

Received 21 February 2025; revised 25 August 2025; accepted 28 October 2025. Date of publication 4 November 2025; date of current version 28 November 2025.

This article was recommended by Executive Editor Teresa Vidal-Calleja.

Digital Object Identifier 10.1109/TFR.2025.3628826

# MarsLGPR: Mars Rover Localization With Ground Penetrating Radar

ANJA SHEPPARD<sup>1</sup> (Student Member, IEEE) AND KATHERINE A. SKINNER<sup>2</sup> (Member, IEEE)

Department of Robotics, University of Michigan, Ann Arbor, MI 48109 USA

CORRESPONDING AUTHOR: ANJA SHEPPARD (anjashep@umich.edu)

This work was supported in part by the National Science Foundation (NSF) under Grant DGE 2241144; in part by the Department of Robotics, University of Michigan; in part by the University of Michigan Space Institute Pathfinder Grant; and in part by the NASA Pennsylvania Space Grant Consortium (PSGC) Mini-Grants Program and Duquesne University Biomedical Engineering (BME) Departmental Funds.

(Regular Article)

**ABSTRACT** In this work, we propose the use of Ground Penetrating Radar (GPR) for rover localization on Mars. Precise pose estimation is an important task for mobile robots exploring planetary surfaces, as they operate in GPS-denied environments. Although visual odometry (VO) provides accurate localization, it is computationally expensive and can fail in dim or high-contrast lighting. Wheel encoders can also provide odometry estimation, but are prone to slipping on the sandy terrain encountered on Mars. Although traditionally a scientific surveying sensor, GPR has been used on Earth for terrain classification and localization through subsurface feature matching. The Perseverance rover and the upcoming ExoMars rover have GPR sensors already equipped to aid in the search of water and mineral resources. We propose to leverage GPR to aid in Mars rover localization. Specifically, we develop a novel GPR-based deep learning model that predicts 1-D relative pose translation. We fuse our GPR pose prediction method with inertial and wheel encoder data in a filtering framework to output rover localization. We perform experiments in a Mars analog environment and demonstrate that our GPR-based displacement predictions both outperform wheel encoders and improve multimodal filtering estimates in high-slip environments. Finally, we present the first dataset aimed at GPR-based localization in Mars analog environments, which will be made publicly available at: <https://umfielddrobotics.github.io/marslgpr/>

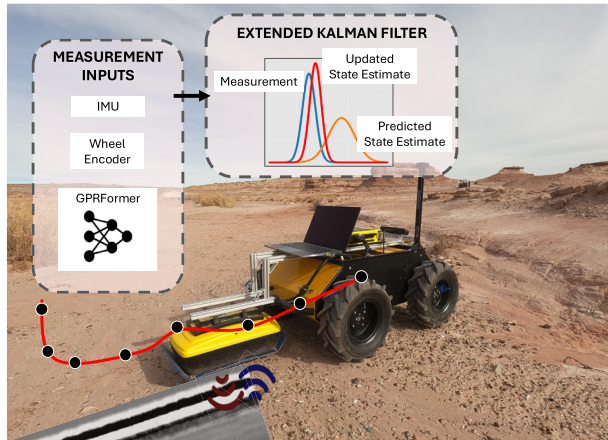
**INDEX TERMS** Ground penetrating radar (GPR), localization, multimodal perception, space robotics.

## I. INTRODUCTION

**P**RECISELY localizing planetary rovers in GPS-denied environments such as the Moon and Mars is a challenging task. Until recently, localization for the Perseverance rover on the surface of Mars has required a human-in-the-loop for manual corrections and tuning [1], [2]. New approaches have seen increasingly autonomous solutions for global state estimation [3], but this problem has not been entirely solved. As interest increases in launching lower cost rovers with shorter lifespans, there is a need for real-time GPS-denied rover localization that does not require a large team of human operators in the loop.

Wheel encoder and inertial sensor data are commonly used to localize a robot. However, wheel encoders are notoriously prone to incorrect predictions due to wheel slip. The surface of Mars has many sandy areas that have caused drastic

enough wheel slip and sinkage to permanently trap the Spirit rover [6], making encoders an unreliable sensor for localization. Visual odometry (VO) has shown the benefits of including additional sensing modalities to improve the state estimation. In VO, features are tracked between subsequent pairs of images in order to characterize the robot's motion and look for loop closure. Although VO typically outperforms inertial and wheel encoder odometry, this method is more computationally costly and is not realistic for rovers to run in real-time given current hardware constraints. In a VO demonstration for Martian rovers, Spirit and Opportunity needed 3 min to complete a single vision-based tracking step [7]. The Curiosity rover has demonstrated on-board VO tests, but with each image processing step taking around 34 s [8]. For reference, state-of-the-art terrestrial VO runs at 30–40 frames/s [9]. The Perseverance rover localization



**FIGURE 1.** Our mobile robot collection platform in a Mars analog environment. The proposed method demonstrates a novel approach to relative displacement estimation with GPR, which then informs a filtering-based state estimation algorithm.

framework [3] includes offline optimization-based localization methods, such as matching against existing orbital elevation maps, but does not mention the use of real-time VO on board the rover.

In the space domain, scientific sensors can be creatively repurposed to provide information about a robot’s environment to the navigation algorithms, such as terrain type or slip characterization [10], [11]. Ground Penetrating Radar (GPR) has already demonstrated promise on Earth as a tool for localization by using subsurface features [12], [13] and for terrain classification using deep learning methods [14]. The Perseverance rover is equipped with the RIMFAX [15] GPR sensor, and the planned European Space Agency ExoMars rover will have a GPR to search for in situ drilling sites [16]. The upcoming CADRE mission to the surface of the Moon plans to have several robots—each equipped with GPR—collect data cooperatively [17].

In this work, we propose a novel framework that leverages GPR to improve the localization of planetary rovers (see Fig. 1). Our main contributions are as follows:

- 1) We develop a novel deep learning transformer-based model for real-time relative pose estimation with GPR.
- 2) We integrate GPR pose estimation into a real-time filtering framework for rover localization.
- 3) We collect and present the first GPR-based localization dataset collected in a Mars analog environment, titled MarsLGPR.

We validate our proposed method on our dataset collected in a Mars analog environment, as well as on a publicly available GPR localization dataset. Through experiments, we demonstrate that incorporating GPR can improve the localization estimation for mobile robot platforms operating on complex terrain.

## II. RELATED WORKS

### A. LOCALIZATION

The Perseverance rover has a robust global localization pipeline that has achieved submeter accuracy [2]. However, it often involves humans-in-the-loop and specially engineered data pipelines that may be too computationally intensive to run locally. As the accessibility of landing a planetary rover increases, there will be a great need for relatively precise rover localization that can run onboard without human support.

There are two main approaches to localization: filtering-based [18] and optimization-based [13]. Filtering for robot localization involves nonlinear dynamics, making Extended Kalman Filters (EKFs) and Unscented Kalman Filters (UKFs) popular approaches [19]. Optimization-based approaches, such as GTSAM [20], estimate a pose within a factor graph framework by minimizing the residual from different sensor inputs. Localization with optimization can lead to higher accuracy, but properly constraining the factors for tractable optimization can be a challenging task in real-time with noisy sensors. In this work, we use an EKF framework for localization with a Inertial Measurement Unit (IMU), wheel encoder, and GPR data. Although EKFs require careful parameter tuning, they are reliable for real-time use.

### B. GROUND PENETRATING RADAR

GPR is a popular sensor in the fields of civil engineering [21], archeology [22], environmental science [23], and planetary geology [24]. The GPR on-board the Perseverance rover has been used to search for water and to characterize the electromagnetic properties of the Martian subsurface [15]. The signal returns from a GPR allow for a nonintrusive view of subsurface materials and features. Single GPR returns or A-scans can be concatenated sequentially to form a B-scan image that encodes 2-D spatial information.

Subsurface features are generally more static than visual features above the surface—for example, a scene might look completely different to a camera after a fresh coat of snow or during a rainstorm. In addition, processing GPR traces is less computationally intensive than performing visual feature matching. Relying on GPR features as an additional mode of localization could improve the absolute positioning of planetary rovers.

### C. LOCALIZING GPR

The use of GPR for localization has been under development for the last decade. Although some work has investigated the use of matching single GPR A-scans [25], most approaches take advantage of patterns in sequential GPR traces that are present in concatenated B-scans. Early approaches rely on a GPS prior and a preregistered map in order to match incoming GPR measurements to an existing database of underground features [26], [27]. In order to reduce reliance on GPS, subsequent methods [4], [12], [28], [29], [30], [31], [32], [33]

TABLE 1. Localizing GPR datasets.

Dataset	RGB-D	GPR	IMU	Ground Truth	Location	Publicly Available
GROUNDDED [4]	FLIR Grasshopper Velodyne HDL-64	Custom LGPR	RT3003 IMU	OXTS RT3003 GPS	Highway Rural Road Urban Road	No
CMU-GPR [5]	Realsense D435	NOGGIN 500	XSENS MTi-30	Leica TS15 Station GPS	Basement Factory Floor Parking Garage	Yes
MarsLGPR (Ours)	ZED2i	NOGGIN 500	ZED2i	RTAB-Map	Mars Analog	Yes

focus on more robust approaches for GPR feature detection that could aid in matching newly collected data with the preregistered map of GPR traces. Some approaches utilize the signal processing for finding and matching features across B-scans [12], [31], [32], [33], while other works use deep learning for feature extraction [4], [28], [29], [30].

The need for a preregistered map greatly constrains the use case of these methods and makes deployment in unfamiliar areas more difficult. Recent localizing GPR work has begun to establish approaches for *relative* pose estimation between B-scan submaps using deep learning [13], [34]. The only work that has fused relative GPR localization with other sensors, such as IMU and wheel encoders [13], has not investigated the use of this method on complex terrain representative of Mars surface terrain. Our proposed work is the first to demonstrate the use of transformer deep learning architectures for GPR-based relative pose estimation targeted at rugged off-road environments.

#### D. LOCALIZING GPR DATASETS

There are currently two localizing GPR datasets: GROUNDDED [4] and CMU-GPR [5]. Both datasets contain GPR, RGB, depth, IMU, and GPS data, as shown in Table 1. The location coverage of both datasets includes roads, parking garages, basements, and factory floors. The GROUNDDED dataset is not currently available to the public.

While these datasets are useful for accelerating localizing GPR for autonomous ground vehicles, there is no dataset that is specifically aimed at exploring GPR localization on Mars. We propose MarsLGPR, a localizing GPR dataset collected on Martian analog terrain with additional stereo camera, IMU, and reference pose data.

### III. TECHNICAL APPROACH

In this section, we review the key technical concepts in our approach to GPR-based localization. We first discuss the GPR sensor model and how we can use it to predict relative displacement. Then, we discuss the model architecture for GPRFormer, our model that predicts the relative displacement from GPR traces. Finally, we include details of our EKF implementation that leverages GPRFormer predictions for improved localization.

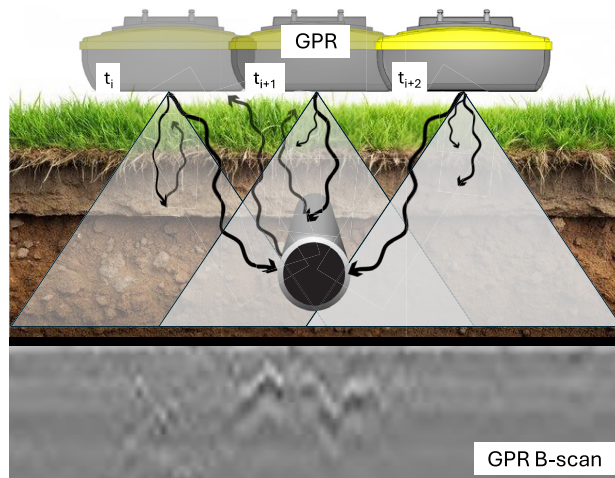


FIGURE 2. Consecutive GPR samples have overlapping beam spreads, which cause them to detect the same object across multiple returns. Because of this, the resulting GPR B-scan image may not intuitively represent the geometry of objects underground. However, we can use this property of the sensor to predict relative pose translation using deep learning.

#### A. GPR SENSOR MODEL

GPR uses radio waves to characterize subsurface properties, often depicted as a time-series B-scan (see Fig. 2). Although a B-scan might appear to be a simple pictorial cross section of the area below the sensor, the sensor returns do not directly capture a visual “snapshot” of subterranean feature geometry. This is due to the fact that GPR samples capture relative permittivity between materials [35].

When a GPR transmits a radio wave, the reflected energy ( $R$ ) captured in the return is a function of the difference in relative permittivity ( $\kappa_1, \kappa_2$ ) between adjacent materials underground [23]:

$$R = \frac{\sqrt{\kappa_1} - \sqrt{\kappa_2}}{\sqrt{\kappa_1} + \sqrt{\kappa_2}}. \quad (1)$$

The larger the difference in the permittivity of the two materials, the stronger the reflected signal will appear. For example, if there is a pipe buried underground, a higher reflectance will be visible at the depth corresponding to the

boundary between the top of the object and the surrounding material.

However, the shape of the object visible in the B-scan may not intuitively match the true geometry of the object. A GPR has a wide beam dispersion, so a submerged object will reflect some of the signal back at the transmitter even if GPR is not directly over the object, as shown in Fig. 2. This results in an almost imperceptible correlation of consecutive A-scans. It is this idea that has inspired our investigation into using deep learning for predicting relative pose translation from GPR returns.

In addition, the frequency of the GPR sensor determines the penetration depth, with a higher frequency corresponding to higher resolution and shallower depth [35]. The subsurface material properties also impact the penetration depth, with materials such as wet sand or clay reducing signal reach.

### 1) FILTERING

In order to reduce the signal noise and emphasize visible features, signal filtering such as background removal, dewow filters, Spreading and Exponential Compensation (SEC) gain, and wavelet denoising may be utilized [5]. An example of a prefiltered and postfiltered B-scan is shown in Fig. 3.

Background removal is one of the most standard GPR denoising approaches. It involves removing “background” horizontal layering artifacts, typically produced from a direct wave [35]. We use a global averaging window over the entire batch of traces  $\mathbf{G} = [\mathbf{g}_0, \mathbf{g}_1, \dots, \mathbf{g}_{k-1}]$ , where  $k$  is the number of traces in the network input:

$$\mathbf{G}'_{ij} = \mathbf{G}_{ij} - \frac{1}{k} \sum_{j=0}^{k-1} \mathbf{G}_{ij}. \quad (2)$$

Dewow filters remove low-frequency drift from the GPR data with polynomial detrending [37]. For a given trace  $\mathbf{g} \in \mathbb{R}^t$ , we find the coefficients  $\mathbf{a}$  of a degree- $\rho$  polynomial representing the drift for each trace  $\mathbf{g} \in \mathbf{G}$ , where  $\rho = 3$ :

$$\mathbf{a} = \arg \min \sum_{i=1}^t \left| \mathbf{g}_i - \sum_{j=0}^{\rho} a_j i^j \right|^2 \quad (3)$$

$$\mathbf{p}_i = \sum_{j=0}^{\rho} a_j i^j, \quad \text{for } i = 0, 1, \dots, t-1 \quad (4)$$

$$\mathbf{g}' = \mathbf{g} - \mathbf{p}. \quad (5)$$

SEC gain compensates for the loss of energy as the wavefront propagates through the ground. It does this by scaling the amplitude depending on the order of return [38]. Parameter  $a$  controls the magnitude of the amplification, while parameter  $b$  controls how exponential or linear the amplification is. Threshold  $T_\gamma$  sets a maximum amplification at a certain index to avoid over-emphasizing deeper returns (especially when  $b$  is closer to 0 and is thus more exponential).



**FIGURE 3. (a) Comparison of a raw B-scan containing  $k = 10$  traces and (b) filtered result after applying background removal, dewow filtering, SEC gain, and wavelet denoising. Note that the direct wave bands are removed, and the deeper features (more rightwards) are more pronounced.**

We set  $a = 0.015$ ,  $b = 0$ , and  $T_\gamma = 100$ :

$$\gamma = \begin{cases} i^b e^{ai} & \text{if } i < T_\gamma \\ T_\gamma^b e^{aT_\gamma} & \text{if } i \geq T_\gamma \end{cases} \quad \text{for } i = 0, 1, \dots, t-1. \quad (6)$$

$\gamma$  is then multiplied, element-wise, across each row in the B-scan  $\mathbf{G}$  [5].

Wavelet denoising aims to separate out the true GPR signal from noise by decomposing the input into the wavelet domain, applying thresholding, and then reconstructing the input in the image space. We selected the Daubechies-6 “mother” wavelet due to its similarity to a GPR signal and previously demonstrated the superior performance [39].

In order to reduce the signal noise, GPR sensors often collect several samples immediately after one another and average them in a process called “stacking.” The stacking process is usually applied during signal postprocessing [35]. We apply stacking over sets of three traces.

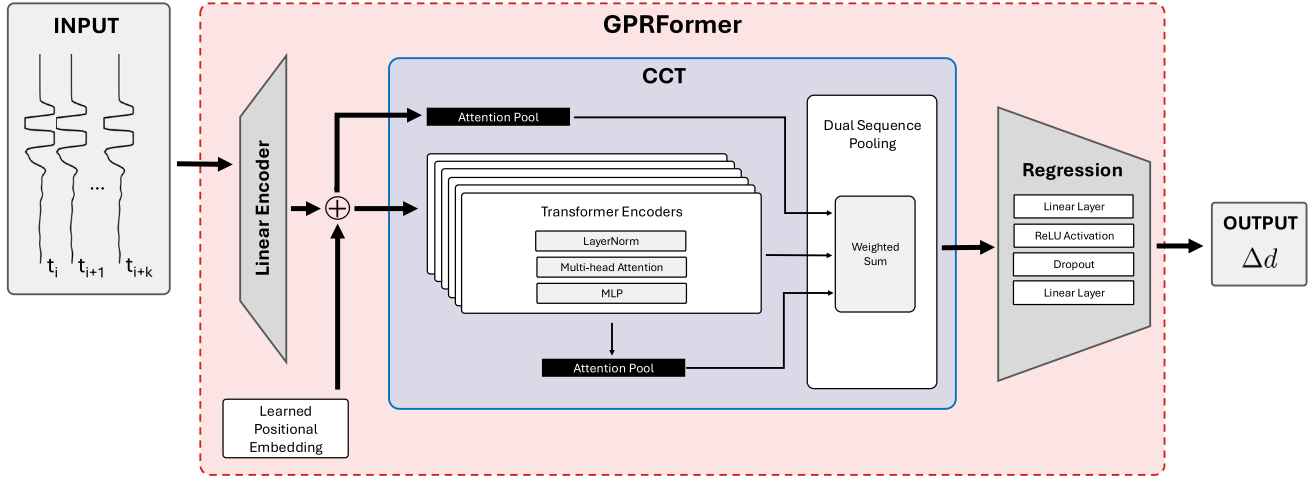
### B. GPRFormer

Fig. 4 shows an overview of our proposed model architecture, GPRFormer, which estimates relative displacement from input GPR data. GPRFormer leverages transformer networks, which uses attention to extract meaningful correlations across tokens or patches in the input data [40]. Transformer networks often rely on large datasets, but recent work introduced the Convolutional Compact Transformer (CCT) [36], which shows improved performance compared with Convolutional Neural Networks (CNNs) on small datasets.

Inspired by methods for 3-D relative camera pose estimation [41], we use a supervised compact transformer architecture to process temporal sequences of 1-D sensor data, concatenated into a GPR B-scan. The model consists of three primary components: a linear input encoder that maps 200-D raw sensor readings to 256-D token embeddings, a modified CCT backbone with dual-sequence pooling, and a regression head for displacement prediction. Mean Squared Error (MSE) loss is used during training.

#### 1) LINEAR INPUT ENCODER

The input encoder is a simple linear projection layer that is applied to each of the  $k$  consecutive sensor readings to transform them from a raw 200-D feature space to a 256-D embedding space. The GPR B-scan format is natively structured and very compact, so we found that this simple encoder architecture to be sufficient.



**FIGURE 4.** Our GPRFormer network takes in a GPR B-scan with  $k$  consecutive traces, performs a linear encoding, adds learned positional embeddings, and then passes the encoded tokens through a compact transformer [36]. The output from the network is an estimate of relative displacement ( $\Delta d$ ) that the vehicle traveled while collecting the GPR traces in the input.

A key component of the model is the positional embedding scheme, which allows the transformer to understand the sequential nature of the input data, as the traces are stacked in order of capture. We specifically utilize learned positional embeddings, which allows the model to adapt positional representations specifically for the sensor data domain. The positional embeddings are added element-wise to the token embeddings before entering the transformer stack.

## 2) CCT TRANSFORMER

Our modified CCT backbone has six transformer encoder layers, each containing pre-normalization, multihead self-attention with four attention heads, and feed-forward networks each with an expansion ratio of 3.0. The architecture includes a dual-sequence pooling mechanism that combines features from both pre-transformer and post-transformer representations. Two separate attention pooling layers generate sequencewise attention weights, producing pooled representations  $\mathbf{x}_1$  (from post-transformer features) and  $\mathbf{x}_2$  (from pre-transformer features applied to post-transformer outputs). The final representation combines these via a learnable parameter  $\alpha$ :

$$\mathbf{x}_{\text{final}} = \alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2. \quad (7)$$

This allows the model to balance between raw sequential patterns and learned contextual representations from the transformer.

## 3) REGRESSION HEAD

The original CCT classification head is replaced with an identity layer, and finally, a regression head consisting of two fully connected layers with rectified linear unit (ReLU) activation and dropout ( $p = 0.1$ ) predicts the final single displacement distance.

## C. EKF FOR LOCALIZATION

We integrate our predicted GPR displacement in an EKF framework to provide rover localization. EKFs are a popular tool for sensor fusion for robot localization, and they have been well-defined in previous papers [18], [19]. We present an abridged description here to describe our specific implementation.

We make a simplifying assumption that the robot is operating in locally 2-D environments in the east–north–up (ENU) frame. The sensor mounted on the rear of the robot platform limits its movements to relatively planar regions. We denote the robot’s state ( $\mathbf{x}_\tau$ ) and measurement ( $\mathbf{z}_\tau$ ) vectors as follows:

$$\mathbf{x}_\tau = \begin{bmatrix} x_\tau \\ y_\tau \\ \psi_\tau \\ \dot{x}_\tau \\ \dot{y}_\tau \\ \dot{\psi}_\tau \\ \ddot{x}_\tau \\ \ddot{y}_\tau \end{bmatrix}, \quad \mathbf{z}_\tau = \begin{bmatrix} x_\tau^r \\ y_\tau^r \\ \dot{x}_\tau^w \\ \dot{\psi}_\tau^w \\ \psi_\tau^m \\ \dot{\psi}_\tau^g \\ \ddot{x}_\tau^a \\ \ddot{y}_\tau^a \end{bmatrix} \quad (8)$$

where  $\tau$  is the continuous time,  $(x, y)$  are the position,  $\psi$  is the yaw angle,  $\cdot$  denotes the velocity, and  $\ddot{\cdot}$  denotes the acceleration. The estimation of a robot’s full pose at a discrete timestep  $t$  can be described with the state equation:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_{t-1} \quad (9)$$

where  $\mathbf{w}_{t-1}$  is the process noise. The measurement model represents the measurements we receive from our sensors (wheel encoder, IMU, and GPR) at each timestep:

$$\mathbf{z}_t = h(\mathbf{x}_t) + \mathbf{v}_t \quad (10)$$

where  $\mathbf{v}_t$  is the measurement noise.

We use the EKF implementation in the robot operating system (ROS) package `robot_localization` [19].

### 1) WHEEL ODOMETRY

The following differential drive integration method is used to calculate the linear ( $\dot{x}^w$ ) and rotational ( $\dot{\psi}^w$ ) velocities of the vehicle in the robot body frame using the encoder values of all four wheels. We calculate the left and right wheel velocities by  $v_{l/r} = \frac{\Delta e_{l/r}}{dt} R$ , where  $R$  is the wheel radius,  $dt$  represents the time duration since the previous time step, and  $\Delta e_{l/r}$  represents the average change in the wheel encoder tick value between the front and back wheel encoders since the previous time step. We use the following equations to calculate the vehicle's linear and angular velocities:

$$\dot{x}^w = \frac{v_r + v_l}{2} \quad (11)$$

$$\dot{\psi}^w = \frac{v_r - v_l}{W} \quad (12)$$

where  $W$  is the wheel separation. The velocities are then integrated into a position measurement in the filter.

### 2) INERTIAL MEASUREMENT UNIT

The IMU directly measures linear acceleration from the accelerometer, angular velocity from the gyroscope, and orientation from the magnetometer. We only fuse the 2-D linear acceleration, yaw angular velocity, and yaw angle measurements ( $\dot{x}^a, \dot{y}^a, \dot{\psi}^g, \psi^m$ ) according to our localized 2-D environmental assumption.

### 3) GPR RELATIVE DISPLACEMENT

We utilize a neural network, as described in Section III-B, to estimate displacement in the  $x$ -axis of the robot body frame. A set of  $k$  consecutive traces  $\mathbf{G} = \{\mathbf{g}_i, \dots, \mathbf{g}_{i+k}\}$ , with each  $\mathbf{g} \in \mathbb{R}^{200}$ , is passed through the model weights to predict  $\Delta d$ , the linear displacement:

$$\Delta d = \Phi(\mathbf{G}). \quad (13)$$

The number of traces used,  $k$ , is tuned as a hyperparameter alongside our network architecture hyperparameters. In order to integrate this into our measurement vector, we convert the relative displacement to an  $(x, y)$  coordinate using the corresponding  $z$ -axis rotational angle  $\psi_t^m$  from the IMU:

$$x_t^r = x_{t-1}^r + \Delta d \cos \psi_t^m \quad (14)$$

$$y_t^r = y_{t-1}^r + \Delta d \sin \psi_t^m. \quad (15)$$

We tune the covariance of the GPR network prediction to be higher while the robot is turning, as the displacement measured while rotating is dependent on where the GPR is mounted on the axis of rotation, which is a platform-specific setup parameter.

## IV. DATASET

In order to evaluate a use case for GPR-based localization for planetary rovers, it was necessary to collect a dataset at a site that is analog to the subterranean features that may be present on Mars. As previously outlined in Table 1, the two existing LGPR datasets [4], [5] were collected on roads or in

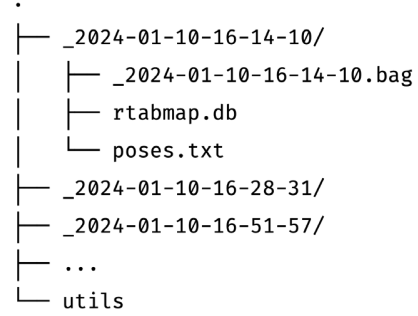


FIGURE 5. Overview of the dataset organization. There are 50 sequences, totaling around 1 TB of data.

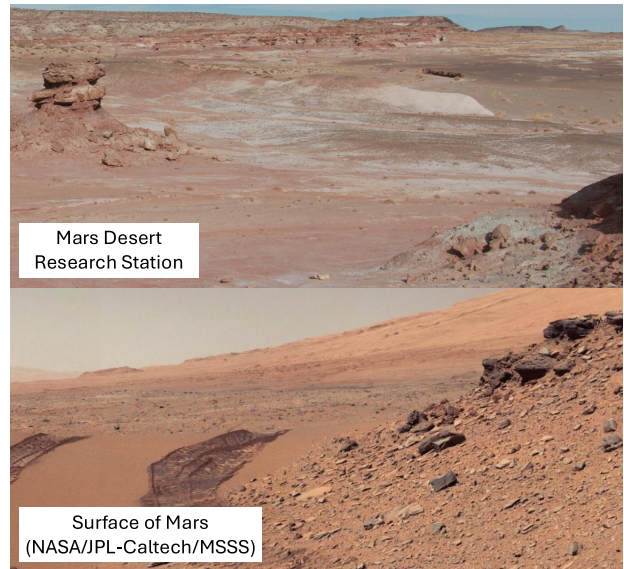


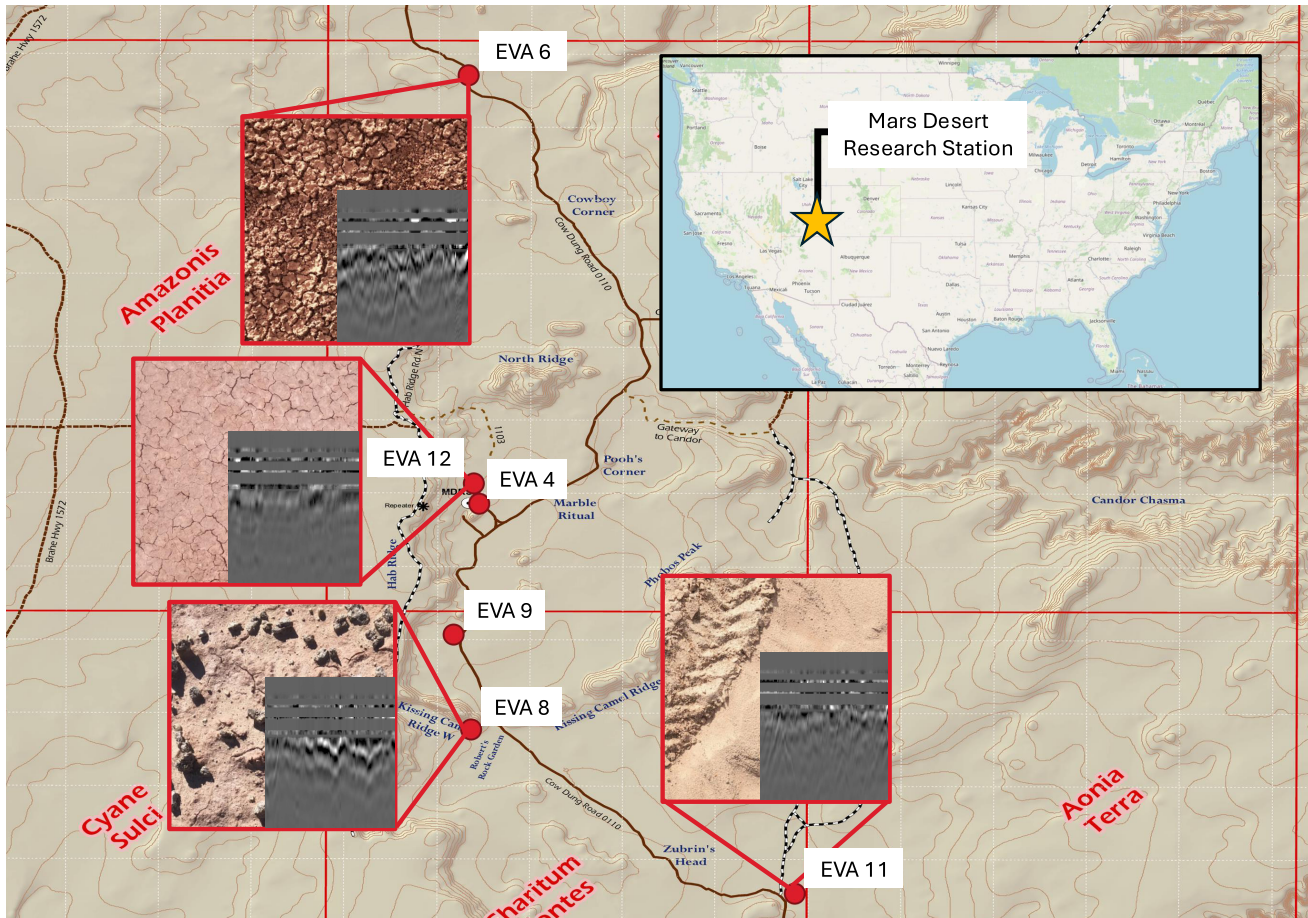
FIGURE 6. Visual comparison of geologic features between the MDRS region (top) and Mars (bottom).

parking garages, which made them unsuitable for determining whether LGPR on Mars, the Moon, or rugged terrain is possible.

We collected a new dataset, MarsLGPR, at the Mars Desert Research Station (MDRS) near Hanksville, UT, USA, over the course of 10 days in January 2024. The resulting dataset contains around 1 TB of GPR, stereo camera, IMU, and robot wheel encoder data, which will be released with the directory structure shown in Fig. 5.

MDRS was chosen as a site for data collection due to its similarity to Martian geology (see Fig. 6). A map of the collection sites is shown in Fig. 7. The area is uniquely devoid of vegetation and has evidence of similar geological processes as Mars, such as fluvial action, layered sediment deposits, presence of clays and salts, and sandy dunes [42].

The data collection platform was a Clearpath Husky A200, specially outfitted with a NOGGIN 500 GPR sensor. An ZED2i stereo camera with polarizer film was mounted on the front of the robot, which produced RGB images, depth

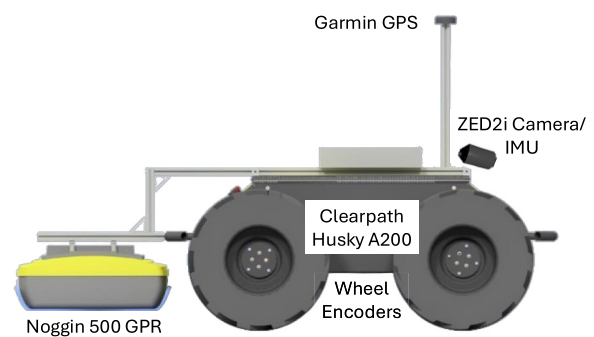


**FIGURE 7.** Areas of dataset collection in the region surrounding MDRS in southeast Utah. Note the variety of terrain types, shown in the red cutouts, which provide a wide distribution of GPR signals for training the relative displacement network. A snapshot sample of a filtered GPR B-scan at each site is shown in the terrain image cutout. Base MDRS map provided by the Mars Society, and cutaway map of United States provided by OpenStreetMap.

images, and IMU data. The Husky wheel encoders track 78 000 ticks per meter. Wheel separation of the Clearpath Husky is  $W = 0.165$  m, and the wheel radius is  $R = 0.5455$  m. A custom ROS package was used to interface with the NOGGIN GPR sensor, which samples at a center radio frequency of 500 MHz. A diagram of the data collection platform is shown in Fig. 8.

For each sequence, the vehicle was remotely controlled and driven around various terrain types, including sandy aeolian plains, cracked clays, and rocky washes. ZED stereo camera data were collected at 5 Hz and a resolution of  $1280 \times 720$  pixels. GPR data were collected at 1.67 Hz in stacks of three repetitive samples to provide better control for noise.

Note that we also collected GPS data on a Garmin 18x unit; however, due to an issue in the parsing of satellite-based augmentation signals, the GPS data are not good enough to provide ground-truth localization. Instead, we provide localizations from a map built and optimized offline by RTAB-Map [43], a popular VO and 3-D reconstruction tool. VO is a well-posed task for the images in our dataset, as there



**FIGURE 8.** Diagram of custom Clearpath Husky A200 dataset collection platform, which was modified to include the NOGGIN 500 GPR sensor, a Garmin 18x module, and a ZED2i stereo camera.

is little camera motion blur and a large number of trackable features. These reference poses from VO are optimized offline, which is a computationally costly task and is infeasible to run in real time on spaceflight hardware. However,

**TABLE 2. Cumulative displacement from the RTAB-Map (reference), Wick.\*, wheel encoder, and GPRFormer network predictions using the RMSE metric in millimeters on the CMU-GPR dataset. Note that the CMU-GPR dataset was collected on flat concrete surfaces with little slip, making the wheel encoder very accurate. Bold is best.**

Method	CMU-GPR Dataset [5] (RMSE ↓)				MarsLGPR Dataset (RMSE ↓)			
	Seq 1	Seq 3	Seq 7	Seq 9	12-17-01-44	15-14-58-25	15-15-42-09	15-15-55-14
Wheel Encoder	9.72	<b>4.12</b>	<b>4.20</b>	<b>3.55</b>	20.65	27.42	29.89	29.62
Wick.* [34]	22.02	18.32	14.54	14.09	-	23.56	22.46	19.37
GPRFormer (Ours)	<b>8.62</b>	7.64	6.51	5.98	<b>15.59</b>	<b>13.32</b>	<b>16.92</b>	<b>17.58</b>

we are able to use it as a reference trajectory for our model training and experiments in the absence of GPS.

In order to prepare the MarsLGPR dataset for training, we first remove anomalous traces, where the recorded values are outside of the  $\pm 50$ -mV range, and then we average the consecutive stacked traces to help reduce noise. In addition, we perform linear and SLERP interpolation along the RTAB-Map poses to have time-synchronized sensor and positional data, which is important for model training.

## V. EXPERIMENTS AND RESULTS

In this section, we present experiments to validate our method qualitatively and quantitatively. We first present accuracy metrics on GPRFormer and two baselines: the wheel encoders and localization network Wickramanyake [34]. Then, we qualitatively and quantitatively evaluate the GPRFormer + EKF localization framework. In addition, we include an ablation study and runtime metrics on GPRFormer. Experiments are performed on both the publicly available CMU-GPR dataset [5] and on our own MarsLGPR dataset.

### A. GPRFormer IMPLEMENTATION DETAILS

We train two versions of our GPRFormer model, one on the CMU-GPR dataset and other on the MarsLGPR dataset. For the CMU-GPR dataset, we follow [34] and use sequences 0, 2, 4, 5, 6, 8, and 10 for training, and sequences 1, 3, 7, and 9 for testing. We apply background removal, dewow filtering, SEC, and wavelet denoising to enhance feature visibility. During training, the model is supervised with displacement calculated from the ground-truth RTK GPS positioning data for the CMU-GPR dataset and from reference RTAB-Map poses for the MarsLGPR dataset. The MarsLGPR training set contains 35 sequences, and the testing set contains four sequences.

The Adam optimizer is used with a batch size of 128, and a linear learning rate scheduler helps with better fine-tuning later in the training process. All training is performed on an NVIDIA GeForce 32-GB RTX 3090 GPU.

### B. BASELINE METHODS

As no code is available for the Wickramanyake et al.'s [34] baseline, we have reimplemented this method from scratch in order to use it as a comparison. We differentiate our

reimplemented version of this baseline as Wick.\*. We directly follow the parameters provided in their article, with a learning rate of 0.0001, an MSE loss function, the Adam optimizer, a batch size of 100, and the same training/testing split. We trained the model for 250 epochs on the CMU-GPR dataset and 60 epochs on the MarsLGPR dataset, after which the MSE loss was no longer reduced. This article reports batch MSE values of 0.266, 0.403, 0.115, and 0.143, our replicated version of the CNN-Long Short-Term Memory (LSTM) had batch MSE values of 0.370, 0.226, 0.123, and 0.105 for sequences 1, 3, 7, and 9, respectively.

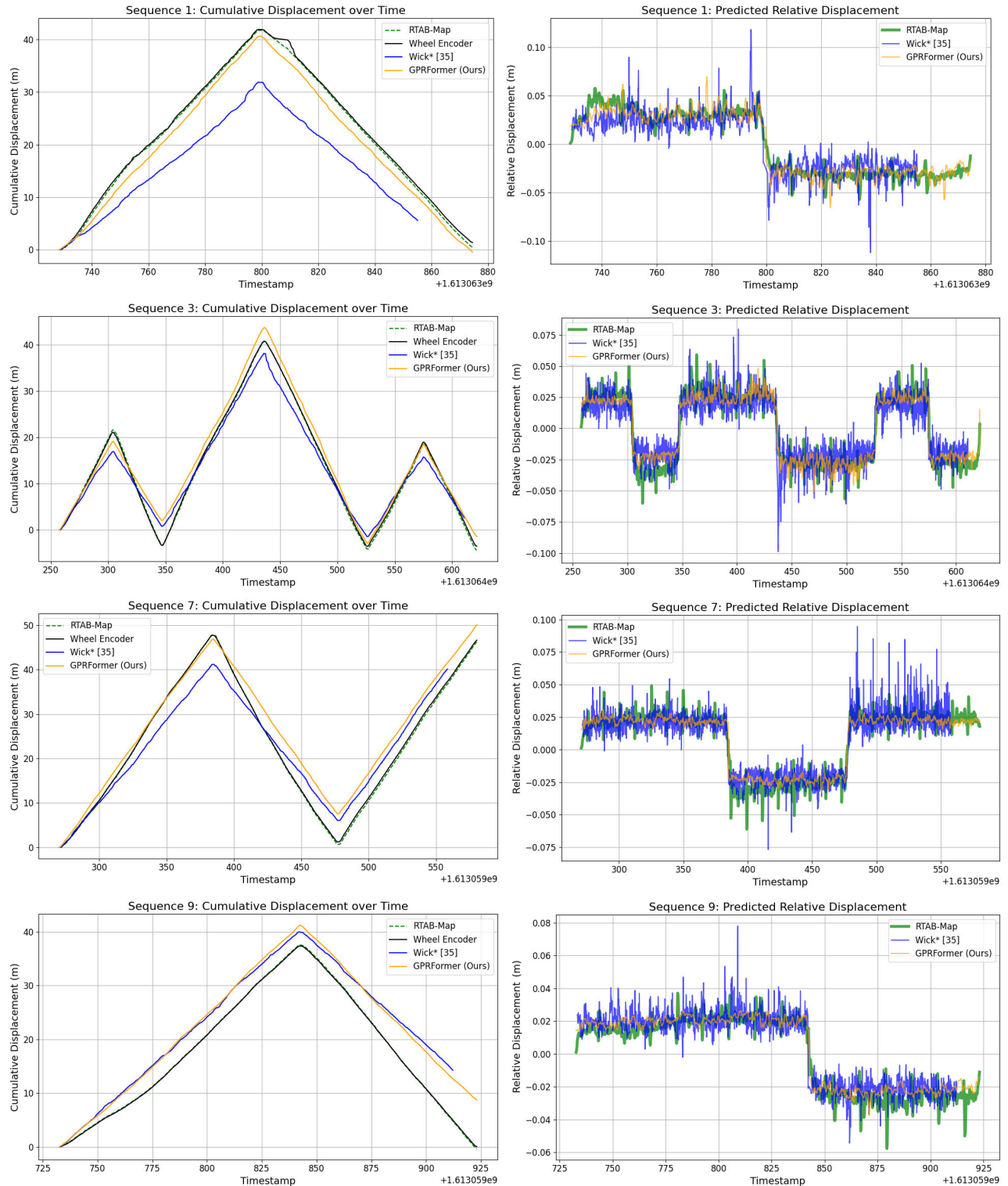
### C. GPRFormer ACCURACY

We evaluate model performance using the Root MSE (RMSE) of displacement estimates, reported in millimeters. RMSE is given by:

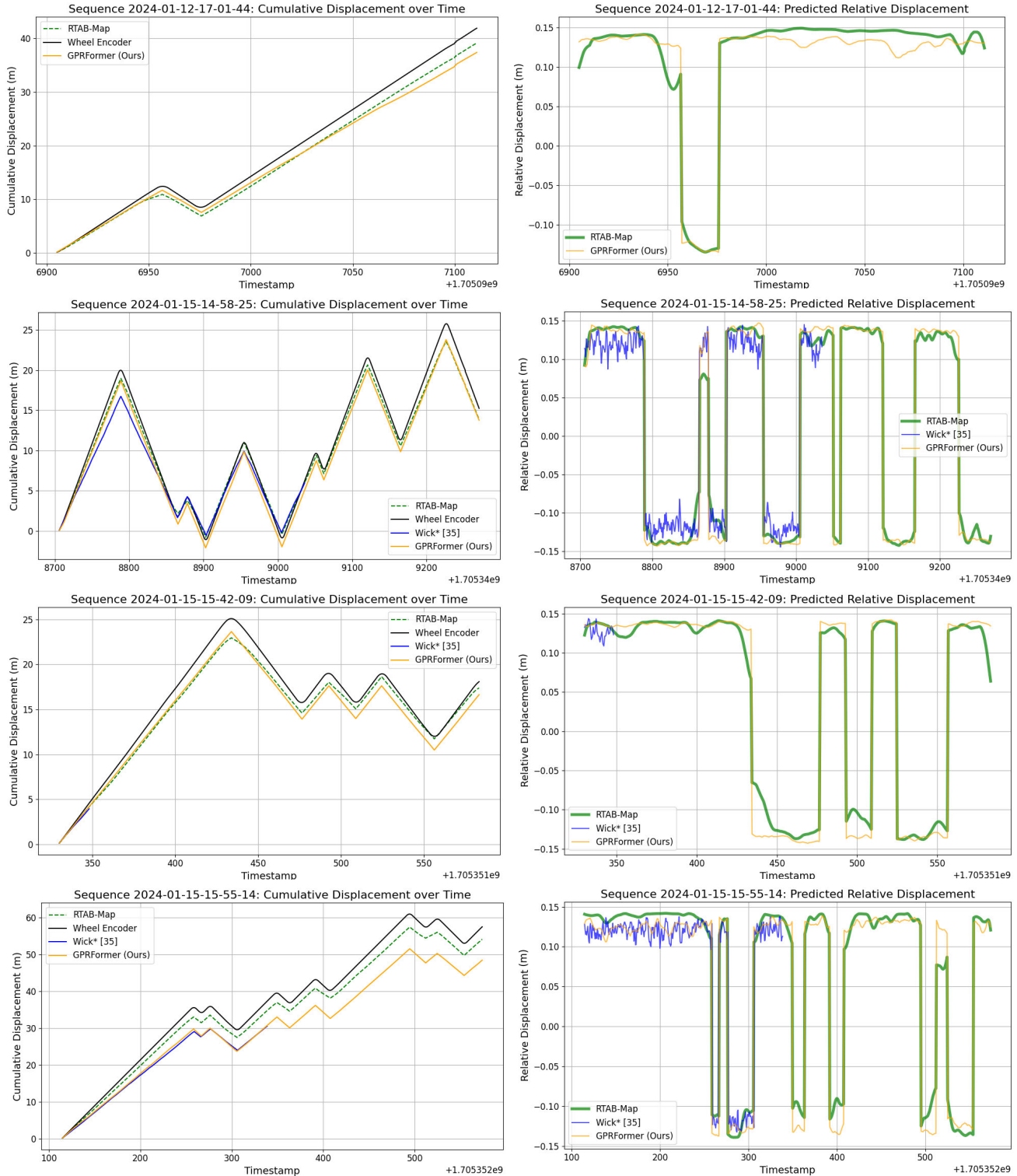
$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2} \quad (16)$$

where  $y_t$  is the ground-truth displacement,  $\hat{y}_t$  is the predicted displacement, and  $T$  is the prediction window. For GPRFormer,  $T = k = 10$ , and for Wick.\*,  $T = 2$ . Since the two models generate predictions in overlapping windows of length  $T$ , we apply an overlap-add accumulation procedure to obtain a consistent per-timestep estimate: each window's prediction is distributed evenly across its span and averaged with other overlapping contributions, producing a single-displacement estimate per timestep. This binning ensures that different prediction granularities are made directly comparable and also makes the predictions more robust to outlier results from the model.

Table 2 shows RMSE computed across the cumulative displacement for each test sequence in the CMU-GPR dataset and the MarsLGPR dataset. It is important to note that the wheel encoder data in the CMU-GPR dataset is very accurate, as demonstrated by the strong quantitative results. This is because the dataset collection occurred on concrete surfaces, where slip was minimal, and there were no slopes. The Mars analog terrain in MarsLGPR proves more challenging for wheel encoder odometry estimation. GPRFormer demonstrates improvement over wheel encoder odometry for the more challenging MarsLGPR dataset. Our method, GPRFormer, outperforms the reimplemented Wick.\* method on all eight test sequences across both datasets.



**FIGURE 9.** Comparison of GPRFormer relative displacement predictions against Wick.\* [34], the wheel encoders, and the ground truth on four test sequences from CMU-GPR [5]. The left column shows the cumulative displacement over time, and the right column shows the accompanying predictions for the displacement at each time step.



**FIGURE 10.** Comparison of GPRFormer relative displacement predictions against Wick\* [34], the wheel encoders, and RTAB-Map on four test sequences from the MarsLGPR dataset. The left column shows the cumulative displacement over time, and the right column shows the accompanying predictions for the displacement at each time step. Note that the wheel encoders accumulate more displacement over time. This is due to the slippery terrain causing the wheels to spin out.

**TABLE 3. Performance metrics (memory and runtime) for Wick.\* and GPRFormer model inference.**

Model	Avg. inference runtime (s)		Avg. memory (GB)	
	CPU	GPU	CPU	GPU
Wick.* [34]	0.0706	0.0640	<b>1.23</b>	2.66
Ours	<b>0.0034</b>	<b>0.0032</b>	1.90	<b>1.39</b>

Fig. 9 displays qualitative results from the ground truth, the wheel encoder baseline, the Wick.\* method baseline, and our model GPRFormer on the CMU-GPR dataset. The right-hand column shows the individual GPRFormer predictions at each timestep compared with the displacement from the ground truth and wheel encoders. This allows us to inspect whether the network is able to learn local trends and patterns from the GPR data. The Wick.\* predictions are very noisy, which is reflected in a higher RMSE metric in Table 2. The left-hand column shows the cumulative displacement over the trajectory, with the direction of the wheel encoder (positive or negative) indicating the direction of movement. Note that in all eight plots, Wick.\* has no results on the last section of the trajectory because the method relies on building up 202 consecutive GPR A-scans before making a displacement prediction.

Fig. 10 shows qualitative results for GPRFormer and Wick.\* on MarsLGPR test sequences. Note how the wheel encoder in the MarsLGPR sequences is much less accurate than the wheel encoder in the CMU-GPR dataset, which makes it a much less reliable source of odometry. Our GPRFormer network visibly outperforms the wheel encoder displacement as compared with the RTAB-Map reference, and this is also reflected quantitatively in Table 2. We note that in Sequence 12-17-01-44, there were not enough GPR traces for the Wick.\* method to make any predictions.

#### D. GPRFormer RUNTIME PERFORMANCE

In this experiment, we discuss the ability of the GPRFormer model to operate in real-time. We present Table 3, which shows the runtime per sample in seconds for our method and baseline Wick.\* [34]. We show the runtime for inference on both GPU and CPU, as GPUs are typically inaccessible in space environments. Both methods have very fast runtimes, on the order of hundreds of Hz, although our method outperforms the Wick.\* model.

Although the model runtimes are very fast, the most critical bottleneck for performance is the GPR data rate. For the MarsLGPR dataset, GPR data are collected at 1.67 Hz, meaning that our model (which requires ten consecutive samples) could begin inference after 6 s, and then, every half second after that. The Wickramanayake method requires two patches of 200 traces, with two traces in between, for a total of 202 traces (over 120 s of data collection) in order to begin inference.

#### E. EKF STATE ESTIMATION ACCURACY

We also demonstrate the resulting full state estimation results when integrating the GPRFormer predictions along with IMU yaw angle into an EKF to estimate  $x$  and  $y$  position. As GPRFormer is only able to predict 1-D displacement, we increase the covariance of the GPR network predictions when the robot is rotating, as determined by the wheel encoder directions. Table 4 shows a comparison of EKF with and without the GPRFormer integrated in. We calculate the RMSE Absolute Trajectory Error (ATE) metric on both trajectories in relation to the ground truth after aligning them in the yaw axis.

For the CMU-GPR dataset, we note that the GPRFormer EKF only very slightly outperforms the wheel encoder baseline. This is because the wheel encoders are very accurate in this dataset, as the vehicle only drove over smooth covered surfaces. For the MarsLGPR dataset, where there is confirmed wheel slippage on various rough terrains, we see a large improvement in the RMSE ATE. For one of the sequences, the error was cut in half after including GPR relative localization. The qualitative results in Fig. 11 further confirm these results. We see that the inclusion of GPRFormer on the MarsLGPR sequence prevented the filter from overshooting the true trajectory due to wheel slippage.

The localization update rate is about 15 Hz, which is a satisfactory real-time rate for most applications.

#### F. ABLATION STUDIES

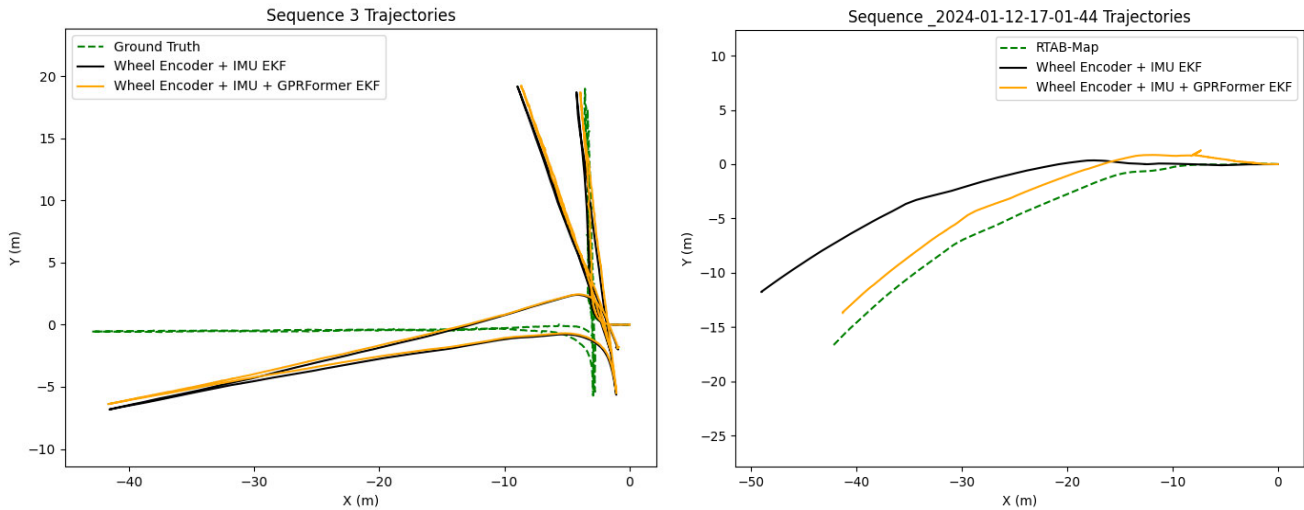
In this experiment, we investigate how different components of the network impact the overall model performance. We train five ablated versions of the model on the MarsLGPR dataset, with two additional experiments to further understand the  $k$  and  $\alpha$  parameters. See Table 5 for a comparison of the performance of each ablation. The metric used in this experiment is RMSE in millimeters across all four MDRS test sequences, where lower is better.

##### 1) B-SCAN FILTER PREPROCESSING

This preprocessing step helps to make features in the B-scan more distinct and visible. We see in the ablation Table 5 that not including this preprocessing step does make the results of the network worse, but not drastically so. This is likely because the network is still able to learn to find the finer-grained features in the B-scan even without the filtering. However, the inclusion of B-scan filtering likely makes the model learn more efficiently, resulting in an improvement when filtering is used.

##### 2) LINEAR ENCODER

Similarly, removing the linear encoder from the GPRFormer architecture has a moderate effect on the overall network performance, slightly increasing the RMSE. This experiment demonstrates that the B-scan representation of the data is already quite compact and structured, which allows the model to learn well despite having no encoder. However, adding a



**FIGURE 11.** Qualitative comparison of filtered trajectories and ground truth on sequence 3 from the CMU-GPR dataset and sequence 12-17-01-44 from the MarsLGPR dataset. Note that the MarsLGPR dataset sequence shows much more improvement over the wheel encoder + IMU EKF due to the slippery terrain which hindered the wheel encoders.

**TABLE 4.** Comparison of localization EKF with and without the GPRFormer relative displacement prediction. Trajectory accuracy is measured with RMSE ATE in meters, with a lower value indicating a closer trajectory to the ground truth. Bold is best.

Method	CMU-GPR Dataset [13] (RMSE ATE ↓)		MarsLGPR Dataset (RMSE ATE ↓)	
	Sequence 3	Sequence 7	Sequence 01-12-17-01-44	Sequence 01-15-14-58-25
Encoder + IMU EKF	3.010	2.113	4.947	8.485
Encoder + IMU + GPRFormer EKF	<b>2.903</b>	<b>2.100</b>	<b>2.576</b>	<b>8.136</b>

**TABLE 5.** Ablation study on GPRFormer network with results in RMSE (millimeters) across all four MarsLGPR test sequences. Bold is best.

B-scan filter	encoder	dropout	seq pooling	# layers	RMSE (↓)
✓	✓	✓	✓	✓	<b>15.57</b>
	✓	✓	✓	✓	16.58
✓		✓	✓	✓	16.94
✓	✓		✓	✓	16.51
✓	✓	✓		✓	18.25
✓	✓	✓	✓		20.37

linear encoder layer does allow for improved results by letting the model learn a more efficient latent space.

### 3) DROPOUT

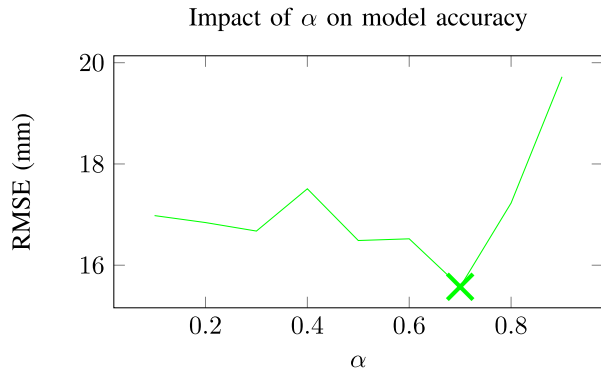
Dropout is an important technique for improving network generalization and avoiding overfitting. In this variation, we reduce the dropout rate from 0.1 to 0 in order to understand the effects of this component in the architecture, which causes a moderate drop in network performance. The worst GPRFormer performance without the dropout module is

likely caused by overfitting on the training set that occurs without this regularization.

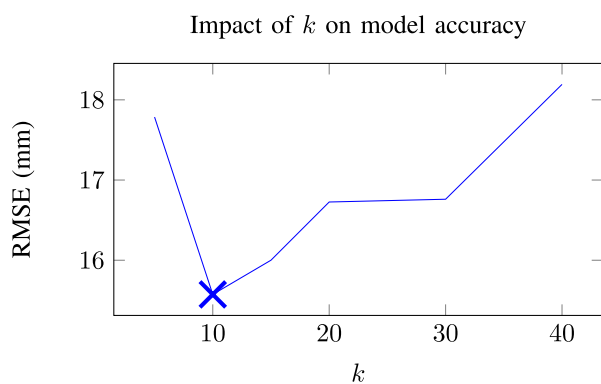
### 4) DUAL-SEQUENCE POOLING

As discussed in the technical approach of this article, dual-sequence pooling is a useful technique to allow the GPRFormer model to benefit from both the pre-transformer encoding and the post-transformer encoding by blending attention weights. In this experiment, we completely remove this from the network, which causes the network to strictly use the transformer encoder latent space. This greatly affects the performance of the model, with the average cumulative RMSE increasing to over 18 mm. This helps to confirm that the inclusion of this attention weight pooling step, which balances between raw patterns and transformer-encoded patterns in the data, is a key component of the GPRFormer network.

In addition, we include a further experiment on the optimal  $\alpha$  value, which is the weighting parameter between the pre-transformer and post-transformer features.  $\alpha$  is a learned term, but for this experiment, we fixed it to a static value in order to understand the impact of the term on the model performance. We see in Fig. 12 that when  $\alpha$  is at a value of 0.7—which weights the post-transformer features a little higher than pre-transformer features—it has the best



**FIGURE 12.** Comparison of different  $\alpha$  (dual-sequence pooling weight) values and their impact on the model accuracy, as measured in RMSE. Note that an  $\alpha = 0.7$  has the best results.



**FIGURE 13.** Comparison of different  $k$  (B-scan length) values and their impact on the model accuracy, as measured in RMSE. Note that a  $k = 10$  has the best results.

performance in terms of RMSE. When  $\alpha$  is unfrozen during normal training, it converges to 0.70102, matching the results of this experiment.

##### 5) MULTILAYER TRANSFORMER ENCODER

In this experiment, we drop the number of layers and attention heads to just one, and we see that this has the most drastic effect on the network performance out of all of the ablations, with an increase in RMSE to over 20 mm. The transformer encoder is one of the core components of the network, and it has enough complexity that it can compensate for some of the other ablation experiments (for example, removing the B-scan filtering step). However, when this block in the network is simplified too far, it loses much of its ability to perform well.

##### 6) B-SCAN LENGTH $k$

In this experiment, we experimentally validate our choice to use a  $k$  (B-scan length) of 10. In Fig. 13, there is a clear minimum RMSE at  $k = 10$ , which is the value we used in our network design. As discussed in this article sensor model, GPR has a very wide beamwidth, resulting in spatial overlap with adjacent A-scans. Stacking several

traces together provides enough contextual information to the model about subsurface spatial features that the model is able to predict displacement. However, for the GPRFormer architecture, the tradeoff between a larger B-scan and better performance starts to degrade after  $k = 10$ . The larger input makes it more difficult for the network to isolate meaningful features.

## VI. CONCLUSION AND FUTURE WORK

In this work, we proposed a novel framework to leverage GPR sensors aboard planetary rovers to aid in localization. Through our proposed network architecture, we demonstrate the ability of GPRFormer to learn relative displacement from GPR measurements on various types of Martian analog rugged terrain. We compare our method to both a wheel encoder and a learned baseline [34], and GPRFormer achieves superior results. In addition, we integrate our prediction network into an EKF sensor fusion framework in order to provide efficient and accurate robot pose estimation. Our GPRFormer EKF outperforms the wheel encoder + IMU only filter, especially in areas of high wheel slippage. We will publicly release our MarsLGPR dataset and GPRFormer code to enable further work in GPR localization for Mars analog environments.

One limitation of the filtering-based framework that we use in this work is the need to carefully tune covariance matrices, which represent uncertainty from GPRFormer. Future work would include an investigation into quantifying the uncertainty from the GPRFormer model in order to provide dynamic and intelligent covariance matrices for uncertainty propagation in the filter. In addition, we note that recent advances in 3-D neural representations with novel sensors, such as NeRFs with synthetic aperture radar [44] as well as 3-D Gaussian splatting with radar [45] and forward-looking sonar [46], motivate further work in GPR feature maps for localization.

## ACKNOWLEDGMENT

The authors would like to thank the mission support crew at MDRS and the Mars Society, namely, Ben Stanley, for their support during field operations. In addition, they would also like to thank Madelyn MacRobbie, Rebecca McCallin, Ben Kazimer, Anna Tretiakova, and Wing Lam Chan for their assistance in conducting field data collection.

## REFERENCES

- [1] R. Li et al., "Spirit rover localization and topographic mapping at the landing site of Gusev crater, Mars," *J. Geophys. Res., Planets*, vol. 111, no. E2, p. 13, Feb. 2006.
- [2] V. Verma et al., "Enabling long & precise drives for the perseverance Mars rover via onboard global localization," in *Proc. IEEE Aerosp. Conf.*, Mar. 2024, pp. 1–18.
- [3] J. Nash et al., "Censible: A robust and practical global localization framework for planetary surface missions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 8642–8648.
- [4] T. Ort, I. Gilitschenski, and D. Rus, "GROUNDED: A localizing ground penetrating radar evaluation dataset for learning to localize in inclement weather," *Int. J. Robot. Res.*, vol. 42, no. 10, pp. 901–916, Sep. 2023.

- [5] A. Baikovitz, P. Sodhi, M. Dille, and M. Kaess, "CMU-GPR dataset: Ground penetrating radar dataset for robot localization and mapping," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2021. [Online]. Available: <https://arxiv.org/abs/2107.07606>
- [6] J. Townsend et al., "Mars exploration rovers 2004-2013: Evolving operational tactics driven by aging robotic systems," in *Proc. SpaceOps Conf.*, 2014, p. 1884.
- [7] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the Mars exploration rovers," *J. Field Robot.*, vol. 24, no. 3, pp. 169–186, Mar. 2007.
- [8] M. Maimone, N. Patel, A. Sabel, A. Holloway, and A. Rankin, "Visual odometry thinking while driving for the curiosity Mars Rover's three-year test campaign: Impact of evolving constraints on verification and validation," in *Proc. IEEE Aerosp. Conf. (AERO)*, Mar. 2022, pp. 1–10.
- [9] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.
- [10] A. Kraut and D. Wettergreen, "Classification of Mars terrain using multiple data sources," in *Proc. CIDU*, 2010, pp. 54–68.
- [11] C. Cunningham, I. A. Nenas, and W. L. Whittaker, "Improving slip prediction on Mars using thermal inertia measurements," *Auto. Robots*, vol. 43, no. 2, pp. 503–521, Feb. 2019.
- [12] T. Ort, I. Gilitschenski, and D. Rus, "Autonomous navigation in inclement weather based on a localizing ground penetrating radar," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3267–3274, Apr. 2020.
- [13] A. Baikovitz, P. Sodhi, M. Dille, and M. Kaess, "Ground encoding: Learned factor graph-based models for localizing ground penetrating radar," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 5476–5483.
- [14] A. Sheppard, J. I. Brown, N. Renno, and K. A. Skinner, "Learning surface terrain classifications from ground penetrating radar," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 3047–3055.
- [15] S.-E. Hamran et al., "Radar imager for Mars' subsurface experiment—RIMFAX," *Space Sci. Rev.*, vol. 216, no. 8, pp. 1–39, Dec. 2020.
- [16] V. Ciarletti et al., "The WISDOM radar: Unveiling the subsurface beneath the ExoMars rover and identifying the best locations for drilling," *Astrobiology*, vol. 17, nos. 6–7, pp. 565–584, Jul. 2017.
- [17] J. P. de la Croix et al., "Multi-agent autonomy for space exploration on the CADRE lunar technology demonstration," in *Proc. IEEE Aerosp. Conf.*, Mar. 2024, pp. 1–14.
- [18] L. Jetto, S. Longhi, and G. Venturini, "Development and experimental validation of an adaptive extended Kalman filter for the localization of mobile robots," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 219–229, Apr. 1999.
- [19] T. Moore and D. Stouch, "A generalized extended Kalman filter implementation for the robot operating system," in *Proc. 13th Int. Conf. Intell. Auton. Syst.* Cham, Switzerland: Springer, 2016, pp. 335–348.
- [20] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Inst. Technol., Tech. Rep. GT-RIM-CP&R-2012-002, 2012, vol. 2, no. 4. [Online]. Available: <https://repository.gatech.edu/server/api/core/bitstreams/b3606eb4-ce55-4c16-8495-767bd46f0351/content>
- [21] W. Wai-Lok Lai, X. Dérobert, and P. Annan, "A review of ground penetrating radar application in civil engineering: A 30-year journey from locating and testing to imaging and diagnosis," *NDT E Int.*, vol. 96, pp. 58–78, Jun. 2018.
- [22] C. J. Vaughan, "Ground-penetrating radar surveys used in archaeological investigations," *Geophysics*, vol. 51, no. 3, pp. 595–604, Mar. 1986.
- [23] R. Knight, "Ground penetrating radar for environmental applications," *Annu. Rev. Earth Planet. Sci.*, vol. 29, no. 1, pp. 229–255, 2001.
- [24] G. R. Olhoef, "Ground penetrating radar on Mars," in *Proc. 7th Int. Conf. (GPR)*, 1998, pp. 387–392.
- [25] E. Skartados, A. Kargakos, E. Tsiogas, I. Kostavelis, D. Giakoumis, and D. Tzovaras, "GPR antenna localization based on A-scans," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5.
- [26] B. Stanley, M. Cornick, and J. C. Koechling, "Ground penetrating radar based localization," in *Proc. Ground Vehicle Syst. Eng. Technol. Symp.*, vol. 1, 2024. [Online]. Available: <https://web.archive.org/web/20250218220354/http://gvsets.ndia-mich.org/documents/AAIR/2013/Ground%20Penetrating%20Radar%20Based%20Localization.pdf>
- [27] M. Cornick, J. Koechling, B. Stanley, and B. Zhang, "Localizing ground penetrating RADAR: A step toward robust autonomous ground vehicle localization," *J. Field Robot.*, vol. 33, no. 1, pp. 82–102, Jan. 2016.
- [28] Z. Ni, S. Ye, C. Shi, J. Pan, Z. Zheng, and G. Fang, "A deep learning assisted ground penetrating radar localization method," *J. Electron. Inf. Technol.*, vol. 44, no. 4, pp. 1265–1273, 2022.
- [29] B. Bi, L. Shen, P. Zhang, X. Huang, Q. Xin, and T. Jin, "TSVR-Net: An end-to-end ground-penetrating radar images registration and location network," *Remote Sens.*, vol. 15, no. 13, p. 3428, Jul. 2023.
- [30] P. Zhang et al., "Looking beneath more: A sequence-based localizing ground penetrating radar framework," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 8515–8521.
- [31] B. Bi, Y. Chen, L. Shen, P. Zhang, X. Huang, and T. Jin, "GPR image retrieval and localization based on spatial prior constraints," *J. Phys.: Conf. Ser.*, vol. 2887, no. 1, Nov. 2024, Art. no. 012016.
- [32] J. Xu, Q. Lai, D. Wei, X. Ji, G. Shen, and H. Yuan, "The ground-penetrating radar image matching method based on central dense structure context features," *Remote Sens.*, vol. 16, no. 22, p. 4291, Nov. 2024.
- [33] H. Li, J. Guo, and D. Song, "Subsurface feature-based ground robot/vehicle localization using a ground penetrating radar," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 1716–1722.
- [34] S. Wickramanayake, K. Thiyagarajan, and S. Kodagoda, "Deep learned ground penetrating radar subsurface features for robot localization," in *Proc. IEEE Sensors*, Oct. 2022, pp. 1–4.
- [35] E. C. Utsi, *Ground Penetrating Radar: Theory and Practice*. Oxford, U.K.: Butterworth-Heinemann, 2017.
- [36] A. Hassani, S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi, "Escaping the big data paradigm with compact transformers," 2021, *arXiv:2104.05704*.
- [37] B. Maruddani and E. Sandi, "The development of ground penetrating radar (GPR) data processing," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 6, pp. 768–773, Dec. 2019.
- [38] N. J. Cassidy, *Ground Penetrating Radar Theory and Applications*. Amsterdam, The Netherlands: Elsevier, 2008, pp. 128–161.
- [39] J. Baili, S. Lahouar, M. Hergli, I. L. Al-Qadi, and K. Besbes, "GPR signal de-noising by discrete wavelet transform," *NDT E Int.*, vol. 42, no. 8, pp. 696–703, Dec. 2009.
- [40] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5998–6008.
- [41] K. Leng, C. Yang, S. Wei, J. Liu, and Z. Li, "Sitpose: A Siamese convolutional transformer for relative camera pose estimation," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2023, pp. 1871–1876.
- [42] C. R. Stoker et al., "Mineralogical, chemical, organic and microbial properties of subsurface soil cores from Mars desert research station (Utah, USA): Phyllosilicate and sulfate analogues to Mars mission landing sites," *Int. J. Astrobiol.*, vol. 10, no. 3, pp. 269–289, Jul. 2011.
- [43] M. Labbe and F. Michaud, "Appearance-based loop closure detection for online large-scale and long-term operation," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 734–745, Jun. 2013.
- [44] Z. Lei, F. Xu, J. Wei, F. Cai, F. Wang, and Y.-Q. Jin, "SAR-NeRF: Neural radiance fields for synthetic aperture radar multiview representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, doi: 10.1109/TGRS.2024.3452505. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=10659918>
- [45] P.-C. Kung, S. Harisha, R. Vasudevan, A. Eid, and K. A. Skinner, "Radar-Splat: Radar Gaussian splatting for high-fidelity data synthesis and 3D reconstruction of autonomous driving scenes," 2025, *arXiv:2506.01379*.
- [46] A. V. Sethuraman, M. Rucker, O. Bagoren, P.-C. Kung, N. N. B. Amutha, and K. A. Skinner, "SonarSplat: Novel view synthesis of imaging sonar via Gaussian splatting," 2025, *arXiv:2504.00159*.

• • •