A Global-Local Probsparse Self-Attention Transformer for LEO Satellite Orbit Prediction

Guan Huang and Tao Shu
Department of Computer Science and Software Engineering, Auburn University, Auburn, United States
{gzh0040, tshu}@auburn.edu

2024 International Conference on Machine Learning and Applications (ICMLA) | 979-8-3503-7488-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMLA61862.2024.00019

***Abstract*—In recent years, the proliferation of LEO (Low-Earth Orbit) satellites and the accumulation of space debris have made Near-Earth space more and more crowded, and hence significantly increased the risk of collisions in this space. As a result, precise orbit prediction becomes essential for LEO satellites to avoid collision, maintain the right constellation, and perform normal space operations. Although machine learning(ML)-based satellite orbit prediction methods have been extensively explored, most existing methods are trained on simulated/synthetic orbit data, which essentially assumes a stationary orbit process and cannot reflect the non-stationary dynamic orbit changes in real-world LEO satellite constellations that are caused by satellite flight status adjustment (e.g., for the purpose of collision avoidance). In this study, we propose a novel multi-range (global-local) self-attention transformer-based ML model, the GloLoSAT (*Global-Local Probsparse Self-Attention Transformer*), and train the model over real-world LEO satellite orbit data crawled from N2YO.com to give more precise orbit prediction in case of a series of orbit adjustments. Theoretical analysis demonstrates our Global-Local Probsparse Self-Attention mechanism can achieve $O(L\log(L))$ computational complexity with respect to the input sequence length. Extensive experiments conducted on real satellite orbit tracking datasets demonstrate the efficacy of GloLoSAT in achieving consistent performance improvements across various prediction scenarios compared to their counterparts.**

***Index Terms*—LEO Satellite, Orbit Prediction, Transformer, Long Sequence Time-series Forecasting**

I. Introduction

Traditionally, space exploration has focused on deep space discovery, such as seeking to identify new astronomical objects and unravel the secrets of the universe [1]. However, a recent shift in focus has emerged towards the practical utilization of near-earth space through low-Earth orbit (LEO) satellites [2]. These satellites, orbiting at altitudes below 2,000 km, are progressively utilized across a spectrum of applications, including terrestrial surveillance, global ubiquitous communication, cataclysm surveillance, navigation, and ecological assessment [3]. The Starlink project, for example, currently operates over 5,800 active satellites and has plans to increase this number upwards to 42,000 in the near future [4]. With the near-earth space becoming more and more crowded, the swift deployment and operation of LEO satellites raises potential risks of collision. To mitigate these risks, there is an urgent need for more precise LEO satellite orbit prediction to ensure more effective collision avoidance.

Conventionally, satellite orbit prediction approaches can be classified into two distinct categories: physics-based approaches and machine learning (ML)-based approaches. Physics-based methods, which include general perturbation methods [5], special perturbation methods [6], and semi-analytical methods [7], have been studied for many decades. These methods are based on the classical laws of motion and principles of gravity, enabling the prediction of satellite trajectories by solving differential equations of motion, either analytically or numerically [8]. However, these methods cannot consistently yield high-precision orbit predictions for LEO satellites due to the difficulty in taking into account a large number of trajectory-perturbing random factors such as environmental influences like atmospheric drag and forces of solar radiation, as well as particular satellite attributes, including mass, shape, and information regarding maneuvers. Consequently, the prediction errors can be substantial (e.g., in the order of hundreds of kilometers), potentially rendering them less effective for practical applications.

Recently, ML-based approaches have emerged as a complementary framework to traditional physics-based models in satellite orbit prediction. At a high level, these methods can be classified into two types: (1) Physics-Enhancing approaches [9], [10] and (2) Error-Correction approaches [11]–[14]. In the first type, ML-based methods are often used to enhance the representation of unmodeled random variables in physics-based models or serve as solvers for motion-related differential equations. In the second type, ML models are specifically trained to characterize prediction errors of physics-based methods. The predictions generated by the physics-based approaches are then re-tuned using the error corrections generated by the trained ML model.

Regardless of their types; however, rather than employing real orbit data for model training, existing ML-based approaches often rely on synthetic datasets that are either entirely simulated based on some space parameters such as gravity field, solar radiation, and atmosphere model [10], [13], [15] or calculated from the well-known SGP (Simplified Perturbations) models, utilizing TLEs (Two-Line Element sets) or CPFs (Consolidated Prediction Format files) [9], [11], [14]. A major limitation of these ML-based methods is that the random processes and factors simulated in their synthetic/simulated dataset are assumed to be stationary. As a result, ML models trained on such datasets fail to capture the non-stationary dynamic orbit changes in real-world LEO satellite constellations. These changes are due to necessary satellite status adjustments and maneuvers to maintain the constellation or avoid collisions, making the accuracy of these

1946-0759/24/$31.00 ©2024 IEEE
DOI 10.1109/ICMLA61862.2024.00019

ML-based methods questionable in real-world scenarios.

In this study, we propose a novel multi-range (global-local) self-attention transformer-based ML model for accurate LEO satellite orbit prediction. In contrast to existing ML-based approaches, the proposed model is trained over real-world LEO satellite orbit data and is capable of giving more precise orbit prediction in case of a series of orbit adjustments. In particular, we first propose a novel web crawler, called N2YO-crawler, designed to collect real-time satellite orbit tracking data from N2YO.com, which provides live real-time trajectory tracking data for over 28,751 objects in outer space, including satellites and space debris. The data for LEO satellites, such as the 5,814 StartLink satellites and the 628 OneWeb satellites, are updated every second on this website. N2YO.com is a highly reputable provider of real-time satellite tracking services. Notably, the space surveillance data used by N2YO.com are diligently collected by the US Space Surveillance Network (SSN), which operates under the rigorous oversight of the US Air Force Space Command (AFSPC) [16].

Secondly, trained over the real orbit tracking data collected by our N2YO-crawler, we propose a novel three-stage transformer-based framework for predicting LEO satellite trajectories, named GloLoSAT. An illustrative overview of GloLoSAT is provided in Fig. 1. To overcome the limitations of existing transformer-based LSTF (Long-sequence Time-series Forecasting) models [17]–[21] that often focus solely on either global or local context, lacking a comprehensive view of the input sequence, we introduce an innovative Global-Local Probsparse Self-Attention mechanism. This mechanism contains both local and global Probsparse self-attention modules, enabling GloLoSAT to capture crucial and non-stationary contextual information at various scales.

The **main contributions** of this work are four-fold: (i) We introduce N2YO-crawler, an efficient tool for collecting and building real-time satellite orbit tracking datasets from N2YO.com. (ii) We propose a general three-stage transformer-based framework, GloLoSAT, for predicting LEO satellite trajectories, featuring a Global-Local Probsparse Self-Attention mechanism that captures both global and local contexts to handle non-stationary orbit changes, improving prediction accuracy. (iii) We conduct a theoretical analysis of the Global-Local Probsparse Self-Attention mechanism, revealing its computational complexity to be $O(L\log(L))$, highlighting its scalability and practical application of the mechanism in tasks that involve long sequences. (iv) We conducted extensive experiments on our crawled real-world satellite (STARLINK-30506) orbit dataset, demonstrating our proposed methods significantly improve the accuracy of LEO satellite orbit prediction, consistently outperforming the counterparts on MGDE (Mean Geocentric Distance Error) by reducing prediction errors by up to 60.2% among various practical scenarios.

II. Related Work

A. ML-based Satellites Orbit Prediction

Over the past several years, ML-based approaches have been recognized as valuable tools to assist conventional physics-based approaches for satellite orbit prediction. These methods can be categorized into two distinct types: (1) Physics-Enhancing approaches. The work presented in [9] introduces a differentiable SGP4 model called δSGP4, which employs the neural networks as the solver for the motion-related differential equations within the conventional SGP4 model. Similarly, the study in [10] proposes a physics-informed machine learning algorithm that conceptualizes the orbital trajectory as partial differential equations and utilizes deep neural operators to capture the system's dynamics. (2) Error-Correction approaches. In [11], [12], ensemble learning methods based on BT (Boosting Tree) algorithms are adopted to identify patterns in the prediction errors of physics-based models, and error corrections are then applied to the physics-based orbit predictions. In [13], [14], the authors employ an SVM (Support Vector Machine) as the error correction model to learn the SGP4-based prediction errors and then refine the SGP4-based orbit prediction based on an error correction frame.

While these ML-based methods significantly contribute to orbit prediction, they rely on simulated satellite orbit datasets generated by orbital numerical simulators or simplified perturbation models. These datasets lack the full dynamic complexity of actual satellite orbits, making their errors more pronounced in the increasingly crowded LEO space with high launch rates and abundant space debris. Considering the inherent limitations of simulated datasets, one can foresee the margin of their orbit prediction error will increase progressively. Therefore, it becomes imperative to incorporate real satellite orbit tracking data into the study of LEO satellite orbit prediction to strengthen the safety and sustainability of LEO space activities.

B. Transformer-based LSTF Models

In recent years, inspired by the effectiveness of transformers in CV (Computer Vision tasks) [22] and NLP (Natural Language Processing [23], there has been increasing interest in exploring transformer-based models for time-series forecasting tasks. In Informer [20], the authors propose an innovative extension of the conventional Transformer architecture by incorporating the probabilistic sparsity function into the conventional self-attention module, termed the ProbSparse Self-Attention mechanism. This design aims to streamline the complexity inherent in standard self-attention and cross-attention mechanisms. In FEDformer [21], the authors opt for applying the Transformer in the frequency domain rather than the time domain. This strategic shift allows FEDformer to better capture the global properties of time series data. The authors of ETSformer [18] introduce a novel approach to time-series Transformer architecture by replacing the vanilla self-attention module with exponential smoothing attention and frequency attention modules, aiming to diminish time complexity. The work Informer(stack) [19] presents a multi-stack informer architecture within its encoder blocks to broaden the model's perspective. The initial stack processes the entire input sequence. Subsequent stacks then focus on progressively smaller halves, with the second stack handling the right half

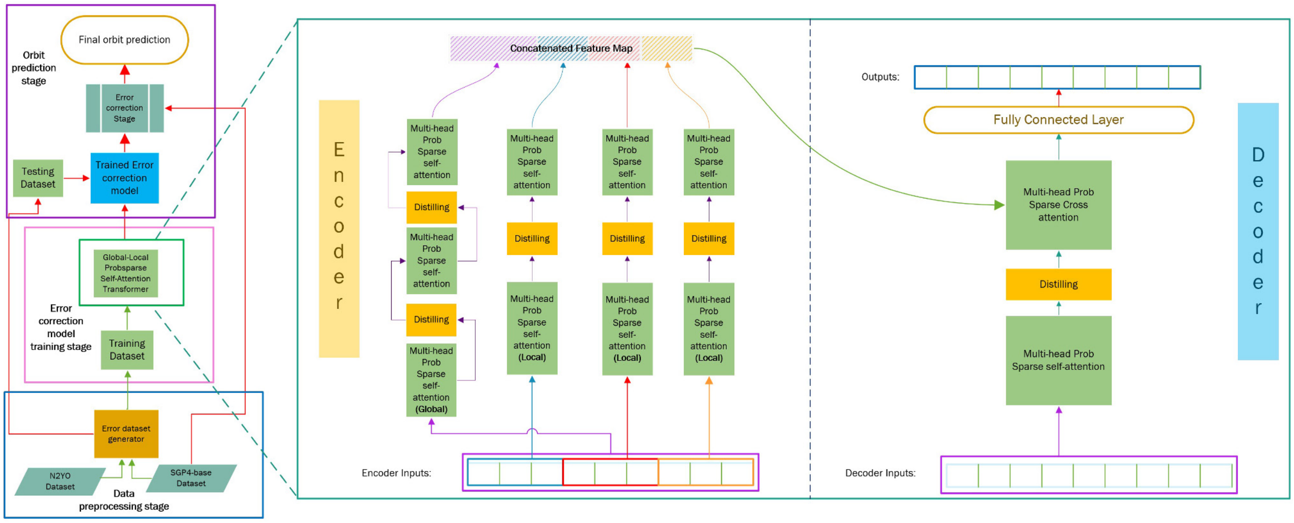
Final orbit prediction
Orbit prediction stage
Error correction Stage
Testing Dataset
Trained Error correction model
Global-Local Probsparse Self-Attention Transformer
Error correction model training stage
Training Dataset
Error dataset generator
N2YO Dataset
SGP4-base Dataset
Data preprocessing stage
Concatenated Feature Map
Encoder
Multi-head Prob Sparse self-attention
Distilling
Multi-head Prob Sparse self-attention
Distilling
Multi-head Prob Sparse self-attention (Global)
Multi-head Prob Sparse self-attention (Local)
Encoder Inputs:
Outputs:
Fully Connected Layer
Multi-head Prob Sparse Cross attention
Distilling
Multi-head Prob Sparse self-attention
Decoder
Decoder Inputs:

Fig. 1. Overview of GloLoSAT Framework. The left side illustrates the GloLoSAT framework, consisting of three stages: data preprocessing (framed in blue), error correction model training (framed in pink), and orbit prediction (framed in purple). On the right side, the error correction model architecture is depicted, featuring an encoder (left) and a decoder (right). Within the encoder, we replace the standard ProbSparse self-attention modules of vanilla Informer with our novel Global-Local Probsparse self-attention modules. A 3-layer stack (purple tensor path) captures the entire input sequence, while three 2-layer stacks (blue, red, and yellow tensor paths) individually process the left, middle, and right thirds of the input sequence. The decoder receives entire input sequences(framed in purple), processes them using attention mechanisms, and generates predictions(framed in blue) in a real-time, generative manner.

and the third stack processing the left half of the remaining sequence. The recent work [17] introduces the Fourier-Mixed Window Attention approach, named FWin-Transformer. This method segments sequences into shorter spans, applying window attention layers to these segments, followed by a Fourier layer to merge all the window attention outputs. The intent is to enhance the model's understanding of micro-level details within the input sequence.

While models like [18], [20] excel at capturing global context, their reliance on self-attention mechanisms that operate solely on the entire input sequence can overlook local contexts, fine-grained details, and short-term dependencies. This limitation may lead to inefficiencies when processing long sequences. Conversely, [17] emphasizes local spans with a local-only window approach, but this may limit the comprehensive capture of the entire input context. Additionally, [19]'s multi-stack architecture aims to balance local and global contexts but inherently overlooks the left half of the input sequence, potentially biasing self-attention modules and affecting forecasting performance.

III. DATASETS

In this section, we first introduce N2YO-crawler. We then outline the acquisition process for the Real-time satellite orbit tracking dataset (D_{Real}), TLEs dataset (D_{TLE}), and the SGP4-based orbit prediction dataset (D_{SGP4}).

A. N2YO-crawler

To dynamically interact with N2YO.com, we use Selenium as our automated web browser. We then develop a tailored URL generator for our N2YO-crawler, which accepts Satellite NORAD IDs and maps them to corresponding real-time orbit tracking URLs, eliminating the need for manual URL inputs. we reconfigure N2YO-crawler's web driver to extract data from the dynamic "tabledata" element on N2YO.com, ensuring focused data acquisition. To avoid delays and missed updates, we optimize the web driver to retrieve real-time satellite tracking data without page refreshes. To prevent data loss, we introduced a csv_writer function that transfers data from RAM to the local hard drive every second, ensuring data persistence even during large-scale crawls.

B. Real-time Satellite Orbit Tracking Dataset Collection

To construct our real orbit tracking dataset D_{Real}, we first utilize N2YO-crawler to gather data from N2YO.com over an 11-day period, from December 8, 2023, at 17:01:06 UTC to December 19, 2023, at 14:16:16 UTC, resulting in 940,511 data points. It is also important to notice that during this period, our target satellite, STARLINK-30506, executed two maneuvers: one on December 16, 2023, at 18:02:09 UTC and another on December 18, 2023, at 05:13:32 UTC. Each collected data point initially comprised 14 features, including *NORAD ID*, *Local Time*, *UTC*, *latitude*, *longitude*, *Altitude [Km]*, *Altitude [Mi]*, *Speed [Km/s]*, *Speed [Mi/s]*, *Azimuth*, *Elevation*, *Right Ascension*, *Declination*, and *Local Sidereal Time*. Noting that the values of the last five features are contingent on the data collector's IP location, and to concentrate on the satellite trajectory prediction task, we trimmed our dataset to include only five essential features: *UTC*, *Altitude [Mi]*, *latitude*, *longitude*, and *Altitude [Km]*. This data trimming was performed to ensure that the data set is specifically relevant to satellite trajectories and is independent of the geographical location of the data collector.

To further facilitate the learning process for the LSTF, we augmented each record with four additional attributes: *sintime*, *costime*, *index*, and *sinindex*. According to [24], the inclusion of these features can provide a consistent distance measure for time-related variables, capturing periodicity and removing discontinuities, thus potentially aiding the model in learning time-based patterns. Precisely, the values of *sintime* and *costime* are computed by applying the sine function to the corresponding date, represented as the number of seconds since January 1, 1970, 00:00:00 UTC. The index is a sequential identifier, commencing with 1 for the first record. The *sinindex* is derived by applying the sine function to the index.

C. Other Datasets Collection

Following the methods used in [14], we build the TLEs dataset (D_{TLE}) by collecting TLEs from the Space-Track.com website throughout December 8, 2023, to December 19, 2023, collecting 30 TLEs. We then constructed an SGP4-based orbit prediction dataset (D_{SGP4}) through a two-step process. First, we generate orbital position predictions using the SGP4 model [25] with the most recent TLEs from D_{TLE}, ensuring each prediction utilized the most current TLE for its corresponding timeframe. Second, since SGP4 outputs are in the TEME (True Equator Mean Equinox) frame [26], which differs from N2YO.com's geodetic coordinates, we use Astropy to convert TEME values to geodetic coordinates.

IV. PROPOSED METHOD: GLOLOSAT

This section defines the task, details the GloLoSAT's three stages, analyzes the computational complexity of the Global-Local Probsparse Self-Attention Mechanism, and describes architecture variants of the error correction model.

A. Task Definition

In this paper, the real orbit tracking dataset (D_{Real}), TLE dataset (D_{TLE}), and SGP4-based orbit prediction dataset (D_{SGP4}) for the satellite STARLINK-30506 are used. It is imperative to notice that the original trajectory values (such as *altitude, longitude,* and *latitude*) in D_{Real} exhibit near periodic behavior, presenting challenges for ML models because the statistical properties of these series, such as mean or variance, change over time. In order to address this learning challenge, instead of directly learning the satellite trajectories, our proposed method attempts to learn the error between the trajectories predicted by the SGP4 model and the actual trajectories provided by N2YO. This error turns out to be less periodic but more stochastic and, hence, more learnable by our ML model. Hence, our primary objective is to improve the accuracy of predicting the aforementioned errors, thereby enhancing the accuracy of orbit predictions.

B. Data Preprocessing Stage

We first introduce the notations. The datapoint in D_{Real} at a given time t_i is denoted as ${}^{GEO}X_{Real}(t_i)$, where GEO means the last three features of the datapoint are in the geodetic coordinate frame. The datapoint in D_{SGP4} at time t_i is denoted as ${}^{GEO}\widehat{X}_{SGP4}(t_i)$. We define $D_{\boldsymbol{e}}$ as the SGP4 prediction error dataset, which incorporates the positional discrepancies between D_{Real} and D_{SGP4}. Similarly, the datapoint in $D_{\boldsymbol{e}}$ at time t_i is denoted as ${}^{\text{GEO}}X_{\boldsymbol{e}}(t_i)$.

The GloLoSAT begins with a data preprocessing stage. Initially, it takes the datasets D_{Real} and D_{SGP4} as input , and subsequently constructs the SGP4 prediction error dataset $D_{\boldsymbol{e}}$ by calculating the positional discrepancies between D_{Real} and D_{SGP4}. Specifically, the computation is detailed by the following equation:

$$ {}^{GEO}X_{\boldsymbol{e}}(t_i) = {}^{GEO}X_{\text{Real}}(t_i) - {}^{GEO}\widehat{X}_{\text{SGP4}}(t_i) \quad \forall t_i \in T, \quad (1) $$

where T is the set of timestamps for which data points exist in both D_{Real} and D_{SGP4}. It is essential to recognize that both D_{Real} and D_{SGP4} encompass three geodetic attributes: $latitude$, $longitude$, and $altitude[Km]$. These common attributes are leveraged to evaluate the positional discrepancies, thereby constituting the error dataset $D_{\boldsymbol{e}}$. In particular, this error dataset $D_{\boldsymbol{e}}$ retains the structural integrity of D_{Real}, substituting the values of the three geodetic features with the respective positional differences, while the values of the remaining features remain consistent with those in D_{Real}. Following the acquisition of the error dataset, $D_{\boldsymbol{e}}$ is partitioned into distinct subsets to support the subsequent stages. Specifically, the initial 75% of $D_{\boldsymbol{e}}$ is allocated for the training dataset, 10% is reserved for the validation dataset, and the remaining 15% is dedicated to the evaluation of orbit prediction accuracy.

C. Error Correction Model Training Stage

We introduce a multi-range self-attention mechanism for the error correction model in the GloLoSAT framework, trained on dataset $D_{\boldsymbol{e}}$ to characterize SGP4-based prediction errors. This mechanism enables the model to provide a multi-scope perspective on the input sequence. Specifically, our proposed model is inspired by Informer. We modify Informer's encoder block's Probsparse self-attention module, replacing it with our Global-Local Probsparse self-attention modules.

A. Global-Local Probsparse Self-Attention Mechanism.

Our proposed Global-Local Probsparse self-attention mechanism, as depicted on the right-hand side in Fig. 1, enables the model to have a multi-scale perspective on input sequence processing by capturing both global and local contexts, surpassing the capabilities of its predecessor, which primarily focused on either global or local information. Specifically, the global Probsparse self-attention module is configured to capture the entire sequence input, while the local Probsparse self-attention modules are intended to concentrate on distinct segments of the sequence input. In the following, we will define some notations and present a formal modeling of the Global-Local Probsparse self-attention mechanism, along with an analysis of its computational complexity.

Notations: The time sequence starting at t and concluding at $t+\Delta$ is represented by the notation $(t : t+\Delta)$. The sequence input for the encoder is denoted as $X(t : t+\Delta) \in \mathbb{R}^{L_{In} \times d_{em}}$, where $L_{In} = \Delta$ represents the input sequence length, and d_{em} denotes the embedded dimension of the model. The queries

(Q), keys(K) , and values(V) for the input sequence are then defined as $Q = X(t : t{+}\Delta)W_Q{+}b_Q$; $K = X(t : t{+}\Delta)W_K{+} b_K$, $V = X(t : t + \Delta)W_V + b_V$, where $W_Q, W_K, W_V \in \mathbb{R}^{d_{em} \times d_{em}}$, and $b_Q, b_K, b_V \in \mathbb{R}^{L_{In} \times d_{em}}$.

Global Probsparse Self-attention: The global Probsparse self-attention mechanism is designed to capture the entire sequence input. Utilizing the ProbSparse Self-attention formula in [20], the calculation of it is defined as:

$$\text{Attention}_g(Q_g, K_g, V_g) = \text{softmax}\left(\frac{\bar{Q}_g K_g^\top}{\sqrt{d_{em}}}\right) V_g, \quad (2)$$

where $Q_g, K_g, V_g \in \mathbb{R}^{L_{In} \times d_{em}}$ represent the queries, keys, and values for the entire sequence input, and $\bar{Q}_g \in \mathbb{R}^{u \times d_{em}}$ represents a filtered sparse matrix comprising the top u Q_gs selected based on the sparsity measurement formula [20]. The selection parameter u is defined by the equation $u = c{\cdot}\ln L_{In}$, where c is a constant sampling factor. The sparsity measurement formula is given by:

$$M(q_i, K) = \ln \sum_{j=1}^{L_{In}} e^{\frac{q_i k_j^\top}{\sqrt{d}}} - \frac{1}{L_{In}} \sum_{j=1}^{L_{In}} \frac{q_i k_j^\top}{\sqrt{d}}, \quad (3)$$

where the q_i, k_j represent the ith and jth row in Q, K respectively.

Local Probsparse Self-attention: Since our local Probsparse self-attention modules are tailored to focus on distinct segments of the sequence input, we divide the entire input $X(t : t + \Delta)$ into N subsequences: $X(t : t + \Delta_l)$, $X(t{+}\Delta_l : t{+}2\Delta_l)$,..., $X(t{+}(n{-}1)\Delta_l : t{+}n\Delta_l)$, where Δ_l stands for each local subsequence input length. Consequently, the queries, keys, and values are also segmented accordingly: $Q = [Q_{l_1}, Q_{l_2}, ..., Q_{l_N}]^\top$; $K = [K_{l_1}, K_{l_2}, ..., K_{l_N}]^\top$, $V = [V_{l_1}, V_{l_2}, ..., V_{l_N}]^\top$. Thus, by using the ProbSparse Self-attention formula Eq.(3), we compute local Probsparse self-attention for each subsequence as follows:

$$\text{Attention}_{l_i}(Q_{l_i}, K_{l_i}, V_{l_i}) = \text{softmax}\left(\frac{\bar{Q}_{l_i} K_{l_i}^\top}{\sqrt{d_{em}}}\right) V_{l_i}, \quad (4)$$

where $Q_{l_i}, K_{l_i}, V_{l_i} \in \mathbb{R}^{L_{In/N} \times d_{em}}$ signifying the queries, keys, and values for the ith subsequence input, and the filtered sparse matrix $\bar{Q}_{l_i} \in \mathbb{R}^{m \times d_{em}}$ contains the top m Q_{l_i}s selected based on the sparsity measurement formula Eq.(3). The parameter m is controlled by the relation $m = c \cdot \ln\left(\frac{L}{N}\right)$.

Global-Local Probsparse Self-Attention: After computing the local Probsparse self-attention and the global Probsparse self-attention for the input sequence, we concatenate these attentions to form the Global-Local Probsparse Self-Attention:

$$\text{Attention}_{gl}(Q, K, V) = \begin{bmatrix} \text{Attention}_g(Q_g, K_g, V_g) \\ \text{Attention}_{l_1}(Q_{l_1}, K_{l_1}, V_{l_1}) \\ \vdots \\ \text{Attention}_{l_N}(Q_{l_N}, K_{l_N}, V_{l_N}) \end{bmatrix}. \quad (5)$$

Computational Complexity: Given a sequence input $X(t : t + \Delta) \in \mathbb{R}^{L \times d_{em}}$, the computational complexity of Global-Local Probsparse Self-Attention is as follows:

$$O\left(\text{Attention}_{gl}(Q, K, V)\right) = O\left(L \log(L)\right). \quad (6)$$

Proof: for the computational complexity of Global-Local Probsparse Self-Attention

$$\begin{aligned} O\left(\text{Attention}_{gl}(Q, K, V)\right) &= O\left(\text{Attention}_g(Q_g, K_g, V_g)\right) \\ &+ \sum_{i=1}^{N} O\left(\text{Attention}_{l_i}(Q_{l_i}, K_{l_i}, V_{l_i})\right) \quad (7) \end{aligned}$$

By applying the time complexity formula of ProbSparse Attention [20] to Eq.(7), we obtain:

$$\begin{aligned} \text{RHS of Eq.(7)} &= O\left(L \log(L)\right) + \textstyle\sum_{i=1}^{N} O\left(\frac{L}{N} \log(\frac{L}{N})\right) && (8) \\ &= O\left(L \log(L)\right) + N \times O\left(\frac{L}{N} \log\left(\frac{L}{N}\right)\right) && (9) \\ &= O\left(L \log(L)\right) + O\left(L \log\left(\frac{L}{N}\right)\right) && (10) \\ \text{RHS of Eq.(10)} &\leq 2O\left(L \log(L)\right) = O\left(L \log(L)\right). && (11) \end{aligned}$$

□

It is clear that Informer's Probsparse Self-Attention mechanism exhibits computational complexity on the order of $O(L \log(L))$, which aligns with our proposed Global-Local Probsparse Self-Attention mechanism. This indicates that our method can achieve processing efficiency similar to that of Informer while also enhancing the model's ability to capture global and local context.

D. Architecture Variants

For the error correction models, we build two variants of GloLoSAT, denoted as GloLoSAT-OF (Orbital Features Focused) and GloLoSAT-TL (Triple Independent). The GloLoSAT-OF variant is achieved by modifying the final FCN (fully connected layer) of the original GloLoSAT so that it only outputs orbit errors under three essential orbital features: *latitude*, *longitude*, and *altitude[Km]*. In contrast, the GloLoSAT-TL variant adopts a more fine-grained approach. Specifically, it is composed of three separate models. Each model diverges from the conventional methods that predict all features concurrently. Instead, their final FCN layer is modified to predict orbit errors under each orbital feature independently. As a result, this approach yields three specialized models, each uniquely responsible for predicting *latitude errors*, *longitude errors*, and *altitude errors[Km]* respectively. These individual predictions are then synthesized to generate the final satellite orbit error corrections. Following a similar modification strategy, we also introduce two variants for Informer [20] and Informer(stack) [19], respectively: Informer-OF, Informer-TL, Informer(stack)-OF, and Informer(stack)-TL.

E. Orbit Prediction Stage

As illustrated on the left-hand side in Fig. 1, the final stage generates the final satellite orbit prediction by integrating the error corrections with the corresponding SGP4 predictions. Specifically, this process first leverages the trained error model to process an input sequence ${}^{GEO}X_{\boldsymbol{e}}(t_i : t_i + \Delta_1)$, where Δ_1 represents the length of the input sequence and obtains the predicted errors ${}^{GEO}\widehat{X}_{\boldsymbol{e}}(t_i + \Delta_1 : t_i + \Delta_1 + \Delta_2)$, where Δ_2 indicating the size of the prediction window. The resulting orbit prediction is then calculated by:

$$\begin{aligned} {}^{GEO}\widehat{O}(t_i + \Delta_1 : t_i + \Delta_1 + \Delta_2) = {}^{GEO}\widehat{X}_{\boldsymbol{e}}(t_i + \Delta_1 : t_i + \Delta_1 + \Delta_2) + \\ {}^{GEO}X_{SGP4}(t_i + \Delta_1 : t_i + \Delta_1 + \Delta_2). \end{aligned} \quad (12)$$

V. EXPERIMENT

A. Baselines

To evaluate our proposed method, we have selected four previous models with two relevant categories as baselines. The first category consists of the SGP4 mathematical orbital propagation model [27], used for direct satellite orbit prediction. The second category comprises three time-series forecasting models: LSTM [28], Informer [20], and Informer(stack) [19]. These models are employed as error correction models within our proposed approach. Here are the key configurations for each model in the second category. The LSTM is parameterized with n_hidden = 128, n_layers = 2, $dropout$ = 0.2, $optimizer = Adam$, $epochs = 20$, $earlystopping = 5$, and $lr = 1e^{-4}$. The Informer is structured with a *2-layer* encoder and a *1-layer* decoder. It is trained with $dropout$ = 0.05, $optimizer = Adam$, $epochs = 20$, $earlystopping = 5$, $lr = 1e^{-4}$, and *embedded dimension* = 512. The Informer(stack) mirrors the Informer's configurations, but its encoder is composed of three stacks with 3, 2, and 1 layers, respectively.

B. Experiment Training Settings

In this study, we conduct a comparative analysis of our proposed method against established baseline models within three practical satellite orbit forecasting scenarios.

(1) Short-Term Prediction: In this scenario, the models are trained on the error dataset D_e, leveraging the preceding 300-second of trajectories to forecast the forthcoming 60-second trajectories with 1-second intervals. This setting is designed to potentially enable near-real-time updates on satellite positioning, which can be beneficial for urgent collision avoidance measures and ensuring the satellite remains within its intended orbit promptly.

(2) Long-Term Prediction: In this scenario, the models are also trained on the D_e but are tasked with utilizing the prior 300-second trajectories to predict the next 720-second trajectories with 1-second intervals. This scenario is crafted with the intention of possibly assisting in enhancing the scheduling of communication windows, ground station passes, and the management of satellite power and thermal conditions.

(3) Extended Long-Term Prediction: In this scenario, the models are trained on a modified dataset D_{eEL}, derived by uniformly sampling one data point per minute from D_e. Specifically, during the data preprocessing phase, once D_e is obtained, it undergoes a selection process to form D_{eEL} before being divided into training, validation, and testing partitions. The training objective here is to predict 720-minute trajectories into the future with 1-minute intervals based on 60-minute historical trajectories. This setting is especially relevant for LEO satellites, aligning with NASA's Spacecraft Conjunction Assessment and Collision Avoidance Best Practices Handbook [29], which states that TLEs of LEO satellites with perigee heights under 500 km are updated thrice daily, while other LEO satellites are updated at least once per day. Additionally, the observations from the Space-Track website also imply that updates for certain LEO satellites, such as Starlink satellites, occur about twice daily. Thus, a half-day prediction interval is practical for long-period LEO satellite orbit prediction, aligning with the typical TLE update frequency. This predictive setting is designed to provide operators with advanced notice to support proactive decision-making for satellite maneuvers, resource allocation, and communication scheduling.

C. Model Parameters

For both GloLoSAT-OF and GloLoSAT-TL, their encoders consist of one *3-layer stack* and three *2-layer stacks*. Specifically, the *3-layer stack* captures the entire input sequence, while the three *2-layer stacks* individually handle the left, middle, and right thirds of the input sequence. Both models include a *1-layer* decoder. Model optimization is performed using the *Adam optimizer*, initialized with a learning rate of $1e^{-4}$, which undergoes a tenfold reduction every two epochs over a total of 20 epochs. Additionally, a *dropout rate* of 0.05 and an *early stopping parameter* of 5 are implemented. The parameter configurations for the Informer-OF, Informer-TL, Informer(stack)-OF, and Informer(stack)-TL mirror those of their corresponding baseline models. Furthermore, for all prediction tasks, including the baseline models, the batch size is set to 50 for Short-Term and Long-Term scenarios and 16 for the Extended Long-Term Prediction scenario.

D. Experiment Evaluation Settings

To assess the efficacy of our proposed method for satellite orbit prediction, we analyzed the experiment outcomes of all evaluated models within the ECEF (Earth-centered and Earth-fixed) coordinate system [30], which is a non-inertial reference framework that synchronizes with the Earth's rotation and is one of the most widely adopted coordinate systems for satellite positioning and orbit prediction. To ensure a rigorous and equitable comparison, we selected the evaluation period for all prediction scenarios from 2023/12/17 23:05:01 to 2023/12/19 05:05:00. Upon collating the prediction outcomes from each model, we apply the coordinate transformation function provided by pyproj library to both the actual and predicted orbits, facilitating the conversion of *latitude, longitude,* and *altitude* values into their corresponding representations within the ECEF system. We then use a set of performance metrics to measure the accuracy and reliability of each method. These metrics include Root Mean Square Error (RMSR), Mean Absolute Error (MAE), and Mean Geocentric Distance Error (MGDE), the latter being computed as $MGDE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2+(z_i-\hat{z}_i)^2}$, where n is the number of observations, (x_i, y_i, z_i) and $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ are the actual and predicted values respectively for the ith observation. All of the ML-based models are implemented using PyTorch and are trained and tested on GeForce RTX 4090 ×6.

E. Experiment Results and Analysis

The experimental results for each orbit prediction method across three different prediction settings are summarized in Table I through III, with the best results in bold.

Performance Comparison With Baselines. The observation of our experiment results shows that our proposed methods

TABLE I
THE PERFORMANCE RESULTS ON THE SHORT-TERM PREDICTION TASK.‡ MEANS OUR PROPOSED MODEL.

Short-Term Prediction (ECEF, unit in meters)			
Metric	MAE	RMSE	MGDE
SGP4	321.989	417.923	637.902
LSTM + SGP4	324.264	422.208	642.029
Informer + SGP4	535.77	648.83	1039.76
Informer(stack) + SGP4	545.47	658.43	1056.71
‡ Informer-OF + SGP4	295.924	403.545	583.802
‡ Informer(stack)-OF + SGP4	303.149	420.524	595.315
‡ GloLoSAT-OF + SGP4	273.481	526.086	540.527
‡ Informer-TL + SGP4	260.285	355.646	510.162
‡ Informer(stack)-TL + SGP4	264.407	365.838	520.202
‡ GloLoSAT-TL + SGP4	**253.732**	**346.548**	**495.715**

TABLE II
THE PERFORMANCE RESULTS ON THE LONG-TERM PREDICTION TASK. ‡ MEANS OUR PROPOSED MODEL.

Long-term Prediction (ECEF, unit in meters)			
Metric	MAE	RMSE	MGDE
SGP4	321.989	417.923	637.902
LSTM + SGP4	326.446	425.655	644.288
Informer + SGP4	639.434	751.176	1228.982
Informer(stack) + SGP4	578.423	690.78	1117.3
‡ Informer-OF + SGP4	283.445	386.59	557.269
‡ Informer(stack)-OF + SGP4	308.734	411.637	610.936
‡ GloLoSAT-OF + SGP4	265.856	385.471	522.81
‡ Informer-TL + SGP4	253.576	346.116	495.361
‡ Informer(stack)-TL + SGP4	256.685	373.633	503.23
‡ GloLoSAT-TL + SGP4	**250.898**	**341.99**	**489.712**

TABLE III
THE PERFORMANCE RESULTS ON THE EXTENDED LONG-TERM PREDICTION TASK. ‡ MEANS OUR PROPOSED MODEL.

Extended Long-Term Prediction(ECEF, unit in meters)			
Metric	MAE	RMSE	MGDE
SGP4	320.575	413.403	634.079
LSTM + SGP4	642.56	806.622	1251.47
Informer + SGP4	543.92	692.17	1057.39
Informer(stack) + SGP4	632.319	788.58	1207.04
‡ Informer-OF + SGP4	302.785	405.393	596.42
‡ Informer(stack)-OF + SGP4	309.561	411.01	606.83
‡ GloLoSAT-OF + SGP4	297.427	398.955	585.484
‡ Informer-TL + SGP4	268.486	363.218	526.142
‡ Informer(stack)-TL + SGP4	276.9	372.576	543.207
‡ GloLoSAT-TL + SGP4	**262.916**	**355.085**	**514.392**

improve the accuracy of satellite orbit prediction under various scenarios compared with the baselines. Our proposed models outperform the SGP4 on MGDE by decreasing the errors up to 22.3% (in Table I: Short-Term Prediction), 23.2% (in Table II: Long-term Prediction), and 18.9% (in Table III: Extended Long-Term Prediction). This implies that the positional discrepancies between D_{SGP4} and D_{REAL} are learnable, and our proposed methods make a positive contribution to the orbit prediction problems. Additionally, when compared to traditional multivariate time-series forecasting models combined with SGP4, such as LSTM + SGP4, Informer + SGP4, and Informer(stack) + SGP4, our proposed models show a potential reduction in MGDE by up to 53.1%, 60.2%, and 58.9% for short-term, long-term, and extended long-term predictions, respectively. This observation underscores the importance of the output feature dimensionality in the final FCN layer for time-series forecasting tasks. Specifically, the original LSTM, Informer, and Informer(stack) models generate outputs for an extensive range of features, which may lead to a dispersion of focus on many task-irrelevant features, such as *sintime*, *costime*, *index*, *sinindex*, and *Altitude [Mi]*. Conversely, our proposed models are tailored to prioritize key orbital features: *latitude, longitude*, and *altitude [Km]*. By concentrating on these orbital features, our models can minimize the dilution of attention across less relevant features, thereby leading to more accurate predictions.

Performance Comparison Between Different Self-attention Mechanisms. The results across tables I through III show that our Global-Local Probsparse Self-Attention-based orbit prediction methods outperform their counterparts in all tested prediction scenarios. The GloLoSAT-TL + SGP4 method exhibits improvements over the Informer-TL + SGP4 and Informer(stack)-TL + SGP4 methods, with a decrease in MGDE by up to 4.7%, 2.7%, and 5.3% for Short-Term, Long-Term, and Extended Long-Term Prediction scenarios, respectively, as shown in Tables I to III. Similarly, the GloLoSAT-OF + SGP4 method exhibits better performance compared to the Informer-OF + SGP4 and Informer(stack)-OF + SGP4 methods, achieving a reduction in MGDE by up to 9.2%, 15.8%, and 3.5% for short-term, long-term, and extended long-term predictions, respectively. These improvements underscore the essential role of the input sequence range in enhancing the self-attention module's efficacy. The Informer model processes only the input sequence's global range, potentially limiting its local contextual understanding. The Informer(stack) utilizes a tiered approach, where the first stack processes the global input sequence, the second stack addresses the right half slices of the input sequence, and the third stack focuses on the subsequent half slices, potentially overlooking the left half slices of the entire input. On the contrary, our proposed GloLoSAT, with its Global-Local Probsparse Self-Attention modules, is designed to fully encompass both global and local sequence inputs. Specifically, the global module captures the entire sequence, while the three local modules focus on the left, middle, and right thirds of the input sequence, leading to a more nuanced contextual understanding and improved orbit prediction accuracy.

Performance Comparison Between Architectural Variants. The results presented in Tables I through III demonstrate that Triple Independent variants perform better than Orbital Features Focused counterparts in various orbit prediction scenarios. Informer-TL + SGP4 outperforms Informer-OF + SGP4 on MGDE by decreasing 12.6% for Short-Term Prediction, 11.1% for Long-Term Prediction, and 11.8% for Extended Long-Term Prediction. Informer(stack)-TL + SGP4 achieves a decrease in MGDE by 12.6% for Short-Term Prediction, 17.6% for Long-Term Prediction, and 10.5% for Extended Long-Term Prediction when compared to Informer(stack)-OF + SGP4. Additionally, GloLoSAT-TL + SGP4 also shows improved performance over GloLoSAT-OF + SGP4, with MGDE reductions of 8.3% for Short-Term Prediction, 6.3% for Long-Term Prediction, and 12.1% for Extended Long-Term Prediction. These observations demonstrate the benefits of a more fine-grained approach adopted by the Triple Independent variants, where the final FCN layer is restructured to predict orbit errors for each orbital feature independently, thereby maximizing the use of additional information from other input features to enhance prediction accuracy.

VI. Conclusion and Future Work

In this paper, we attempt to address LEO satellite orbit prediction using real satellite orbit tracking data. Our approach primarily concentrates on mitigating the orbit prediction errors that arise from traditional physics-based models by implementing an ML-based error correction framework. We introduce N2YO-crawler for high-integrity, real-time data collection and present GloLoSAT, a novel three-stage transformer-based framework for real-time LEO satellite orbit prediction. To overcome the limitations of existing methods that focus only on global or local contexts, we propose a Global-Local Probsparse Self-Attention mechanism within the GloLoSAT framework, capturing both contexts with computational efficiency of $O(L\log(L))$. Extensive experiments demonstrate GloLoSAT's effectiveness in reducing SGP4 prediction errors and improving orbit prediction performance. We note that this study focuses on applying the ML-based approach to the orbit prediction of a single satellite. In our future work, we plan to extend the proposed ML-model to the orbit prediction of multiple satellites in the same constellation, which will enable the routing research in the mobile ad hoc network formed by the constellation.

Acknowledgment

This work is supported in part by the United States National Science Foundation (NSF) under grants CNS-2308761 and CNS-2006998. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

References

[1] B. Rodriguez, "Exploring the mystery of our expanding universe," https://www.jpl.nasa.gov/edu/news/2023/7/24/exploring-the-mystery-of-our-expanding-universe/, 2023.

[2] T. Darwish, G. K. Kurt, H. Yanikomeroglu, M. Bellemare, and G. Lamontagne, "Leo satellites in 5g and beyond networks: A review from a standardization perspective," *IEEE Access*, vol. 10, pp. 35 040–35 060, 2022.

[3] I. Ali, *Doppler Applications in LEO Satellite Communication Systems: CD-Rom Included*. Springer Science & Business Media, 2002.

[4] C. Daehnick, J. Gang, and I. Rozenkopf, "Space launch: Are we heading for oversupply or a shortfall?" https://www.mckinsey.com/industries/aerospace-and-defense/our-insights/space-launch-are-we-heading-for-oversupply-or-a-shortfall, 2023.

[5] D. Brouwer, "Solution of the problem of artificial satellite theory without drag," *Astronomical Journal, Vol. 64, p. 378 (1959)*, vol. 64, p. 378, 1959.

[6] O. Montenbruck, E. Gill, and F. Lutze, "Satellite orbits: models, methods, and applications," *Appl. Mech. Rev.*, vol. 55, no. 2, pp. B27–B28, 2002.

[7] J. Liu and R. Alford, "Semianalytic theory for a close-earth artificial satellite," *Journal of Guidance and Control*, vol. 3, no. 4, pp. 304–311, 1980.

[8] F. Caldas and C. Soares, "Machine learning in orbit estimation: A survey," *Acta Astronautica*, 2024.

[9] G. Acciarini, A. G. Baydin, and D. Izzo, "Closing the gap between sgp4 and high-precision propagation via differentiable programming," *arXiv preprint arXiv:2402.04830*, 2024.

[10] F. Alesiani, M. Takamoto, T. Kamiya, and D. Etou, "Leo satellite orbit prediction with physics informed machine learning," in *NeurIPS workshop*, 2023.

[11] B. Li, J. Huang, Y. Feng, F. Wang, and J. Sang, "A machine learning-based approach for improved orbit predictions of leo space debris with sparse tracking data from a single station," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4253–4268, 2020.

[12] B. Li, Y. Zhang, J. Huang, and J. Sang, "Improved orbit predictions using two-line elements through error pattern mining and transferring," *Acta Astronautica*, vol. 188, pp. 405–415, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576521004045

[13] H. Peng and X. Bai, "Exploring capability of support vector machine for improving satellite orbit prediction accuracy," *Journal of Aerospace Information Systems*, vol. 15, no. 6, pp. 366–381, 2018.

[14] H. Peng and X. Bai, "Machine learning approach to improve satellite orbit prediction accuracy using publicly available data," *The Journal of the astronautical sciences*, vol. 67, no. 2, pp. 762–793, 2020.

[15] H. Peng and X. Bai, "Artificial neural network–based machine learning approach to improve orbit prediction accuracy," *Journal of Spacecraft and Rockets*, vol. 55, no. 5, pp. 1248–1260, 2018.

[16] n2yo.com, "Terms of use for n2yo.com," https://www.n2yo.com/about/?a=terms., 2024.

[17] N. T. Tran and J. Xin, "Fourier-mixed window attention: Accelerating informer for long sequence time-series forecasting," *arXiv preprint arXiv:2307.00493*, 2023.

[18] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Etsformer: Exponential smoothing transformers for time-series forecasting," *arXiv preprint arXiv:2202.01381*, 2022.

[19] H. Zhou, J. Li, S. Zhang, S. Zhang, M. Yan, and H. Xiong, "Expanding the prediction capacity in long sequence time-series forecasting," *Artificial Intelligence*, vol. 318, p. 103886, 2023.

[20] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[21] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International conference on machine learning*. PMLR, 2022, pp. 27 268–27 286.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Pre-training of deep bidirectional transformers for language understanding in: Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)," *Minneapolis, MN: Association for Computational Linguistics*, pp. 4171–86, 2019.

[24] T. Core, "Time series forecasting," https://www.tensorflow.org/tutorials/structured_data/time_series#feature_engineering, 2024.

[25] D. Vallado and P. Crawford, "Sgp4 orbit determination," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008, p. 6770.

[26] r project.org, "Convert coordinates from teme to itrf," https://search.r-project.org/CRAN/refmans/asteRisk/html/TEMEtoITRF.html#:∼:text=Description,Earth%20surface%20in%20its%20rotation), 2024.

[27] D. Vallado, "Fundamentals of astrodynamics and applications fifth edition," https://celestrak.org/software/vallado-sw.php, 2023.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] F. J. Krage, "Nasa spacecraft conjunction assessment and collision avoidance best practices handbook," p. 10, 2023.

[30] "Coordinate systems," https://dirsig.cis.rit.edu/docs/new/coordinates.html, 2023.