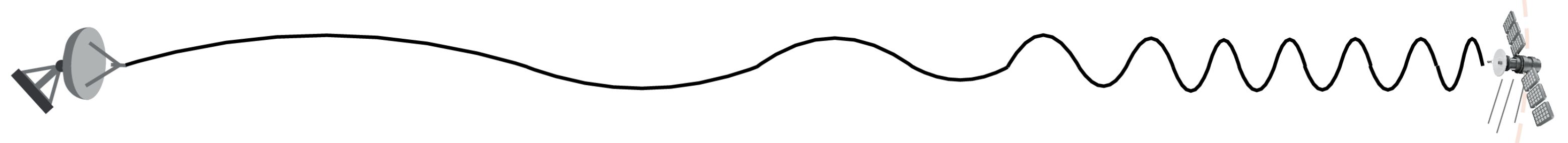
[1]Special Program in Science (SPS), National University of Singapore, Singapore [2]Satellite Technology and Research (STAR) Centre, National University of Singapore, Singapore

[3]School of Marine Science and Technology, Tianjin University, China

Introduction

Low-Earth Satellites (LEO) are used extensively for communication, internet, and disaster management[1].

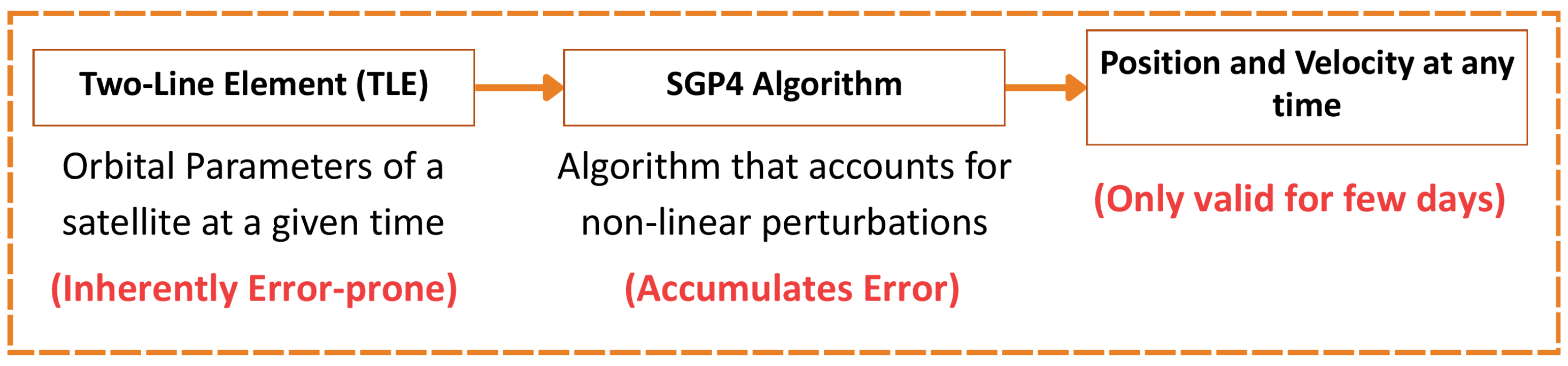
But what if we can't receive their data?!



Relative motion induces **Doppler shift**, which is highest for LEO satellites[4]! Accurate prediction of **position and velocity** paramount to account for it.

This work tests the efficacy of two algorithms to better predict position and velocity of LEO satellites.

Current prediction method[2,3]



Two-Line Element (TLE) → **SGP4 Algorithm** → **Position and Velocity at any time**

Orbital Parameters of a satellite at a given time **(Inherently Error-prone)**

Algorithm that accounts for non-linear perturbations **(Accumulates Error)**

(Only valid for few days)

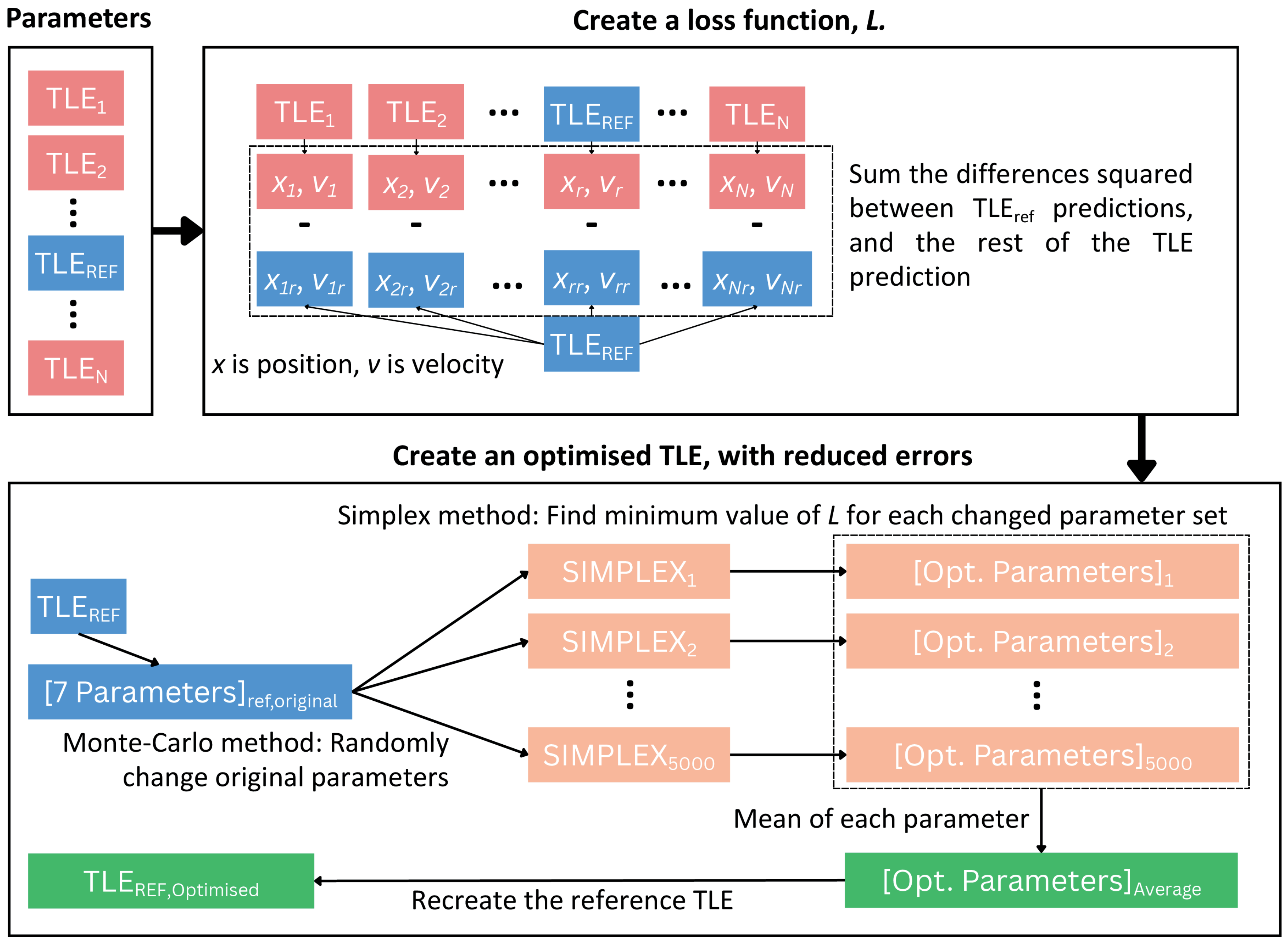
Monte-Carlo Simplex (MCS) Algorithm

Aim: Reduce the inherent error in TLEs

Original MCS Algorithm[5]: **Tailored for 1 satellite with fixed reference point**; **unable to use SGP4**

Our work generalises the MCS algorithm to function with any satellite and any reference point, while making use of SGP4.

MCS architecture and methodology



Input Parameters

TLE$_1$, TLE$_2$, ⋮, TLE$_{REF}$, ⋮, TLE$_N$

Create a loss function, *L*.

TLE$_1$, TLE$_2$ ••• TLE$_{REF}$ ••• TLE$_N$

x_1, v_1 − x_{1r}, v_{1r}; x_2, v_2 − x_{2r}, v_{2r} ••• x_r, v_r − x_{rr}, v_{rr} ••• x_N, v_N − x_{Nr}, v_{Nr}

TLE$_{REF}$

x is position, *v* is velocity

Sum the differences squared between TLE$_{ref}$ predictions, and the rest of the TLE prediction

Create an optimised TLE, with reduced errors

Simplex method: Find minimum value of *L* for each changed parameter set

TLE$_{REF}$ → [7 Parameters]$_{ref,original}$

Monte-Carlo method: Randomly change original parameters

SIMPLEX$_1$ → [Opt. Parameters]$_1$

SIMPLEX$_2$ → [Opt. Parameters]$_2$

⋮

SIMPLEX$_{5000}$ → [Opt. Parameters]$_{5000}$

Mean of each parameter → [Opt. Parameters]$_{Average}$

Recreate the reference TLE → TLE$_{REF,Optimised}$

Results and Discussion

We created 4 loss functions, and observed $L_{standard}$ ($\Delta x^2+\Delta v^2$) and L_{pos} (Δx^2) performing the same and better than $L_{weighted}$ ($w\Delta x^2+\Delta v^2$) and L_{pos} (Δv^2). **MCS algorithm converges better when magnitude of *L* is larger.** $L_{standard}$ used for all propagations.

Keeping the reference TLE constant while changing the total number of TLE, or keeping total number of TLEs while changing the reference TLE led to **MCS always performing worse than SGP4 for all initial conditions.**

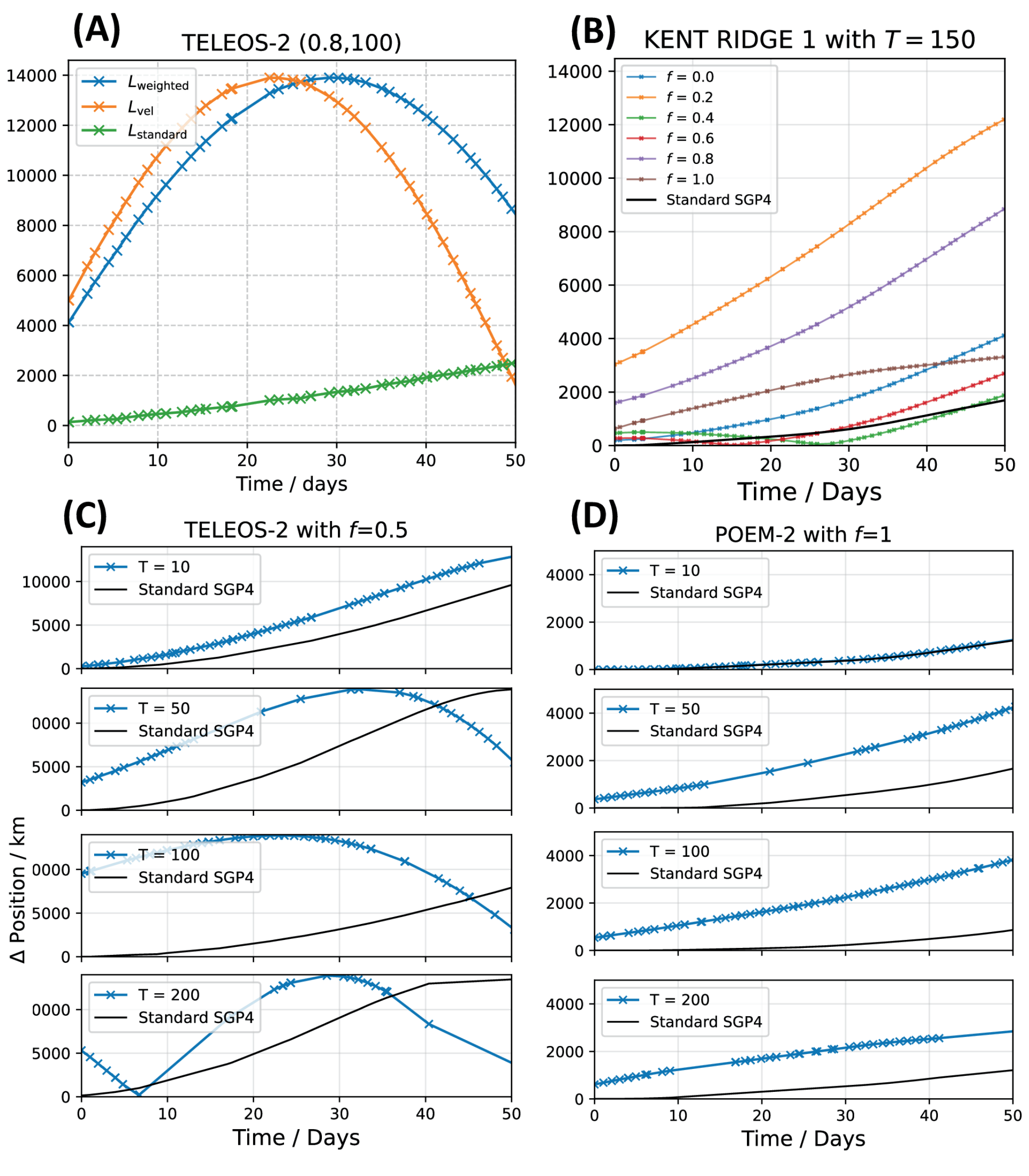


Figure 1: Only position error is shown as velocity error scales similarly. (A) Performance of different loss functions. (B) MCS performance with different reference TLE, *f*, and constant total number of TLE, *T*. (C) and (D) MCS performance with constant *f* and different *T*.

Why did MCS perform so terribly?

Hypothesis 1: MCS is physics unaware

MCS randomly perturbs orbital elements without any respect for physical laws. The orbit resembling the optimized TLE is very different from initial TLE.

Minimizing loss ≠ physically valid orbit

Hypothesis 2: TLE data set limitation

TLEs are randomly spaced in time. Optimization takes place at different points on the orbit.

If TLEs spacing is constant, optimisation may be better.

Conclusion and Future work

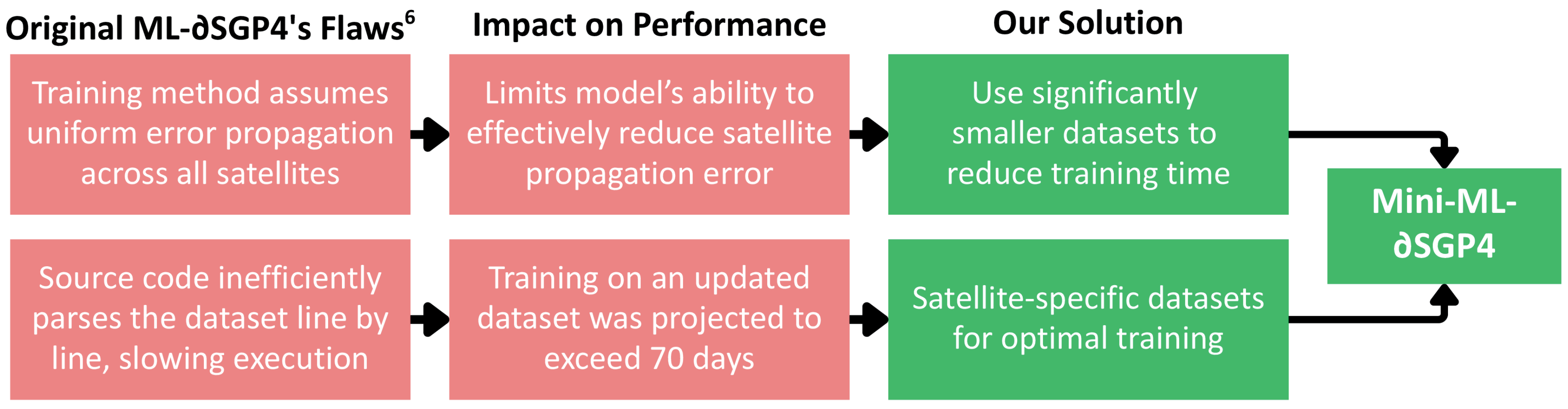
This project robustly showed MCS algorithm performs worse than SGP4 in the current setup, and Mini-ML-∂SGP4 performs well. Improvements for both should be ablation studies, where we only perturb single parameters in MCS and analyze hidden layers and their neurons in Mini ML-∂SGP4 to find exact stress points in the algorithms, which will allow for better Doppler shift calculations due to accurate position and velocity.

Acknowledgements

We express our gratitude to STAR's director and Lab Mentors for their constant support, feedback and patience, to our SPS Mentors for constant and even last minute feedback, and our SPS friends for the impromptu help. Lastly, we thank the NSCC Singapore for allowing us to use their resources.

Machine Learning differentiable SGP4 (ML-∂SGP4)

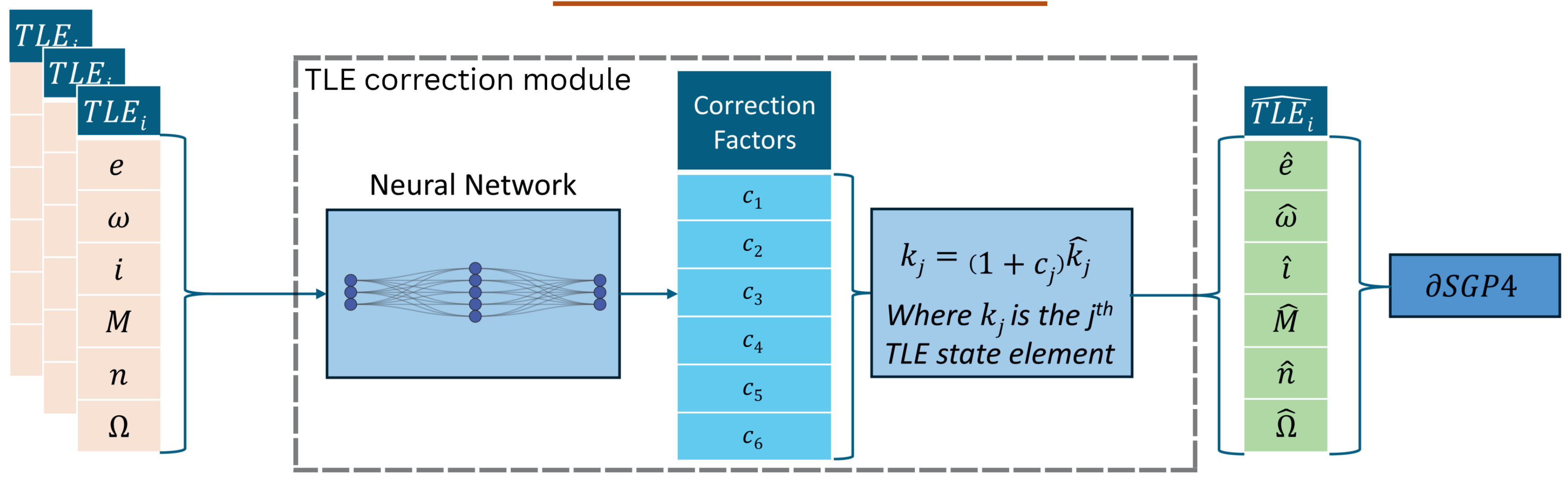
Aim: To design a more accurate and efficient model by addressing the flaws in the original ML-∂SGP4 training approach and source code



Original ML-∂SGP4's Flaws[6]	Impact on Performance	Our Solution
Training method assumes uniform error propagation across all satellites	Limits model's ability to effectively reduce satellite propagation error	Use significantly smaller datasets to reduce training time
Source code inefficiently parses the dataset line by line, slowing execution	Training on an updated dataset was projected to exceed 70 days	Satellite-specific datasets for optimal training

→ **Mini-ML-∂SGP4**

ML-∂SGP4 architecture[6]

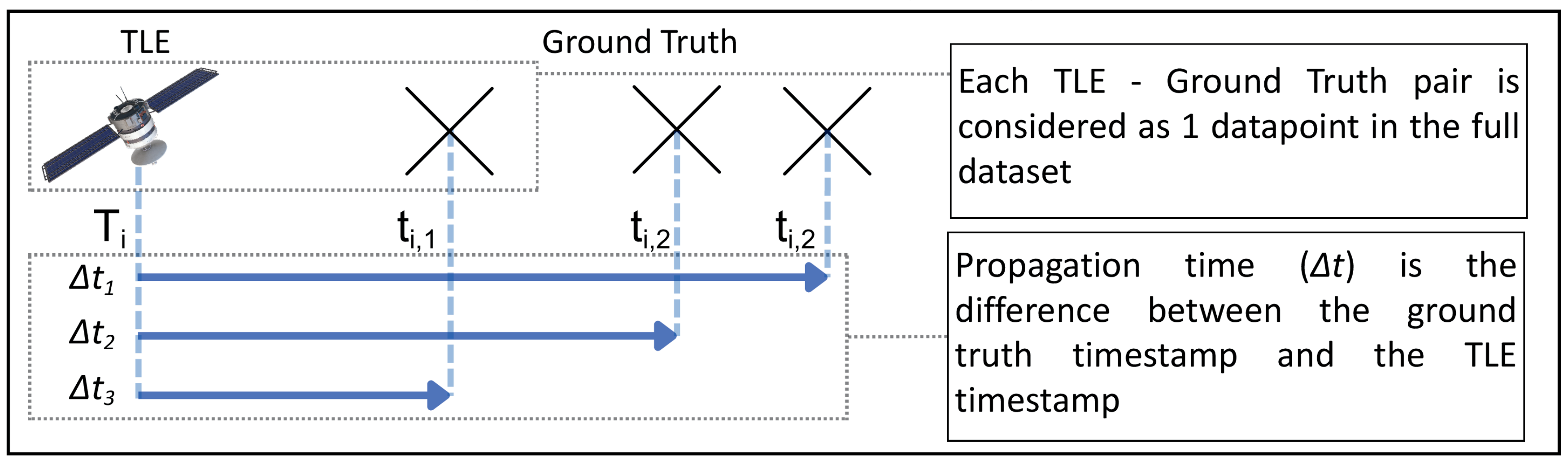


$k_j = (1 + c_j)\hat{k}_j$

Where k_j is the j^{th} TLE state element

Data Preparation

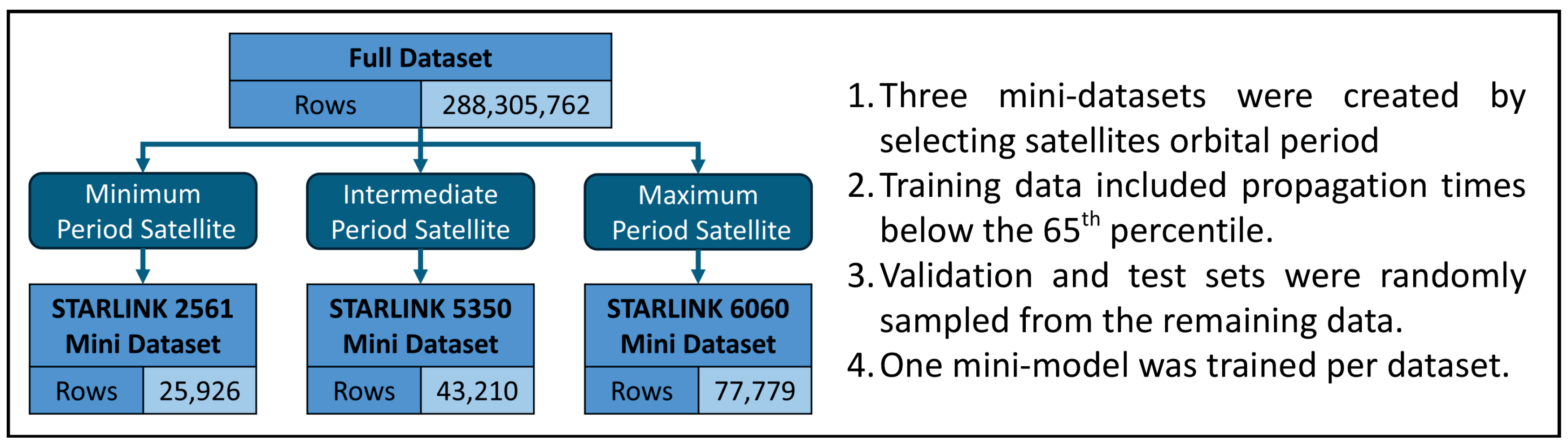
Creating the dataset and computing Propogation Time



Each TLE - Ground Truth pair is considered as 1 datapoint in the full dataset

Propagation time (*Δt*) is the difference between the ground truth timestamp and the TLE timestamp

Creating datasets for Mini-ML-∂SGP4 models



Full Dataset — Rows 288,305,762

Minimum Period Satellite	Intermediate Period Satellite	Maximum Period Satellite
STARLINK 2561 Mini Dataset	**STARLINK 5350 Mini Dataset**	**STARLINK 6060 Mini Dataset**
Rows 25,926	Rows 43,210	Rows 77,779

1. Three mini-datasets were created by selecting satellites orbital period
2. Training data included propagation times below the 65th percentile.
3. Validation and test sets were randomly sampled from the remaining data.
4. One mini-model was trained per dataset.

Results and Discussion

All three Mini-models demonstrated healthy convergence during training and completed training in under 14 minutes.

Mini-models for STARLINK-2561 and STARLINK-5350 **consistently produced lower errors** than the rival models (SGP4 and ML-∂SGP4), with a **significant reduction in error magnitude**. STARLINK-6060's mini-model did not perform as well. **Mini models trained for lower orbital period satellites performed better**

Why did Mini-ML-∂SGP4 models perform better?

Each model was trained on a single satellite, allowing it to learn that satellite's unique error propagation patterns without being influenced by the aggregated noise of a multi-satellite dataset.

The smaller datasets also circumvented source code inefficiencies, enabling rapid training.

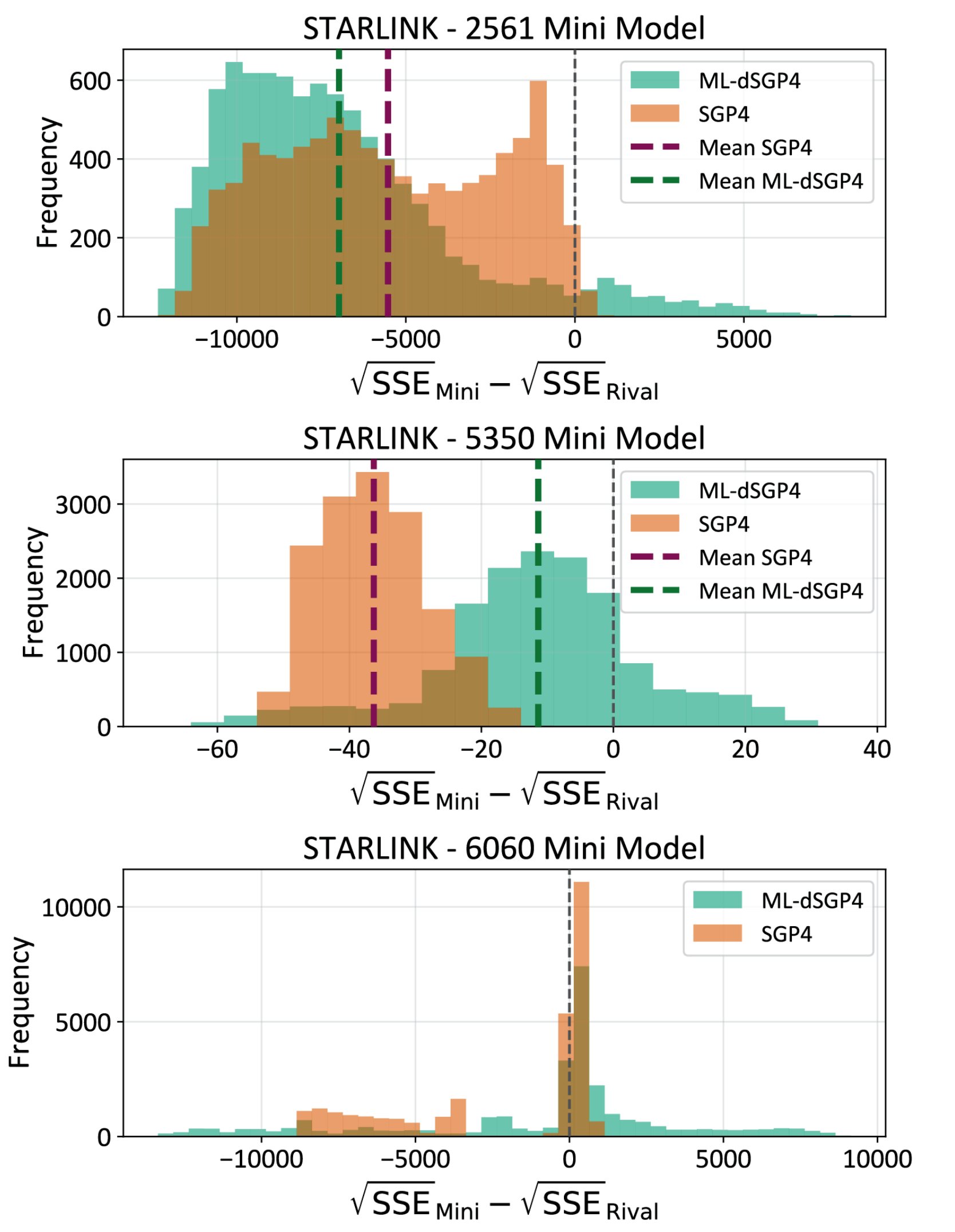


Figure 2: Histogram of error distribution for all mini-models compared against SGP4's and ML-∂SGP4's predictions.

References

[1] European Space Agency, Space Debris Office. (2024). *ESA's Annual Space Environment Report* (tech. rep. No. ESA/GEN-DB-LOG-00288-OPS-SD) (Accessed: 2025-03-27). European Space Agency. Darmstadt, Germany. https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report latest.pdf | [2] Kelso, T. S. (2022). Norad two-line element set format [Accessed: 30 March 2025]. https://celestrak.org/NORAD/documentation/tle-fmt.php | [3] Hoots, F. R., & Roehrich, R. L. (1988). *Models for propagation of norad element sets* (tech. rep. No. Spacetrack Report No. 3). CelesTrak. https://celestrak.org/NORAD/documentation/spacetrk.pdf | [4] Yll, D. N. (2018). *Doppler shift compensation strategies for leo satellite communication systems* [Ph.D. Dissertation]. Universitat Politecnica de Catalunya. https://upcommons.upc.edu/bitstream/handle/2117/123510/DanielNietoYll_Doppler_compensation_for_LEO.pdf | [5] Liu, J., Long, W., Wu, Y., Xu, J., Sang, J., & Lei, X. (2023). TLE orbit determination using simplex method. *Geodesy and Geodynamics, 14*(5), 438–455. https://doi.org/10.1016/j.geog.2023.03.004 | [6] Acciarini, G., Baydin, A. G., & Izzo, D. (2025). Closing the gap between sgp4 and high-precision propagation via differentiable programming. *Acta Astronautica, 226*, 694–701. https://doi.org/10.1016/j.actaastro.2024.10.063