# High-resolution Algorithms for Distributed LiDAR-based Big Data Analysis of Ice-surface Morphological Characteristics of the Greenland Ice Sheet with an MLP-based Feature Classifier

by

**Adam B. Hayes**

B.A., University of Colorado, Boulder, 2021

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Masters of Computer Science

Department of Computer Science

2023

Committee Members:

Ute Herzfeld, Chair

Sriram Sankaranarayanan

Youjian Liu

Hayes, Adam B. (M.S., Computer Science)

High-resolution Algorithms for Distributed LiDAR-based Big Data Analysis of Ice-surface Morphological Characteristics of the Greenland Ice Sheet with an MLP-based Feature Classifier

Thesis directed by Dr. Ute Herzfeld

The ubiquity of computing crosses all disciplines in the modern world, and with a domain as important as the cryosphere during the era of global climate change, integrating modern methods in computer science becomes paramount. As Earth observation data becomes increasingly available in unprecedented size and scope, it necessitates methodology which can efficiently process and extract value from Big Data. This thesis focuses on Light Detection and Ranging (LiDAR) data collected in the cryosphere, specifically over the Greenland Ice Sheet (GrIS). In order to leverage modern, high-resolution sensory technology my advisor Dr. Herzfeld has developed the Density Dimension Algorithm for Ice Surfaces (DDA-ice) to extract high-resolution ice-surface morphological characteristics from LiDAR data. This work improves the computational efficiency and scalability of such an algorithm, as to not compromise resolution for scale with the goal of analyzing the surface morphological characteristics of the GrIS over time in an efficient manner.

By integrating parallel computing techniques and distributed systems I can leverage the DDA-ice to process seasonal renderings of the ice-sheet morphology of the GrIS at a pace which keeps up with satellite data collection. In order to further highlight the morphological characteristics of the ice I focus on crevasses as manifestations of the underlying ice dynamics. A physically-informed multi-layer perceptron (MLP) is implemented to classify different crevasse types across the GrIS from LiDAR data by using the DDA-ice and relevant geostatistical methodology.

## Dedication

To my sister Mado. For all the support and wisdom you've provided me over the years.

## Acknowledgements

This experience was made possible with the support I received from people along my journey. I would especially like to thank my advisor Ute Herzfeld for taking a chance on me and for constantly guiding and empowering me, and for providing me the resources and freedom to experiment and grow. Everyone who I have had the chance to work with in Dr. Herzfeld's lab over the last four years has helped me get to where I am today. I would especially like to thank my friend and colleague Tom Trantow for integrating me into the lab, and for his constant dedication and meaningful insight. I would also like to thank my colleagues and friends Matt Lawson and Jack Hessburg for guiding me into Computer Science, and always being willing to teach me something new. I also want to thank Huilin Han for being an extremely supportive and helpful friend and colleague throughout the last two years.

I am very grateful for the NASA ICESat-2 mission and the science team members and researchers who I have had the opportunity to work with over the last few years. Without your help and support I could not have completed this research. Special thanks to David Hancock, Stephen Palm, Kristine Barbieri, and the whole atmosphere research group for always being willing to work with me and providing guidance along the way.

I would like to thank my professor and committee member Sriram Sankaranarayanan for the effort and care he puts into teaching, and I sincerely want to thank he and Youjian Liu for taking the time and being willing to serve on my thesis committee.

Most importantly I would like to thank my friends and family for their constant love and support throughout this journey. Without their support I surely would not be here.

# Contents

**Chapter**

# Tables

**Table**

# Figures

**Figure**

# Chapter 1

# Introduction

The collection of Earth observation data from satellites has rapidly advanced with increasingly large and detailed Earth observation data sets becoming available for scientific investigations. This has lead to further accelerated progression and importance of modeling physical Earth processes, along with higher standards for doing so. The cryosphere is defined as the area on Earth where water is in its solid form (i.e. ice), and it is one of the most heavily observed environments on Earth because of the key role it plays in the global climate system [44]. However, the cryosphere also includes some of the most complex dynamical systems that have yet to be fully understood by modelers and researchers alike [44]. Major advancements in remote sensing technology have necessitated the need for simultaneous progression in the methodology used to process and analyze such data. This presents a unique opportunity to leverage modern computational capabilities to facilitate efficient high-resolution analysis of remotely sensed data over the cryosphere in order to develop a more comprehensive understanding of glacial dynamics which becomes increasingly important in the era of climate change.

One way to improve the scientific community's understanding of glacial dynamics is to study the features present on the ice surface as they are direct manifestations of the underlying ice dynamics [21, 45, 11]. By analyzing physical deformation patterns, and their historical evolution, there is significant potential to improve the field of glacier modeling [23, 45]. It has been shown that deformation characteristics such as crevasse patterns can be indicative of specific types of glacial acceleration which is one of the main sources of uncertainty in assessing global sea-level-rise [21, 45].

Any form of glacial acceleration will inevitably lead to ice deformation which often form crevasse fields composed of a variety of different crevasse types that may shed light on the underlying ice dynamics of the glacier itself. Therefore by studying physical properties of an ice sheet and how they evolve such that one begins to identify certain physical properties with specific types of glacial acceleration, the cryospheric community will be able to develop better models for understanding complex glacier dynamics.

## 1.1 Project Goals

Much of the previous work on glacier surface analysis has relied on satellite imagery, but active remote sensing techniques, such as light detection and ranging (LiDAR), have made recent advancements in the capability to provide complimentary and alternative methods for such analysis. LiDAR data have long been used for surface elevation determination and feature mapping [44]. For example, NASA's ICESat-2 mission provides previously unprecedented resolution and coverage for LiDAR based surface height measurements throughout the cryosphere [31]. The high-repetition rate of the multi-beam micro-pulse photon counting Advanced Topographic Laser Altimetry System (ATLAS) aboard ICESat-2 collects surface signal photons at a nominal 0.7m along-track resolution (under clear sky atmospheric conditions), and the 91-day repeat cycle allow for seasonal and annual assessment of important deformation features [31]. The goal for this project is to leverage these capabilities to analyze ice-surface deformation features on the large and small scale in an automated and efficient fashion using proven mathematical methodology.

For ICESat-2, methodology currently exists for extracting ice-surface heights at the original resolution of the data [22, 21], and part of the research presented here involves improving the efficiency and scalability of such methods without compromising the benefits of such unprecedented resolution. LiDAR data has been shown to improve object extraction accuracy from traditional satellite imagery-based methods [1], so this project aims to leverage these capabilities to automatically extract high-resolution ice-surface features This provides the community with a depth-oriented picture of glacier surface topography which can be leveraged to better understand what type of

deformation exists and to what extent the ice fractures over time in order to uncover information on the underlying ice dynamics. With enough coverage of such measurements one could begin to analyze deformation patterns ice-sheet-wide in order to aid large scale glacial modeling processes. Since different ice-surface features can indicate specific glacial dynamics [45], the potential for automatically extracting pre-defined ice-surface features from LiDAR data becomes valuable. The work presented here incorporates a physically informed machine learning-based classification approach for this task. It attempts to automatically distinguish three rather general ice-surface characteristics identified over the Greenland Ice Sheet, but over time, as domain experts continue to define distinct surface features classes from LiDAR data, this technique can be extended as a more robust ice-surface feature detector.

Ultimately the research to be presented aims to develop an automated and efficient method for visualizing and quantifying changes in ice-surface topography indicative of changes in the underlying ice dynamics. From these measurements, proven geostatistical methodology [16] will be applied to extract numerical quantifications of ice-surface deformation to create a physically informed neural network capable of identifying different ice-surface features. Thus a semi-automated method to leverage LiDAR data for classification of ice-surface deformation features will be presented which, in turn, will aid the development of data driven models of complex ice dynamics.

## 1.2 Challenges

With the vast amount of LiDAR-based Earth observation data and remote sensing methods currently available over the cryosphere, harnessing the value of such data becomes a difficult Big Data problem. Therefore, methods must be developed to analyze massive amounts of LiDAR data in an efficient manner. This often includes issues with scalability where visualizing large datasets and being able to extract value from them in an efficient manner proves difficult.

While the potential value of the proposed classification methodology is immense, there are several challenges in this specific problem domain which require careful attention. Firstly, no publicly available dataset currently exists for such a classification task, and a need certainly exists

to create such a dataset with thorough input from domain experts. This task will likely include numerous iterations and refinements as more data is observed and more important features are identified, and such a task necessitates significant domain knowledge. Ultimately the performance of such a classification model will rely heavily on the quality of the labeled training data.

Furthermore, being able to automatically filter low-quality data is necessary to legitimize such an approach. With the automatic collection of mass amounts of LiDAR data it is inevitable to have areas where the data is significantly obstructed by atmospheric conditions, and it would be extremely inefficient to manually sort through unusable data. Therefore, there is a need to automatically filter out cloud-obstructed data before such data is passed into computationally intensive signal extraction algorithms. Various instrument effects must also be accounted for, thus a data pre-filtering module also requires the separation of instrument effects from the raw data.

Finally, in order to make the overall approach feasibly efficient, such that it can keep up with data collection, significant work must be done to integrate parallelization and high performance compute power into the pipeline. With the data now hosted on various NASA-supported cloud compute platforms, there is an outstanding need to move the code for such analysis algorithms and models to the data, in order to eliminate the time and space overhead involved in data collection. With such platforms one can directly access ICESat-2 data, along with numerous other NASA-supported data products, from a high-performance computing environment. Therefore, an environment that best suits such an approach must be developed on top of the NASA compute cloud infrastructure. This will require substantial knowledge of Linux-based cloud computing methodology, and much trial and error.

# Chapter 2

# Background

The study of glaciers and ice sheets is particularly important during an era of global climate change because glaciers are highly sensitive indicators of changes in temperature and precipitation patterns [46]. Underlying glacier flow dynamics could then, to an extent, provide insight into changes in global climate patterns. One way to study underlying glacier dynamics is by analyzing surface topography. Glacier surface topography is roughly defined as the physical features and characteristics present on an ice sheet, such as crevasses, moraines, icefalls, glacial lakes, and more. The physical features present on a glacier surface provide insight into the past dynamics of the glacier, and may inform a greater understanding of how a glacier will respond to future climate changes. Therefore the investigation of current glacier surface topography can provide insight into the underlying glacier dynamics. Mapping the topography of a glacier's surface can also help to identify where meltwater is likely to accumulate and how it will flow downstream, which can inform decisions about water allocation and flood management.

Using the Greenland Ice Sheet (GrIS) as a primary source of investigation I plan to demonstrate how modern advances in satellite sensor technology and computing capabilities can be leveraged to improve current methods of ice-surface topography analysis. Greenland is a particularly important area of study because the GrIS is a significant contributor to global sea level rise, and is only expected to increase it's contribution in the coming years [33]. Greenland contributes to sea level rise primarily from the melting of its marginal outlet glaciers, which depend more on the local ice dynamics than ice-sheet-wide dynamics.

Surface topography data can be collected with a variety of remote sensing methods such as Light Detection and Ranging (LiDAR), satellite imagery, Radar, and more. For extremely reflective surfaces, such as ice sheets, an active remote sensing technique such as LiDAR proves to be extremely valuable. In this chapter I will introduce an algorithm for detecting ice surface height at a high resolution from LiDAR data. The specific datasets to be used are a result of NASA's Ice, Cloud, and Land Elevation Satellite-2 mission (ICESat-2). The Density Dimension Algorithm for ice surfaces (DDA-ice) is an algorithm that was originally developed by my advisor, Dr. Ute Herzfeld for the purpose of leveraging ICESat-2 photon point cloud data in performing high-resolution surface-height detection over ice surfaces. The motivation being that it is trivial to detect surface heights over smoothed ice, but difficult over areas of complex deformation which are typically the most important areas to track and assess [21].

## 2.1 Crevassing

Crevasses are deformation features on glaciers and ice sheets, and, while striking to they eye, they directly reflect the product of past ice-dynamic processes that the glacier has experienced [17]. Crevasses have also been shown to indicate tensional stress in the ice-surface layers providing important information on the local stress regime of the ice [11, 45]. Nye et al. present examples of various crevasse patterns that would appear under different configurations of lateral shear and longitudinal stress [37]. Typically a crevasse moves from where it first opens during it's lifecycle due to the natural flow of the glacier causing phenomena like shifts, rotations, or ablations in the ice sheet [11]. Therefore, the ability to track the development of individual crevasses or crevasse fields could has potential applications in the investigation of ice-sheet wide dynamics such as ice-velocity, surface-mass balance, and bedrock topography. The study of the relationship between crevasse patterns and ice flow regimes has long been studied. For example, Harper et al. show a distinct relationship between crevasse patterns and the glacier strain-rate tensor which is a mathematical representation of the deformation of a glacier over time [11]. Crevasses can either be topographically or dynamically induced. Topographically induced crevasses are typically the result of changing

bedrock topography causing enough tension on the ice surface to fracture it. Dynamically induced crevasses on the other hand are the result of some ice-dynamic process, such as glacial acceleration, creating tension on the ice sheet and fracturing the ice surface.

One important cryospheric process that produces noticeable crevasse patterns is a glacier surge. A surge is a relatively rare phenomenon in which the glacier experiences a sudden and drastic increase in ice-velocity and mass-transfer. During the surge, a glacier can accelerate to flow speeds on the order of 10-200 times the normal velocity transporting large amounts of ice and sediment downstream, and the surge phase can last on the order of 7-10 years as observed in the Svaldbard glaciers [45, 25, 38] . Glacier surges are of particular interest because they are the least understood form of glacial acceleration, yet can lead to significant contributions to global sea level rise [45, 21]. Surge-type glaciers are those which experience dynamic cycles of a long quiescent phase, where the glacier moves predictably, followed by a shorter surging phase where the glacier accelerates rapidly [9, 45]. While glacier surges typically happen quasi-cyclicly and are not directly coupled with climatic forces, there is evidence that specific climate and glacier mass-balance conditions can initiate them as well [6], thus the identification and tracking of surge-type glaciers becomes quite important in understanding Arctic glaciers responses to climate change [9, 38]. It has recently been shown that crevassing in particular can be used to better inform glacier surge models because crevasses are the most distinct manifestation of the rapid acceleration that accompanies a glacier surge [45]. The relevance to glacier modeling motivates the need for a method of identifying crevasse types associated with specific cryospheric processes, such as a glacier surge.

## 2.2 Light Detection and Ranging (LiDAR)

An active system for remote sensing is one in which the sensor emits energy pulses at a target, and measures the corresponding reflected energy in response. The advantage inherent to producing its own energy source allows active remote sensing systems to operate independently of lighting or some atmospheric conditions, such as clouds. On the other hand, a passive system is one in which the sensor simply detects and measures the naturally occurring energy of a target, such as solar

radiation or thermal emissions. The advantage is that passive systems are less complex and more cost efficient, while the disadvantage is that they are more dependent on the quality of the emitted natural energy which is heavily influenced by lighting and atmospheric conditions. LiDAR is an active system for remote sensing [44].

In a LiDAR system, the sensor transmits pulses of laser light aimed at a target, and measures the return of reflected laser energy [44]. When the laser pulse hits the surface, or any objects in its path, some of the light is reflected back to the sensor, in the form of photons, and the sensor measures distance by computing the time it takes for the transmitted energy to return to the sensor. Nowadays, most airborne LiDAR systems can record numerous returns per pulse, which allows for distinguishing between multiple surface-like features such as vegetation or canopy cover and clouds [44]. The primary data product output by LiDAR observations is a geolocated point cloud [44], which renders a coordinate-like representation of the surface and any objects above the surface.

For cryospheric applications, LiDAR proves extremely effective because ice surfaces are highly reflective by nature. Also, while the quality of satellite imagery may suffer in low-light scenarios, an active remote sensing method such as LiDAR produces extremely high resolution returns in low-light situations because the signal to noise ratio is much higher without sunlight. Given that much of the cryosphere is located in areas of the world that receive little to no sunlight during certain seasons, one can imagine the advantage of using LiDAR to study it. Another unique feature of using LiDAR data to study the cryosphere is that because it actively transmits a laser pulse along track it is able to follow the contours in the glacier in such a way that produces a quantifiable depth dimension which can be more difficult to interpret from imagery alone.

## 2.3 Ice, Cloud, and Land Elevation Satellite-2

Given the cases where LiDAR data can be advantageous or complementary to satellite imagery for analyzing ice sheet properties, NASA's ICESat-2 mission serves as a valuable tool. ICESat-2 was launched to leverage modern laser altimeter technology in providing cryospheric measurements for investigating elevation and mass change over glaciers and ice sheets. One of the primary

mission objectives is to "provide height information that will allow to understand changes in the cryosphere that occur in the present realm of a changing climate" [21]. The mission evolved from a previous satellite with similar goals, ICESat, with the motivation to incorporate a higher resolution sensor and provide more dense spatial coverage of the Earth at more frequent sampling intervals [31]. ICESat was launched in January 2003, and was the first satellite carrying a LiDAR system for cryospheric applications [44]. ICESat-2 deployed a more advanced LiDAR system called the Advanced Topographic Laser Altimetry System (ATLAS), which has provided a more powerful and higher resolution sensor. ATLAS is a micropulse, high repetition rate, photon counting laser altimeter which records photon returns at a nominal resolution of 0.7m along track under clear-sky atmospheric conditions, allowing for time-tagging and geolocating of individual photons [31, 29].

The ATLAS sensor acquires observations from 6 total beams, 3 pairs of strong and weak with each pair separated by 3.3km and a pairwise-width of 90m [31, 29]. This high-resolution sensor allows ICESat-2 to record previously unprecedented dense coverage of surface returns across the Earth, with particular focus on the polar regions. The dense per-cycle coverage of Greenland is represented well in Figure 2.1, where all 6 channels of the ICESat-2 tracks are plotted (in red) for a 3 month period over Greenland which includes one full cycle of the satellite plus 2 days worth of repeat data. This figure demonstrates why ICESat-2 is an ideal mission to leverage for studying glaciers because it provides uniquely dense coverage for a LiDAR based sensor, and with it's 91-day repeat cycle it can provide quarterly updates to the data at hand. Also, one of the main objectives of the ICESat-2 mission is observing the Greenland and Antarctic ice sheets in order to quantify their contributions to global sea-level rise [31]. In conjunction with dense coverage of the polar regions, numerous campaigns have been carried out to compare airborne LiDAR data with ICESat-2 along-track photon data in order to validate the geolocation accuracy [30, 19, 3, 48].

With this level of coverage and accuracy, ICESat-2 provides opportunities for LiDAR-based glacier analysis methods that were previously only possible by launching individually funded spacecraft missions [24, 28]. Because of ICESat-2, it is now freely accessible to extract precise ice-surface height measurements in the presence of complex ice morphology. It has already been shown that

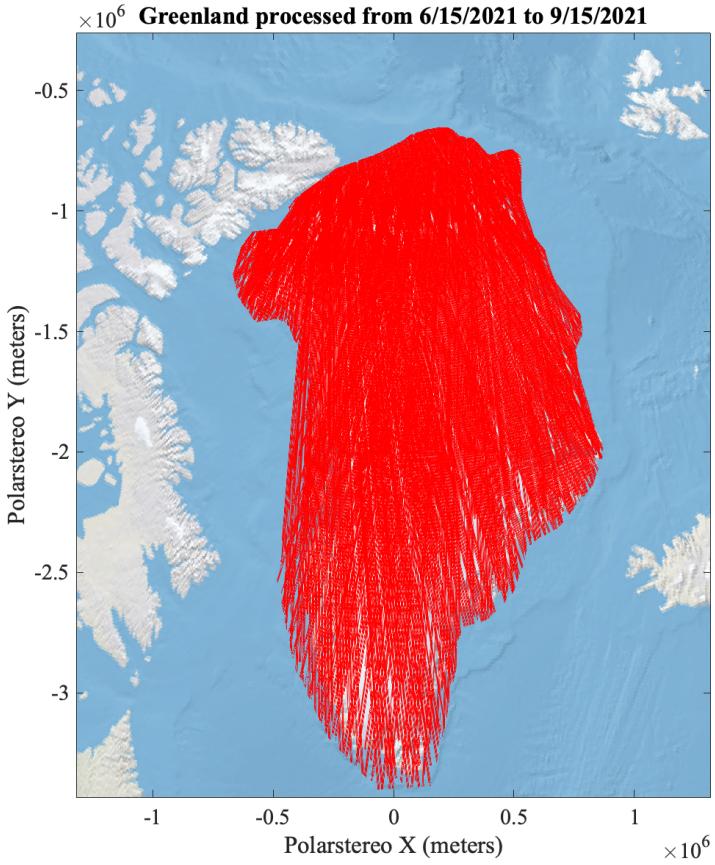
we can use the data to distinguish and characterize complex ice features, such as different glacier acceleration types, that will advance our understanding of the underlying ice dynamics [21]. The ICESat-2 Global Geolocated Photon Level 2A data product, identified as ATL03, provides the geolocated photon data detected by the ATLAS instrument, and is used in the other surface height products such as sea ice (ATL07), land ice (ATL06) and vegetation canopy height (ATL08) [36]. The ATL03 data product is primarily used in this research.

## 2.4 The Density-Dimension Algorithm for Ice Surfaces (DDA-ice)

The DDA-ice was developed to leverage the full measurement capabilities of the ATLAS sensor aboard the ICESat-2 satellite, and extract accurate, high-resolution surface-height measurements over morphologically complex ice in various atmospheric conditions [21, 22]. It has been shown that the DDA-ice produces surface height returns at the resolution of the of the ICESat-2 geolocated photon data (0.7m along-track), and can automatically adapt to changing environmental conditions [21]. High-resolution surface height returns are especially important when analyzing glacial features such as crevasses whose size and shape characteristics can be indicative of different ice deformation processes. Due to the active nature of LiDAR and the capabilities of the ATLAS sensor, it becomes possible to extract crevasse depth measurements at a 70cm along-track resolution to produce detailed renderings of complex crevasse patterns that would be impossible to determine from imagery. The DDA-ice was developed to accomplish this high-resolution extraction over morphologically complex ice surfaces [21, 22].

The algorithm is parameter-driven and auto-adaptive by nature. It builds on the ICESat-2 geolocated photons product (ATL03) to automatically separate surface-signal photons from "noise" photons, and use these signal photons to produce a high resolution surface height interpolation. It starts by using large scale histograms to identify 50m high signal slabs every 10m along-track which contain the highest concentration of photons. It then colocates a noise slab directly above the signal slab to develop a photon concentration contrast between surface signal photons and noise photons. The main idea behind the DDA-ice is that surface signal photons will have a

Figure 2.1: ICESat-2 satellite's ground track coverage of the Greenland ice sheet for 3 months of orbit data. Each individual red line represents a single beam from an ICESat-2 Reference Ground Track (RGT), over Greenland during the time period 6/15/2021 to 9/15/2021. A time period of 3 months is used for illustrative purposes because it roughly aligns with the 91-day ICESat-2 repeat orbit cycle.



higher photon concentration among their neighbors compared to background or noise photons. The algorithm employs the Gaussian radial basis function (rbf) for computing photon density with an anisotropic kernel which favors the horizontal direction to better encapsulate ice surface returns along track [21, 22].

Density is computed by centering the rbf on each photon, $c$, in the noise and signal slab:

$$\Phi(x, c) = \Phi(||x - c||) \tag{2.1}$$

Letting $r = x - c$ we use a Gaussian rbf:

$$\Phi(r) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(\frac{r}{\sqrt{2}\sigma})^2} \tag{2.2}$$

For a pre-defined standard deviation, $\sigma$. An anisotropy norm $||\cdot||_a$ is used to weight horizontally separated points higher than vertical ones:

$$||\cdot||_a = ||A \cdot||_2 \tag{2.3}$$

$$A = \begin{pmatrix} \frac{1}{a} & 0 & 0 \\ 0 & \frac{1}{a} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.4}$$

Then for each photon, $c$ we calculate a density field as follows:

$$f_{d,a}(c) = \sum_{x \in \mathcal{D}_c} \Phi(||x - c||_a) \tag{2.5}$$

For all neighboring photons $x$ within search region $\mathcal{D}_c$

An auto-adaptive, quantile-based threshold function is then applied to further separate noise and signal photons, using density as an additional dimension [21, 22]. The concept here is to use the statistical properties of the photons in the noise slab compared to those in the signal slab to further the signal-noise separation because the signal slab will contain background noise photons as well as signal photons, while the noise slab will only contain noise [21, 22].

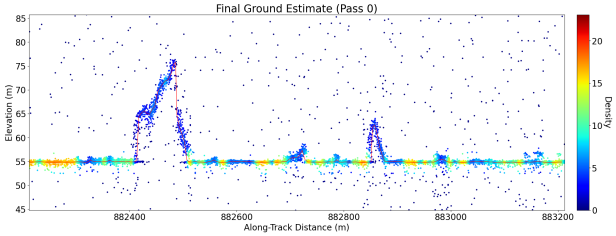
At this point of the algorithm the signal photons from the ice-surface return are isolated from any background noise photons, and a piece-wise linear function, weighted by density, is used to interpolate a ground follower [21, 22]. The ground follower is roughness-adaptive, which means over smooth surfaces the piece-wise linear function uses a fixed bin-size, $R = 5m$. The roughness criterion is based on the standard deviation, $s$, of the elevation of the signal photons within this fixed bin, and if $s > s_{rough}$, for some adaptive threshold parameter $s_{rough}$, then the surface is deemed to be crevassed and the ground follower resolution factor, $r = 2$, is employed and the piece-wise linear function uses a bin size of $R = 5/r = 2.5m$ [21, 22]. Much work has been put in to calibrate the algorithm such that it can adapt to different atmospheric and illumination conditions without compromising the integrity of the results. These flexible capabilities provide a unified method for studying specific glacier surface phenomena and overall mass change at the local and ice-sheet-wide scales.

With the density in coverage provided by ICESat-2 over the polar regions [31, 29], the DDA-ice is an ideal tool for extracting important, small-scale ice sheet features on a large scale. Figure 2.2 shows a variety of features that the DDA-ice is able to automatically detect at a high resolution. These features demonstrate how the algorithm can effectively process surface topographic data across a dynamically changing ice surface without compromising the integrity of the results. The implications of such an auto-adaptive algorithm lie in the potential for hyper efficient micro and macro scale analyses. For example, a Greenland-wide DDA-ice output can be used to detect large scale topography changes and overall glacier structure, and the same data can be utilized for investigating individual outlet glaciers, crevasse fields, or other regional features (See Section 4.3). It should also be noted that the DDA-ice is generally valid for all micro-pulse photon-counting laser altimeter data and can be easily ported to process data from different LiDAR sensors [21].

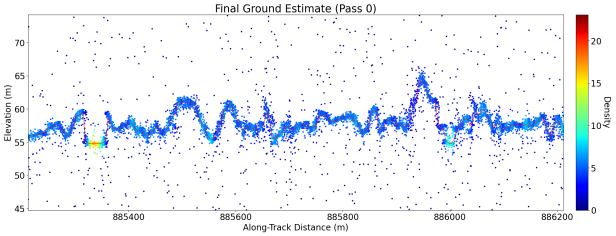
## 2.5 Variograms

The variogram is a classic geostatistical tool for quantifying spatial variability in geolocated data [10]. It has various applications in geostatistics such as spatial interpolation (i.e. Kriging),

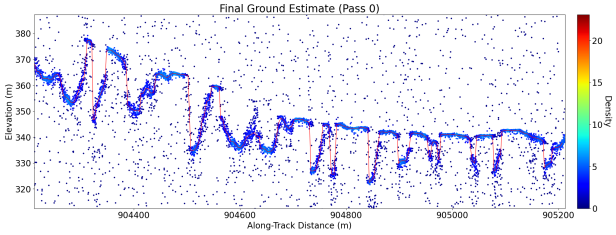
Figure 2.2: 1km segments of interpolated surface height estimates (including signal photons) from the DDA-ice over Greenland. ICESat-2 ATL03 granule: ATL03_20220401061928_01401503_005_01.h5, all figures from the same strong beam: gt1l. **(a)** is a segment from icebergs detected over smooth ice or sea-ice **(b)** is a crevasse field which has undergone numerous phases of deformation over a long period of time **(c)** is a more structured crevasse field representative of more recently opened crevasses **(d)** is a segment of undisturbed snow on the ice sheet.



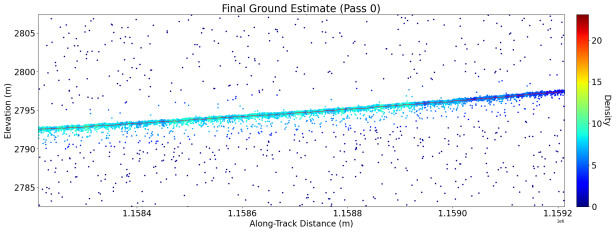

(a) Icebergs in a Fjord




(b) Late Deformation Phase Crevasses




(c) Early Deformation Phase Crevasses




(d) Undisturbed Snow

natural resource distribution, and climate modeling and weather forecasting. The purpose of the variogram is to quantify spatial variability or model spatial correlation by aggregating the covariance between pairs of points that fall within a pre-determined distance of each other, called lag distance, as defined in Equation 2.6. Over an increasing range of lag values, the variogram provides insight into the spatial correlation of observed data relative to distance. It takes a segment of geolocated data and sorts the points into lag bins, defined as all pairs of points separated by the lag distance or less. For example, a 4x4m grid of data with a lag value of 2m would have 2 lag bins, one for points separated by 2m or less and one for points separated by 4m or less. It then computes the variance of some target variable between all pairs of points within each lag bin, and normalizes relative to the bin size producing a function that illustrates the spatial structure of the data and how the data values tend to vary as the distance between geolocated points increases.

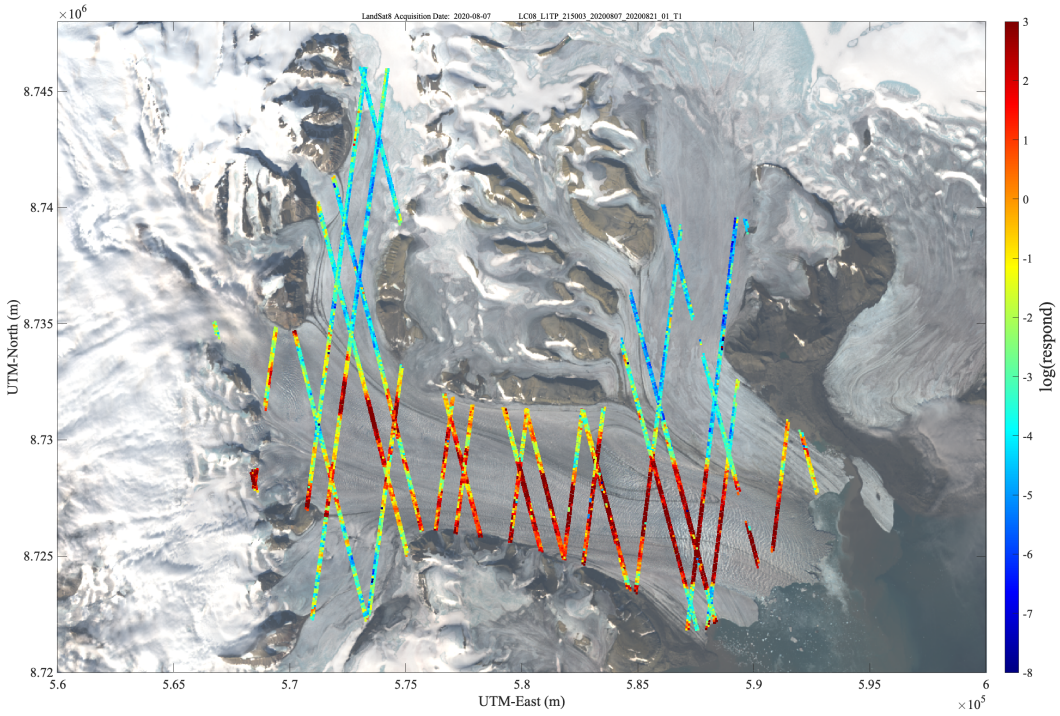
The estimation of a variogram is permitted if a sample dataset satisfies the assumption of homogeneity [15]. In the context of glaciers, this would imply if one is to estimate the variogram for a closed segment of ice-surface data, that segment of ice should have the same surface structure throughout, even if the structure is complex [16]. This assumption allows for valid geostatistical parameterizations of the ice-surface morphology. The contrary case would be heterogeneity, where a closed segment of ice-surface data reflects more than one different signature of surface structure. Because the variogram is a measure of spatial variability, the values increase as a data sample becomes more spatially dissimilar [10]. In the context of ice sheets, by computing variograms over surface topography data one can extract statistical measures which capture the spatial structure of the ice sheet at an intermediate scale [15]. Where the values are low, we can assume the surface topography is relatively smooth, whereas higher variogram values will indicate the presence of deformation in the ice surface. Moreover, it has been shown that the variogram can operate as somewhat of an information filter that retains spatial characteristics which capture properties of crevasses such as depth, spacing, and general shape [20, 16]. Ultimately, the variogram proves to be a valuable tool for extracting statistical information from geolocated photon point cloud data in order to identify relevant deformation processes on an ice sheet.

Moreover, the residual variogram, defined in Equation (2.3), is used to complement the original variogram function by compensating for local drift or trend [15]. This fact is extremely useful for this area of study because the residual measure is quantifying the extent of surface deformation irrelevant of the overall slope. Removing trend, or slope in this case, is less important over smooth or relatively flat surfaces, but over areas of heavy deformation there could be an anomalous trend which is not indicative of the actual morphology, especially if the deformation is over a sloped area. Furthermore, from a variogram computed over a segment of ice surface, the maximal value can be interpreted as a measure of surface roughness [16] which quantifies surface deformation and provides direct insight on overall glacier surface morphology. Surface roughness can be defined as the spatial derivative of surface elevation, which is useful for quantifying morphological properties of the ice which cannot be determined from surface height alone [16]. Now, surface roughness derived from a non-residual variogram can lead to smooth but heavily sloped areas of the ice surface registering as high roughness. Zooming in on Figure 2.3, on can see where the higher residual variogram values (red-orange) correspond to crevassed or otherwise deformed areas of the Negribreen glacier system, especially near the calving front of the glacier where the crevassing is the most significant.

In this project, the goal is to use the residual variogram to quantify the spatial variability in surface topography across ice sheets. I will use surface height as the target variable, with the observational data resulting from 1) the DDA-ice interpolated surface height estimate and 2) the geolocated signal photons from which the surface estimate is defined. Computing the residual variogram individually for each dataset, and then combining, provides robust quantifications of the spatial structure of the ice surface. Classically, the semivariogram is defined as half the variance of the difference between 2 points (in this case difference in elevation), and going forward I will use the term variogram to mean experimental semivariogram which is mathematically defined as, for some lag value $h$:

$$\gamma(h) = \frac{1}{2}\text{Var}\left(Z(x) - Z(x+h)\right) \tag{2.6}$$

Figure 2.3: DDA-ice surface roughness plotted over a LandSat8 image of the Negribreen Glacier System (NGS) collected on 8/7/2020. All strong beams for all ICESat-2 tracks which intersect NGS during the year 2020 are represented (removing tracks that were unusable due to cloud coverage). The coloring is representative of the degree of surface roughness, quantified as the log of the maximum residual variogram value for a 150m along-track segment with 5m lag.



More specifically, for observed data we derive the following equation:

$$\gamma(h) = \frac{1}{2n} \sum_{i=1}^{n} [Z(x_i) - Z(x_i + h)]^2 \tag{2.7}$$

Where $Z$ is a measure of elevation, so $Z(x_i)$ is the elevation of point $x_i$ in meters, and $n$ is the total pairs of points separated by $\leq h$ meters. From here, we remove the trend to get the equation for the residual variogram [15]:

$$r(h) = \gamma(h) - \frac{1}{2} \left( \frac{1}{n} \sum_{i=1}^{n} [Z(x_i) - Z(x_i + h)] \right)^2 \tag{2.8}$$

The challenge in producing structurally representative variograms lies in determining an ideal parameter set for the domain area. One must always determine a predetermined geographical window size of which to compute a variogram, hereto referenced as *window_size*, and an appropriate lag value, $h$, for which to partition the geolocated data. For this project, I will also introduce a variable, *window_step*, for how far (in meters) to offset the along track window when computing successive variograms. If *window_size* and *window_step* were the same then there would be no overlap in data between consecutive along track variograms, but given the dynamic nature of ice surfaces, it is important to include a certain amount of overlap because the important features are not constrained uniformly across the ice sheet. Given 2D elevation data over an ice sheet, *window_size* and *window_step* will be measured in distance along track (meters), and the lag value, $h$, will also be in meters, but has the extra constraint that *window_size* must be divisible by $h$ in order to include all the data points within the current window.

# Chapter 3

# DDA Code Development

This chapter is intended to highlight some of the improvements I have made to the DDA-ice codebase over the last two years. The motivating force behind these changes was to increase the computational efficiency of the algorithm such that one could feasibly leverage it's high-resolution capabilities in a Big Data analysis pipeline. As satellite data becomes increasingly available it is of greater importance for the Earth science community to have access to Big Data analysis methods and tools which are not only highly accurate and insightful, but efficient as well.

## 3.1 Performance Improvements

The DDA-ice algorithm is developed in Python, and leverages various data-analysis libraries such as NumPy, GeoPandas, SciPy, and more. Python is inherently slower and more memory intensive than lower-level programming languages such as C or C++ [2]. Python is also a garbage collected language which causes delays in memory reclamation leading to an even higher memory overhead [13]. More specifically, Python uses reference counting for garbage collection, which means every time an object is initialized Python stores the object and its reference count (1 at initialization) in memory. Anytime another piece of code references that object, the reference count is incremented by 1. If the variable which references the object is reassigned or deleted, the reference count is decremented by 1. If an object has a reference count of 0, the Python garbage collector will free the memory associated with that object such that this memory space can be reused later in the program. A trivial example of reference counting in Python is shown below:

```
a = 1 (reference count = 1)
- the stored object is 1, the reference to that object is a
b = a (reference count = 2)
a = None (reference count = 1)
del(b) (reference count = 0)
```

In this instance, garbage collection works seamlessly, however the automatic, reference counting garbage collector in Python can not handle cyclical references (aka reference cycles). This means if an object refers to itself by variable assignment, appending itself to its own list, or other circular reference, then the reference count is incremented by 1, which means deleting this object will not decrement the reference count to 0. A trivial example of this is shown below:

```
a = [1] (reference count = 1)
a.append(a) (reference count = 2)
del(a) (reference count = 1)
```

Here the memory for the object referenced by **a** is not actually freed, but the Python program no longer has access to the object. These situations can cause memory leaks when a programs' memory becomes overloaded with unusable memory space due to the garbage collector not deallocating the memory associated with deleted objects, eventually causing a program to crash.

When I took over the DDA-ice code development at the beginning of this project, I encountered significant issues with memory leaks when processing large datasets with high photon concentrations. Upon investigation, I realized that at each step of the algorithm the signal and noise photon data were copied, with the copy being passed into a function to perform computations, then the result was returned and used to update the original data, and the copies were deleted. However, each of the algorithm modules involved some form of cyclical reference to the data, which means when a copy was deleted after a function return that memory was never actually deallocated. Below is a pseudo-code representation of what was happening:

```
data_copy = data.copy()
subset = data_copy[x:y] (example of cyclical reference)
... do something with subset ...
del(data_copy)
```

Here, when the copy of the data is deleted, it is impossible for the reference count to be 0 because of the second line. This approach becomes problematic in a garbage collected language because memory is not always freed at the moment that data objects are deleted, and when dealing with large datasets this effect can snowball quickly.

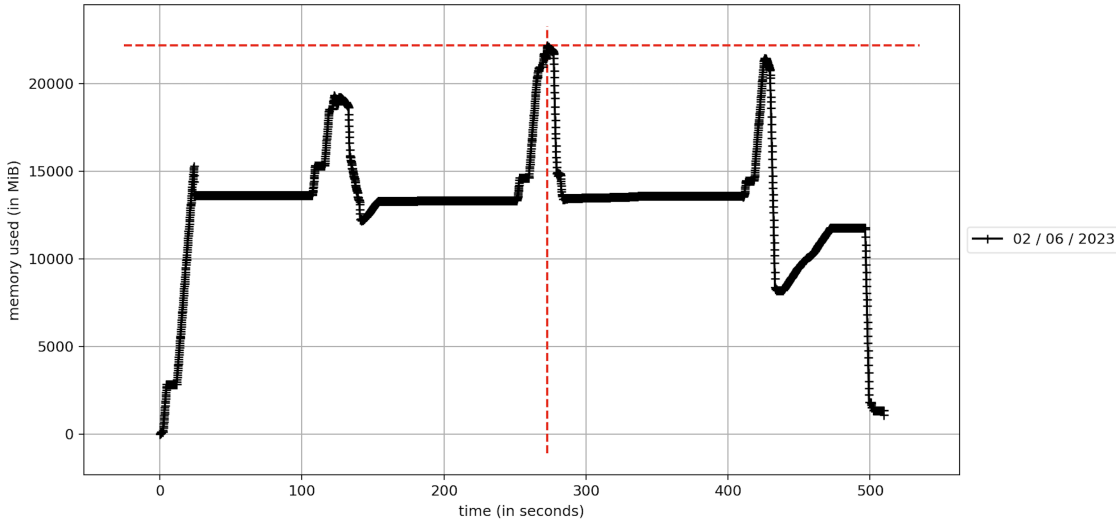
The memory leak issues can be visualized in Figure 3.1a, where an input dataset of less than 30MB in size caused the algorithm to eat up more than 20GB of memory during execution before crashing around 500 seconds in. In order to directly address these memory management issues, I refactored the code from a function-only codebase into a class-based architecture where a single class object contained the data as attributes and algorithm modules as methods. I then integrated binary masks as attributes to separate the signal and noise photons at each successive algorithm step, rather than making copies and subsets of the original data. With this approach the data is first loaded into the DDA-ice class object, and each algorithm step modifies the binary masks for signal and noise photons based on the statistical properties inherent to the algorithm. This approach introduced one true source of the data available to all algorithm components, and the binary masks provided an extremely lightweight method of data transformation necessary for the different computational algorithm modules. Instead of saving numerous derived subsets of the original dataset in memory, the binary masks are saved to serve the same purpose, but take up far less memory space.

Along with the structural refactor, I integrated parallel computing into the density computation module using Python's multiprocessing package. The density module is the most computationally intensive module in the algorithm, consuming 90-95% of overall algorithm execution time, therefore parallelizing only this module would significantly improve the overall algorithm efficiency.

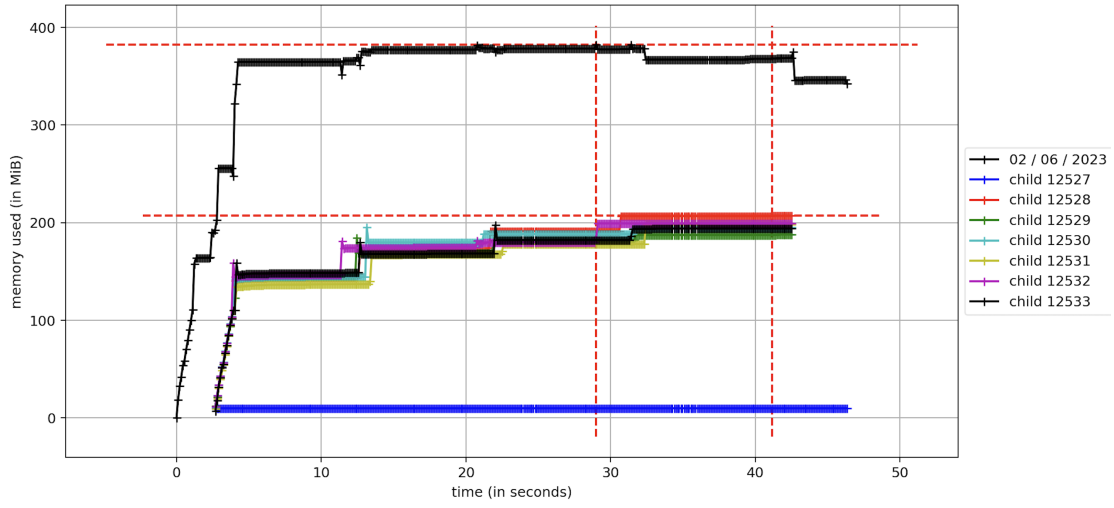
However, since the other modules efficiently execute serially on a single thread, parallel computing only increased the overhead and worsened overall algorithm execution time. The implementation of parallel computing into the density module involved spawning a pool of worker processes equivalent to the host machine's available machine cores, and distributing each photon density center (see Equation 2.1) along with the neighboring photons according to the Gaussion rbf (see Equation 2.2) to their own thread iteratively until density is computed for each photon in the dataset. By spawning a pool of workers equivalent to the available machine cores, the computationally expensive density module can uniformly distribute the work amongst more machine cores which drastically reduces time and memory overhead. Python's multiprocessing package abstracts away many of the low-level details of process management, and handles the creation of individual processes separate from the main thread, by forking the main process into direct copies of itself, and assigning each process its own memory space and compute resources (such as an individual compute core on the host machine).

The results of this comprehensive refactor are illustrated in Figure 3.1b where the algorithm now maxes out at less than 0.4GB of memory and finishes running in less than 50 seconds where previously it was occupying over 20GB of memory and crashing after 500 seconds. The plot also illustrates the effective use of parallel processes.

Figure 3.1: Results from profiling the DDA-ice code with Python's memory-profiler package. Each figure is representative of running the same dataset of approximately 30MB in size, but at different steps of efficiency improvement implementation. **(a)** is before any refactoring had been done, and there was a significant memory leak issue with this specific dataset **(b)** is a result of the fully refactored code with a robust implementation of parallel computing. ICESat-2 granule info: ATL03_20200628151943_00500804_005_01.h5, sea-ice dataset near Antarctica.




(a) Old Code




(b) Refactored Code

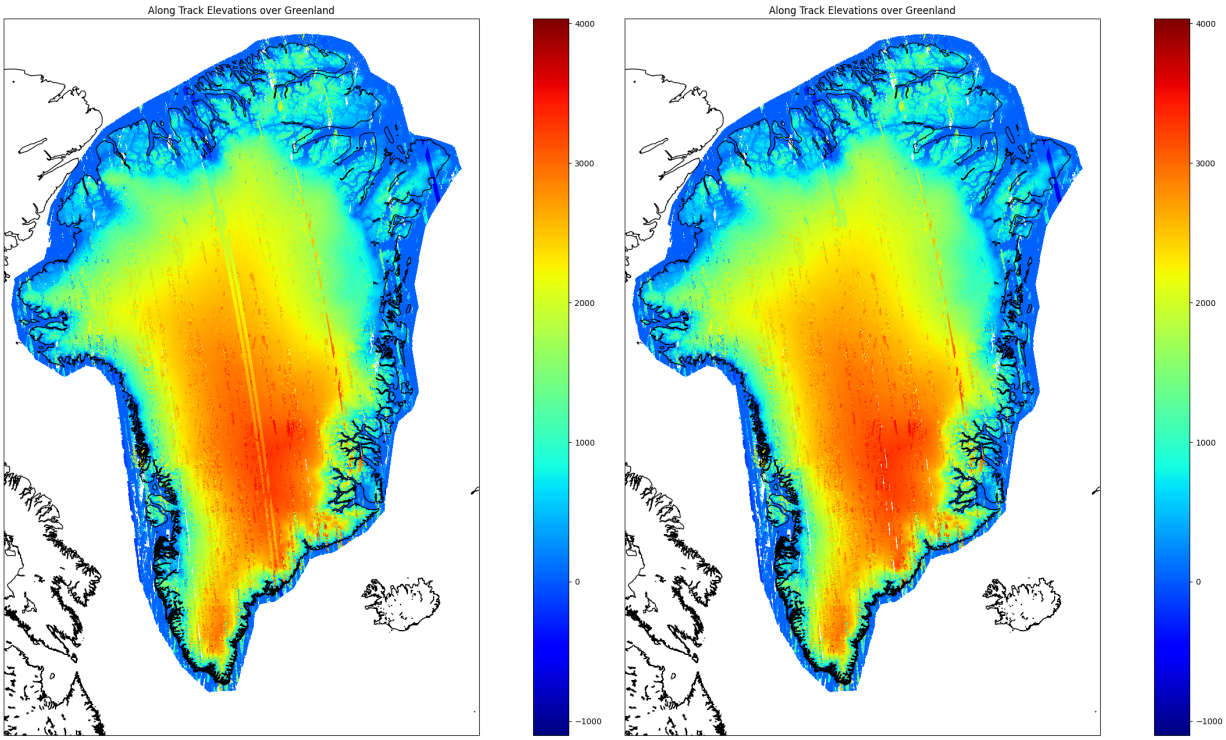
## 3.2 Automated Cloud Filtering

After testing the previous code performance improvements on a 3-month interval of data over Greenland, I noticed inconsistencies in the surface elevation results throughout the ice sheet, as can be demonstrated by Figure 3.2. Visualizing the results on such a large scale made it evident that there were many places in the data where the surface returns from the sensor were being obstructed by clouds without the algorithm catching it, causing what appears to be sporadic jumps in elevation that were not representative of the surface topography. When the ATLAS sensor hits strong cloud cover or other high background atmospheric conditions it sometimes causes the attenuated band to return a photon point cloud which does not even include the surface returns. In other cases, the attenuated band returned from ATLAS includes the surface return and a strong low-lying cloud, reflecting a sparse ground signal return, if any at all. These cases will introduce significant inconsistencies into any surface extraction algorithm, and for the DDA-ice these inconsistencies can be visualized in the signal slab detection module (see Appendix A Figure A.1).

Upon further investigation I also realized there were various instrument effects that the algorithm was not accounting for such as track off-pointing and data quality filters. The first step in filtering out erroneous returns was then to develop a DDA-ice data pre-filtering module which incorporates various sensor-defined data quality flags. This approach eliminated some of the larger inconsistencies where either the sensor was mis-calibrated or was temporarily pointing at a different location, which can be visualized in Figure 3.2b where 2 significant off-pointing tracks were removed that previously crossed the center of the ice sheet. However, the approach failed to solve the problem of surface height returns seemingly jumping into the clouds.

In order to extend the approach to filter out cloud-obstructed data mathematically I chose to leverage the principles of the DDA-ice in the initial signal-noise slab separation module. By doing so, I could identify areas in the raw photon data where the surface signal was inconsistent or completely erroneous before advancing into the more computationally expensive parts of the algorithm. The challenge became removing enough data to eliminate all areas where the sensor

Figure 3.2: DDA-ice surface heights projected over Greenland from data collected over the 3-month time period: 6/15/2021 - 9/15/2021. Both images reflect the status of the algorithm before implementation of automated cloud filtering. **(a)** reflects the algorithm before any data pre-filtering modules were implemented, while **(b)** demonstrates the effect of including the sensor-specific data quality filters for data collected during this time period.




(a) No Data Pre-filtering

(b) Pre-filtering Sensor-specific Quality Flags

could not accurately distinguish between ground and cloud, but retaining high-resolution surface signal even if sporadic throughout intermittent cloud cover. My approach became twofold: 1) use the onboard Digital Elevation Model (DEM), where available, to ensure that each along-track signal slab was centered within a valid range of the DEM, and 2) employ a minimum photon count threshold for signal slabs. The reasoning for including a second step has to do with the fact that the DEM is not always available and it can also be obstructed by heavy cloud cover. Since photon returns are more sparse from clouds compared to ground because of the level of reflectance, a minimum photon count threshold for the signal slabs will aid in avoiding areas where the signal slab contains atmospheric rather than surface returns.

To implement such an approach for each along-track signal slab I integrated the two criteria in Equations 3.1 and 3.2, such that if each of these returned false, this 10m along-track chunk of data would be masked out (or removed from consideration for successive algorithm modules). Once the criteria were established, I began an iterative process for determining ideal, data-driven parameter values for $\epsilon$ and $\alpha$. I sourced test data from multiple granules over the GrIS that represented the following cases: 1) cloud-obstructed data with a faint ground return, 2) completely noise-saturated photon point-cloud and 3) strong, upper-atmosphere cloud cover where the sensor returns an attenuated band which includes the cloud but not the surface. By testing a wide range of values over both cloudy and non-cloudy data and over a variety of glaciers in the polar regions I was able to formulate robust parameter values that could be applied throughout the cryosphere to produce accurate results. I started with $\epsilon = 100$ and $\alpha = 50$ and made incremental tweaks as I observed the results, and ended up settling on $\epsilon = 200$ and $\alpha = 35$ The results of introducing these criteria can be visualized on the macro-scale in Figure 3.3, which when compared to Figure 3.2, one can see how removing cloud-obstructed data produces a consistent topographic mapping across the GrIS. Figure 3.4 illustrates the before and after effects of the automated cloud-filtering module on a smaller scale, with individual 1km segments of along-track ground interpolation data over Greenland.

The following equations reflect the automated cloud-filter criteria I implemented into the

DDA-ice signal slab detection module:

Let $Y_{slab\_center}$ represent the height of the center of the DDA-determined signal slab, in meters, and $Y_{DEM}$ be the onboard DEM height from the point closest to the center of the signal slab (horizontally), in meters, and $\epsilon = 200$ be the heuristically determined tolerance distance.

$$|Y_{slab\_center} - Y_{DEM}| > \epsilon \tag{3.1}$$

Let $S$ be the set of all photons in the current signal slab, and $\alpha = 35$ be the heuristically determined minimum photon count threshold.

$$|S| < \alpha \tag{3.2}$$

Clouds and other atmospheric conditions will always introduce inconsistencies into remotely sensed surface data to some extent which can create significant inefficiencies in data analysis when forced to manually sift through unusable data. The approach outlined in this section lends itself to reducing such inefficiencies by automatically extracting only the areas in the photon point cloud which produce high resolution surface data to be used for effective topographic analysis. The method of data pre-filtering limits the shear number of photons under consideration in the density and ground interpolation modules, improving algorithm execution efficiency, and in return removes the burden of ad hoc data quality inspection from the scientist. The added benefit of using a parameter driven auto-adaptive algorithm, such as the DDA, in this manner is that one can simply adjust parameter values to increase or decrease the robustness of data filtering. For example, one may favor keeping some inconsistencies to ensure maximal representation when investigating a smaller area where the burden of ad hoc analysis is light. However a larger area of investigation may require slightly over filtering because it can be difficult to distinguish errors from features when looking at data on a scale represented in Figure 3.2. Therefore an automated method of pre-filtering out inconsistencies based on the mathematical logic of an already proven algorithm (DDA-ice) leads to a much more efficient manner of small-scale and large-scale glacial analysis. The rest of the DDA-ice specific parameters used for these runs can be found in Table 4.2

Figure 3.3: DDA-ice surface heights projected over Greenland from data collected over the 3-month time period: 6/15/2021 - 9/15/2021. This image reflects the status of the algorithm after implementation of the automated cloud-filtering module.

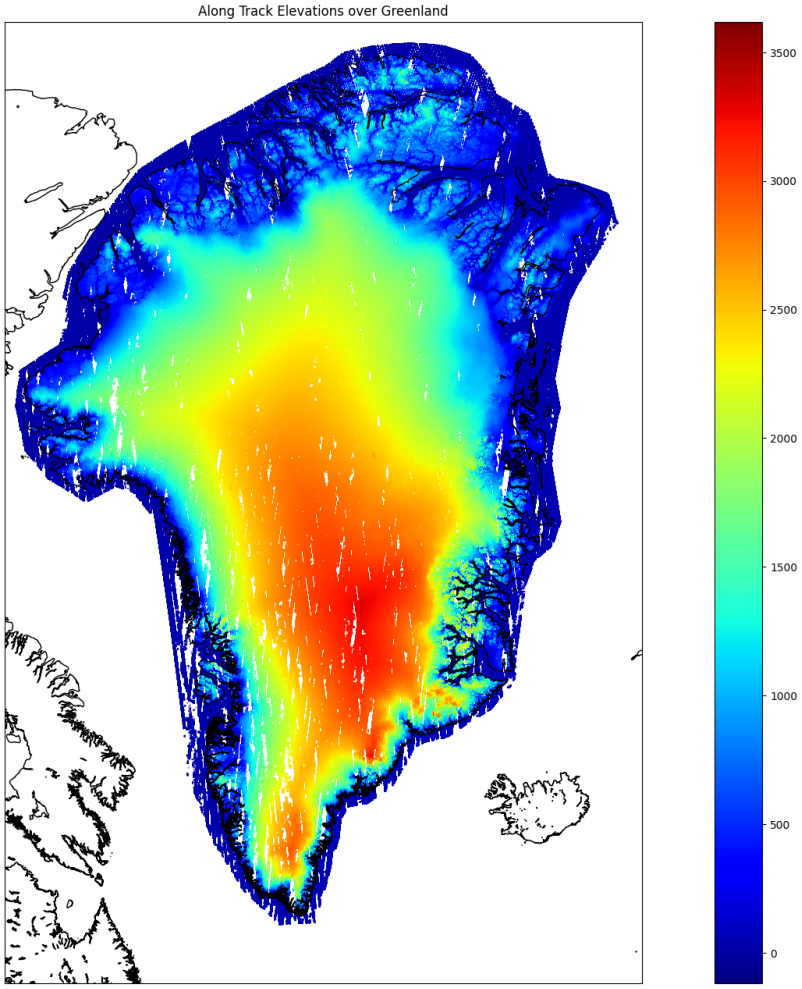
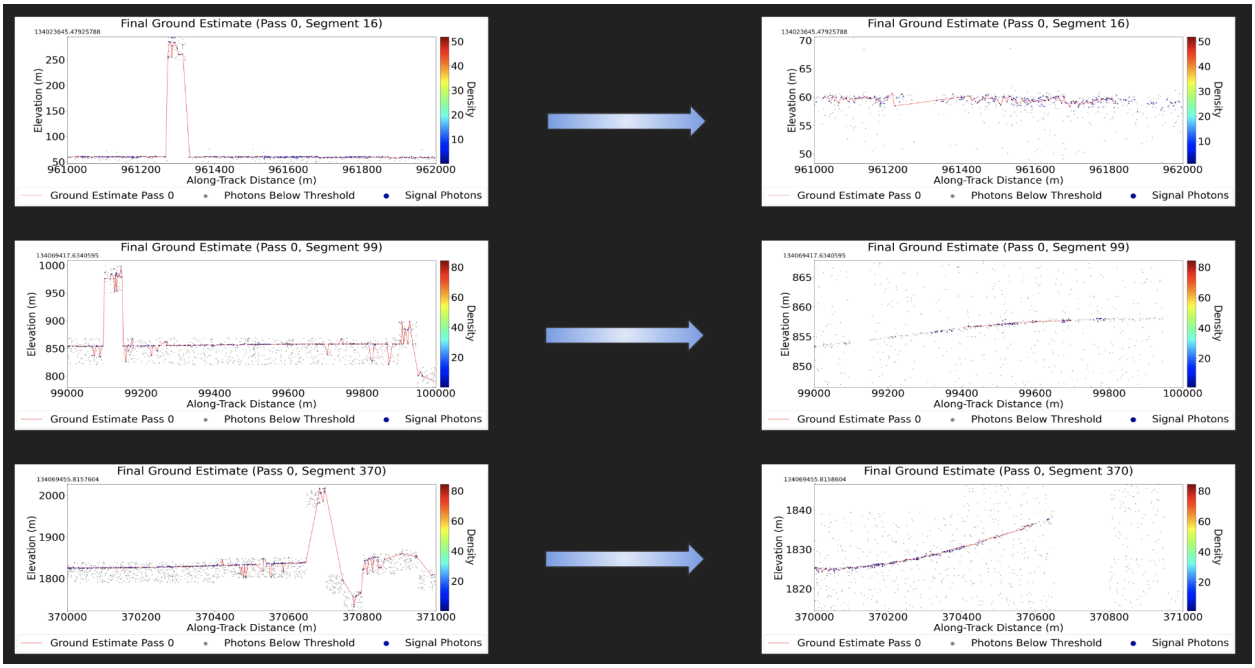


Figure 3.4: 1km DDA-ice interpolated surface height segments over the GrIS. ICESat-2 ATL03 granule: ATL03_20220401044511_01391503_005_01.h5, channel: gt1l (strong). The segments in the left column were produced before the automated cloud-filtering module was implemented, and the same segments on the right reflect the effect that the cloud-filtering module had on these particular segments.



# Chapter 4

## High Performance Cloud Computing

Cloud computing has revolutionized the way many businesses and individuals operate by delivering on-demand compute services such as computing power, storage, databases, software and more over the internet. Now, instead of only relying on locally available resources, one has on-demand access, via the internet, to remote servers hosted in data centers across the world delivering previously unimaginable levels of computational flexibility. These advancements have provided individuals and organizations with convenient, scalable, and cost-effective access to computing resources.

Cloud computing services take many different forms, with two of the major categories being Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). Both formats are ways to provide on-demand access to computing resources over the internet. IaaS provides access to virtualized computing resources which the user can configure and manage to support their desired environment. This includes creating virtual machines (VMs), installing operating systems and software specifications, and configuring networking and storage rules. In a PaaS setup the user has access to a platform pre-built upon existing infrastructure for which a user can develop applications, scale, and integrate database services. The difference is that IaaS allows the user more control of the infrastructure setup, while PaaS provides a more streamlined workflow by abstracting the underlying infrastructure to varying extents.

Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Center for Climate Simulation (NCCS) at Goddard Space Flight

Center.

## 4.1 Explore Platform as a Service

The Explore Platform as a Service (PaaS), also known as the Advanced Data Analytics Platform (ADAPT), is a private, NASA-supported cloud computing environment built with managed Linux virtual machines (VMs) specifically designed for large-scale data analysis. The platform incorporates storage, compute, container, and cloud computing capabilities. The architecture is comprised of over 300 physical hypervisors that host one or more VMs each. A unique advantage of this environment is that every VM has access to the same centralized data repositories which include dynamic collections of some of the most commonly used NASA-supported datasets such as Landsat, MODIS, MERRA/MERRA-2, and ICESat-2, where newly acquired data is continuously added. The NASA ADAPT cloud was awarded the HPCwire Readers' Choice Award at the 2015 and 2016 International Conference for High Performance Computing, Networking, Storage and Analysis.

The Explore PaaS provides numerous benefits to the scientific community by offering an accessible and cost efficient solution for interacting with large datasets. By dynamically hosting a wide variety of crucial scientific datasets on the cloud it removes the burden of data collection from the user which can save researchers weeks if not months of time searching for, ordering, and downloading data. Instead a user can move their analysis software to the data itself [moving code?] which requires some overhead at first when portaging software to a new cloud-based environment. The upfront cost of learning the new environment and adapting one's code to run seamlessly can seem daunting at first, but over time will save significant time and resources associated with data collection. The Explore PaaS also provides a support team who actively aid users in setting up their own environments.

Apart from the ease of access to data, the platform provides a cost efficient alternative to investing in high-performance computing hardware for Big Data analysis tasks, as any NASA funded principal investigators (PI) can access it for free. The platform is set up such that one can

flexibly scale horizontally, by adding or subtracting cores or machines, and vertically, by increasing or decreasing compute power, with ease and without cost. Therefore a user can scale their own infrastructure up or down depending on the requirements of the task at hand. During a time when the shear amount of available Earth observation data is skyrocketing, this flexibility to scale efficiently with low barrier to entry is becoming increasingly important to the scientific community. Another massive advantage is that the platform removes most of the configuration and maintenance burden from the shoulders of the scientist, thus allowing scientists to focus more on their work rather than learning how to maintain a high performance compute cluster.

## 4.2 DDA on Explore

For this project, I use the DDA-ice to analyze surface topographic characteristics of the GrIS on a seasonal (3-month) basis. The density in coverage of ICESat-2 over Greenland produces around 1500 intersecting ATL03 granules for a 3-month period resulting in 4-6TB of data, which equates to the scale of trillions of geolocated photons. Due to the amount of data and the computational requirements of the DDA-ice, producing a high-resolution surface topographic analysis becomes an extremely computationally expensive task. With the goal of conducting these analyses seasonally, and keeping up with data collection, I needed to utilize a modern cloud computing environment which already hosts the data. It would not have been feasible to perform this study with just locally available compute resources, as the time and space it would require to download the data alone would render the approach incredibly inefficient.

The Explore PaaS configuration provides the user with access to a specific cluster of pre-configured compute nodes can be used for executing large-scale data processing pipelines incorporating any combination of the available data collections. For the purposes of this project, with my advisor being funded as a PI on the NASA ICESat-2 Science Team, I've had direct access to the ICESat-2 compute cluster which consists of 12 individual compute nodes, each equipped with 10 computational cores. Each node in this cluster is a virtual machine running the Rocky Linux 8 operating system. In order to leverage the parallel nature of such an environment, the ICESat-2

compute cluster supports the Slurm Workload Manager, which I was able to use to increase the level of parallelization in my computational pipeline. Slurm allowed me to distribute the work the DDA-ice over multiple compute nodes within the cluster. This way there are two levels of parallelization: 1) modularized, internal parallelization in the density computation of the DDA-ice (described in Section 3.1) and 2) execution-level parallelization for distributing the work over high-performance virtual machines. To give perspective, a 3-month period of data over Greenland typically involves around 1500 ATL03 granules, each comprised of 6 individual data channels, which results in approximately 9,000 iterations of the DDA-ice over a data stream. The execution-level parallelization allows me to distribute these 9,000 individual jobs over multiple compute nodes, while the internal parallelization fully utilizes all 10 cores of each node during computation.

The current setup on Explore utilizes 6 compute nodes with 10 cores each. Before the algorithm execution starts I pre-allocate the work such that each compute node handles roughly the same amount of load. The reasoning for pre-allocating the tasks per compute node is that the available Slurm configuration struggles to keep track of job execution status when dealing with a job involving 9,000+ tasks, and it often produced breakages and inconsistent results. Therefore, by pre-allocating the work, each compute node receives a contained array of tasks to execute. Each task can be thought of as a granule and beam combination. For each granule there are 3 strong and 3 weak beams, and the strong beams always contain more data and thus use more resources. The pre-allocation of work is done such that the strong beam tasks are roughly equally distributed amongst the 6 compute nodes in order to avoid bottlenecks.

Before all the code performance improvements discussed in the previous chapter were implemented, it took well over a month on average to process 3 months of Greenland data. This was due to the fact that frequent memory leak issues forced one to manually intervene and restart the execution, and the code only reliably ran on a single thread. After implementing the various performance improvements and updating the compute environment on Explore to properly meet the new specifications, I am now currently able to process a 3-month period of Greenland data in 5 days or less total compute time without any manual intervention. Table 4.1 shows the overall

execution time for each compute node involved in processing a Fall 2021 Greenland-wide dataset. Computing along-track surface height at 0.7m resolution for the entire Greenland Ice Sheet is an extremely computationally expensive task, but with our workflow this becomes a feasibly efficient task even with basic commodity hardware.

In the future, this workflow could be made even more efficient by incorporating more compute nodes with more cores as more cores per node would speed up algorithm execution and more compute nodes would allow the work to be further distributed, lowering the overall execution time. It could also be made more efficient by allocating the jobs amongst the compute nodes in parallel at runtime. This would avoid manual pre-allocation of work by iteratively distributing each task to the next available compute node until all tasks had completed, eliminating any downtime for the compute nodes.

Table 4.1: Total compute time involved in processing a 3-month period of data over Greenland. The time is per compute node because the work is pre-distributed amongst each node individually before execution begins. The date range for this dataset is 9/16/2021 to 12/16/2021, and includes all ATL03 granules that intersect the GrIS during this time period.

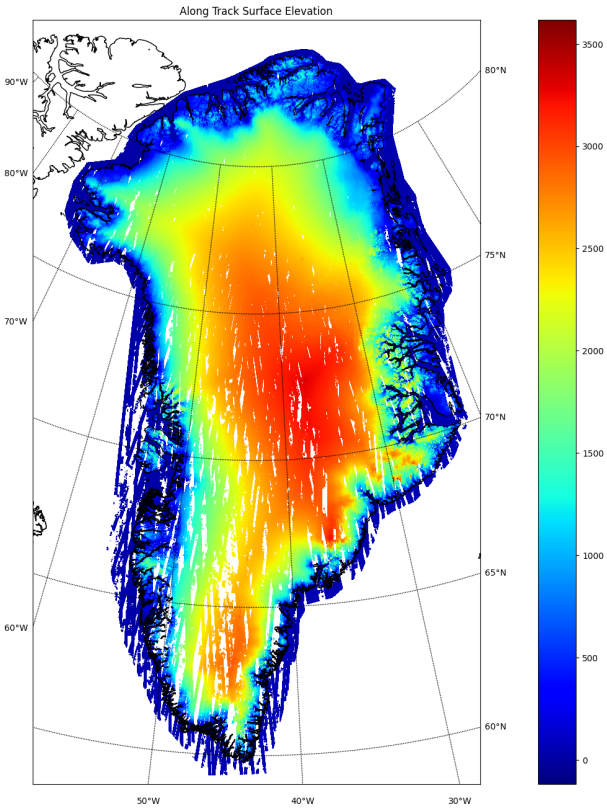
| compute node | total execution time (seconds) | total execution time (days) |
|---|---|---|
| icesat203 | 407,368 | 4.71 |
| icesat204 | 246,358 | 2.86 |
| icesat205 | 321,430 | 3.72 |
| icesat206 | 216,035 | 2.50 |
| icesat207 | 371,008 | 4.29 |
| icesat208 | 232,761 | 2.69 |

## 4.3 Results: Greenland-wide DDA-ice Results

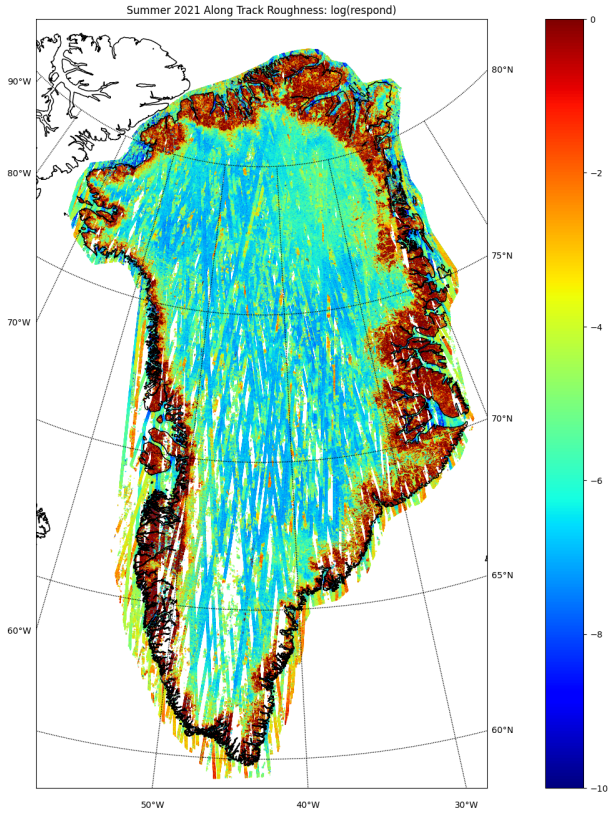
All results presented in this section were processed on NASA's Explore/ADAPT Science Cloud with the DDA-ice algorithm. The DDA-ice parameter values used in these analyses are outlined in Table 4.2.

A time series of seasonal maps of the DDA-ice-interpolated surface elevation and roughness

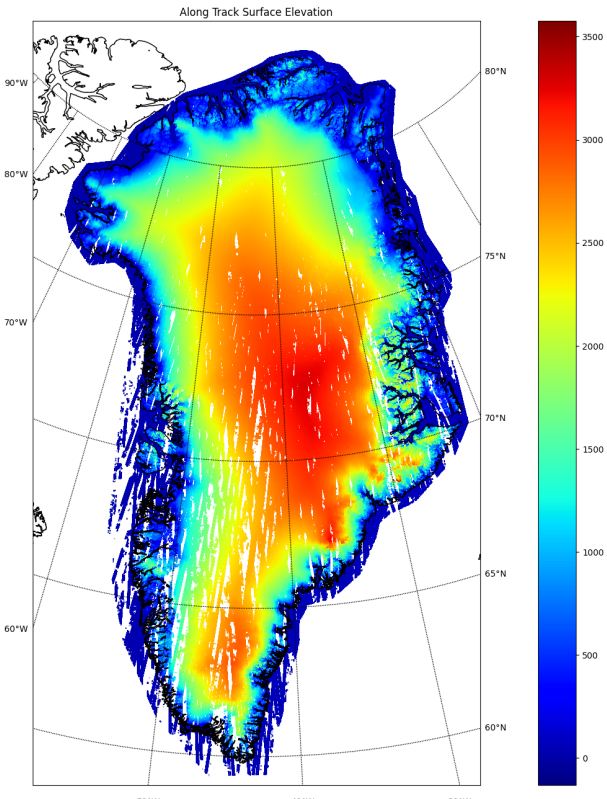
Figure 4.1: Seasonal, 3-month, Greenland-wide DDA-ice surface height and surface roughness. Each figure is a result of processing all ICESat-2 ATL03 granules which intersect the GrIS during the specific time period. Date range for **(a)** and **(b)**: 6/15/2021 to 9/15/2021, date range for **(c)** and **(d)**: 6/15/2022 to 9/15/2022. All results processed on NASA's Explore PaaS with the DDA-ice surface height extraction algorithm, roughness values derived from the maximum value of the residual variogram function over the DDA-ice-interpolated ground height.



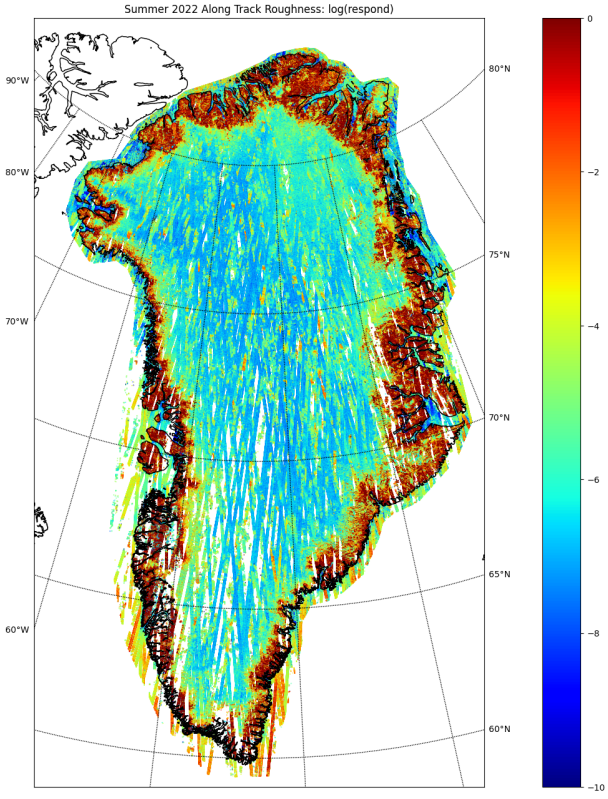

(a) Summer 2021 Elevations




(b) Summer 2021 Roughness




(c) Summer 2022 Elevations




(d) Summer 2022 Roughness

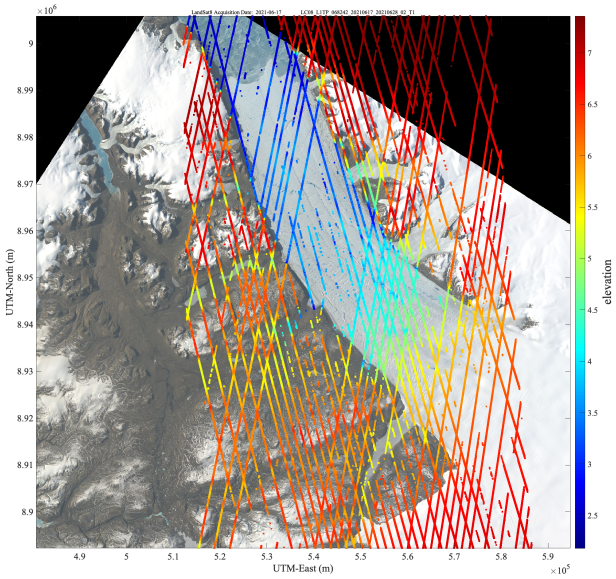
over the GrIS is presented in Figure 4.1. Not every map reflects the same coverage because dynamic atmospheric conditions sporadically obstruct the surface return data, especially in the Southwest region of the ice sheet. Each figure is a product of approximately 1500 ATL03 granules (totaling more than 4TB of data and trillions of individual photons), and around 9,000 individual executions of the high-resolution DDA-ice and corresponding surface roughness algorithm. Each was produced on the NASA Explore PaaS in less than 5 days total compute time.

These images serve as a macro-scale visualizations of the surface topographic characteristics and of the GrIS, and the annual evolution of such. In the surface roughness images (right column of Figure 4.1, one can see just by the eye a distinct margin around the entire GrIS where the surface roughness is significantly higher than the core of the ice sheet. This phenomenon is known as the process margin because throughout the center of the GrIS the surface is smooth, while around the margin exists the important outlet glaciers and other cryospheric processes which cause all of the ice mass loss of the GrIS leading to rises in sea level. These areas are those of most interest to the cryospheric community.

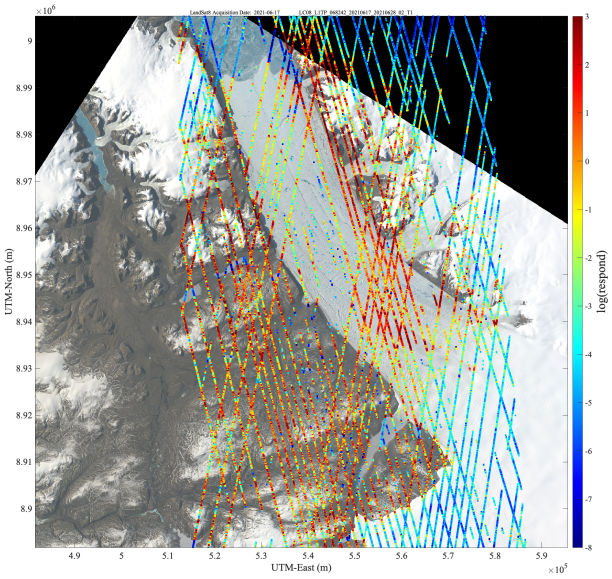
Table 4.2: DDA-ice parameter values for Greenland-wide surface-height analysis.

| Symbol # | Description | Value |
|---|---|---|
| s | standard deviation | 3.0 |
| a | anisotropy | 5.0 |
| q | threshold quantile | 0.5 |
| k | threshold offset | 3.0 |
| l | slab thickness (m) | 50 |
| R | resolution of ground follower (m) | 5.0 |
| r | factor to reduce R parameter over rough ice | 2.0 |
| $\epsilon$ | max DEM tolerance distance for cloud filtering | 200 |
| $\alpha$ | min photon count threshold | 35 |

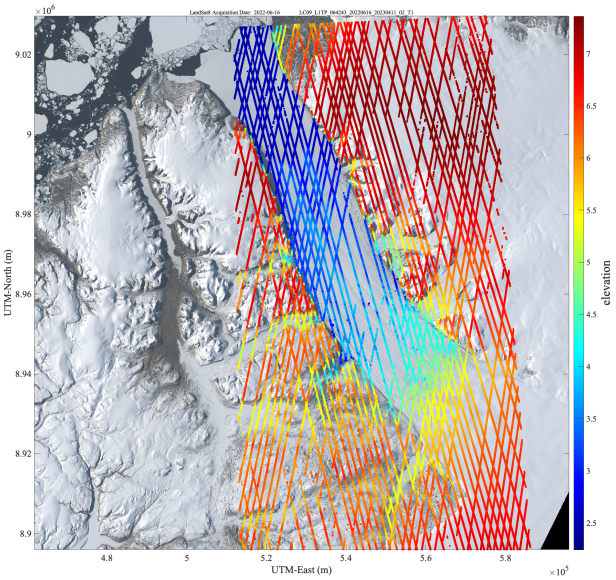
Figure 4.2: Glacier-specific extractions of surface height and roughness from the results visualized in Figure 4.1 overlaying LandSat8 imagery. The glacier represented here is Petermann Glacier, which is a very large outlet glacier located in North-West Greenland. The date ranges coincide directly with those in Figure 4.1.



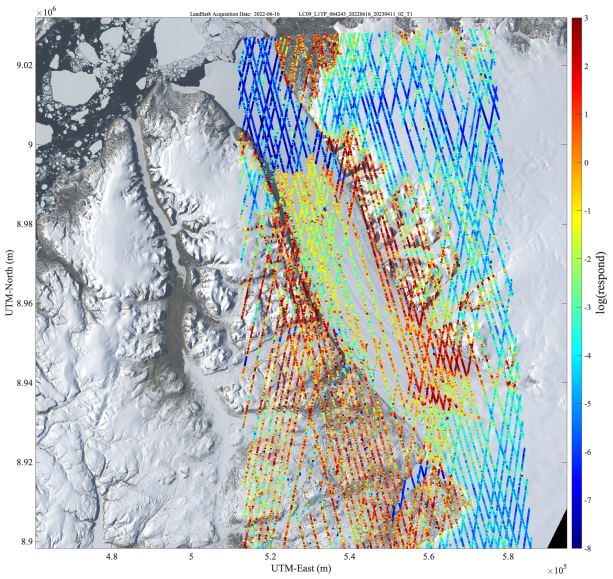

(a) Summer 2021 Elevations




(b) Summer 2021 Roughness




(c) Summer 2022 Elevations




(d) Summer 2022 Roughness

### 4.3.1 Glacier Specific Products

The results illustrated in Figure 4.2 reflect the ability to zoom into regions of interest from the Greenland-wide product with a simple JSON glacier outline file and overlay the results over a LandSat8 image from the same season. This capability is made possible because both the ICESat-2 data and LandSat imagery are hosted on NASA's Explore computing platform. Therefore the pipeline can take in a glacier outline and extract the high-resolution surface topography data in order to combine with satellite imagery for more thorough visual investigation, providing the ability to zoom into specific cryospheric processes such as crevasse fields, fjords, ice streams, and more.

# Chapter 5

# Classification

In this chapter I will present an approach to use a physically informed, feed-forward multi-layer perceptron (MLP) for classifying different ice surface deformation features from LiDAR data. The idea is to use variograms and surface roughness to inform a simple and generalizable model with a task of distinguishing between a small number of high level surface features that can be extended across an entire ice sheet. For this analysis, I will take constant 150m along-track segments of ice surface data and compute the experimental (residual) variogram function over: 1) the DDA-ice-interpolated surface height and 2) the associated signal photons. These two discrete functions will serve as the input features for the classification routine because they present a robust statistical representation of the physical characteristics of a given segment of ice surface data. Using a lag step value of $h_{step} = 5$, I derive these functions by varying $h$ (in Equation 2.8) from $h = 5$m to $h = 150$m, at steps of $h_{step}$ to produce $150/5 = 30$ discrete values for each function. Therefore the feature vector for this classification task is of length 60. The model presented in this chapter will henceforth be referred to as the "DDA-MLP".

## 5.1 Motivation

The surface height and roughness results in Section 4.3 provide high level insight into the surface topography and morphology characteristics of the GrIS. The goal for an ice-surface deformation feature classifier is to provide a more in-depth characterization of the morphological properties of the ice sheet in order to better understand the distribution of certain glaciological processes. To

accomplish this, I will identify a few distinct crevasse patterns representative of certain glaciological deformation processes and extract quantifications of the ice-surface deformation characteristics via residual variogram functions. These functions will be fed into an MLP in order to train a machine learning model to identify different crevasse patterns over the GrIS.

MLP's have proven to be robust in predicting and interpolating environmentally sourced data when classical numerical methods either fail to capture the environmental dynamics or only succeed with a small, contained sample size [7, 34]. Since an MLP requires no prior knowledge, it can be advantageous over numerical models in application areas where the physical dynamics are complex and non-linear, and has been shown to be effective over traditional methods when applied to the atmospheric sciences [7]. A cryospheric area as large as the GrIS presents countless different types of surface deformation, even when just considering crevasses, which is a specific deformation type, one could identify many different physical patterns. Currently no standard exists for distinguishing different crevasse types, as little work has been done to leverage crevasse patterns in glacier dynamic analysis or modeling, and most of the crevasse pattern identification work has focused solely on imagery [16, 4].

Instead of trying to extract a wide range of different crevasse types present on the GrIS from LiDAR data, I aim to use the DDA-MLP as an automated large-scale information filter. This rough information filter is intended to distinguish a few physical phenomena which reflect different important glacial processes such as complex crevassing resulting from numerous phases of deformation over time and freshly opened crevasses from more recent glacial movement or acceleration which could indicate changes in the glacier dynamics. When applied to the GrIS this classification network will aid in identifying areas of interest for deeper analysis and help highlight seasonal or year-over-year changes in different areas of the ice sheet. It also has potential applications in influencing glacier dynamic models which typically do not include any information on surface deformation characteristics.

## 5.2 Previous Work

The field of machine learning based classification of crevasses is still relatively novel, especially when considering methods which do not involve imagery. Herzfeld and Zahner presented the first geostatistical method for image classification in a glaciological context in 2000 [4]. Kodde et al. [27] presented an early method for automatically distinguishing crevassed areas from smooth areas from airborne laser data by comparing the mathematical morphology of a smooth glacier surface to the data in question. Williams et al. [49] and Walker et al. [47] both used Ground Penetrating Radar (GPR) data with a Support Vector Machine (SVM) to automatically detect shallow and deep crevasses over crevassed area of Greenland and Antarctica. The DDA-MLP will provide an alternative to these previous approaches with the benefit of increased stability and resolution of LiDAR data, and without needing to employ a field data collection mission.

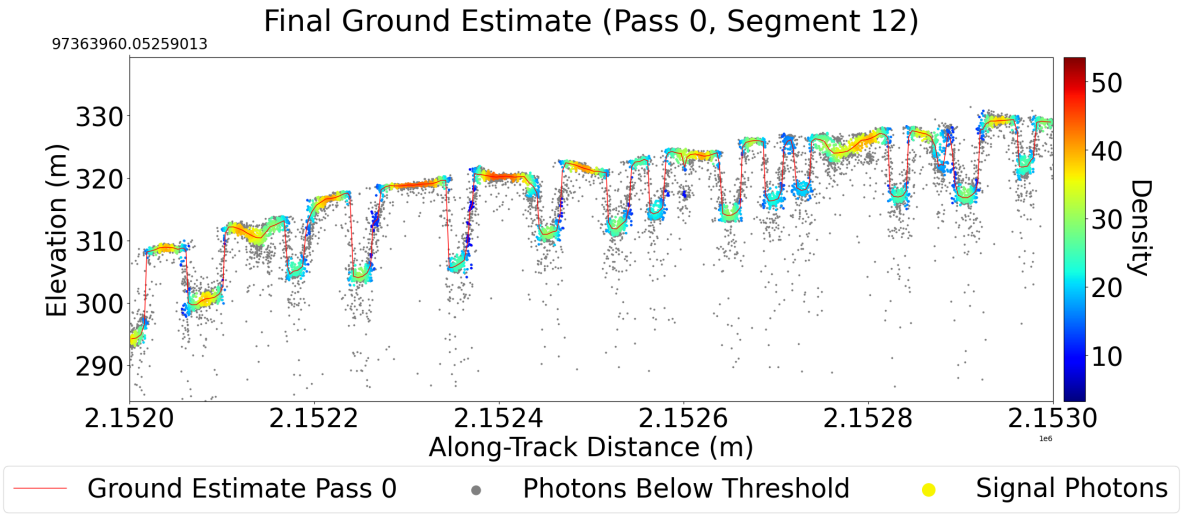
## 5.3 Classes and Labeling

Table 5.1: Descriptions of the surface feature classes to be used in this analysis.

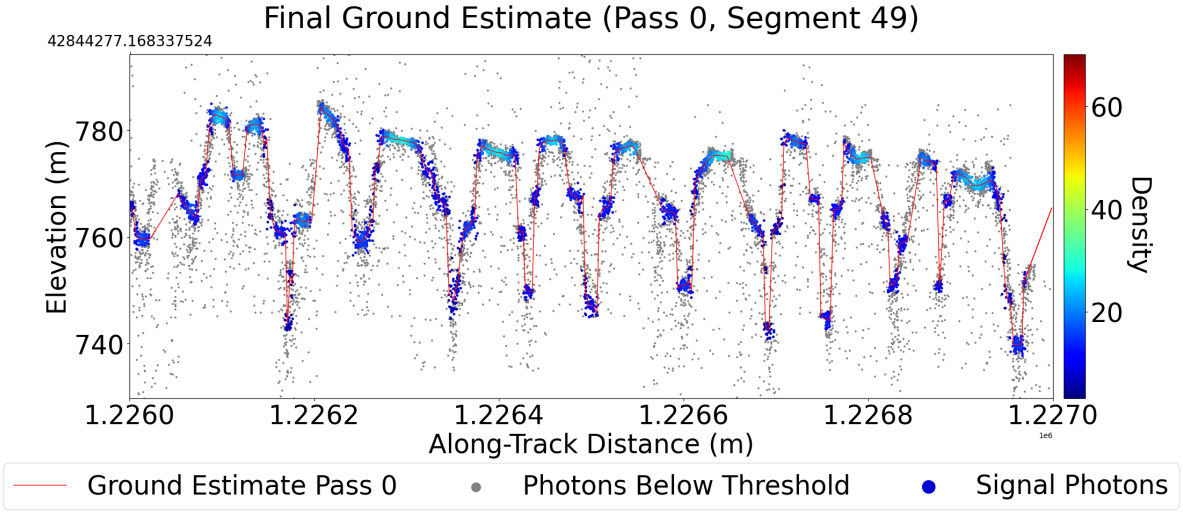
| Class # | Class Name | Class Description |
|---|---|---|
| 0 | Single-event / Surge-type | freshly opened crevasses from more recent glacial movement |
| 1 | Multi-event / Complex | complex crevasse patterns after numerous phases of deformation |
| 2 | Undisturbed Snow | smooth surface / undisturbed snow |

The three ice-surface feature classes outlined in Table 5.1 are reflected in Figure 5.1. These three classes encapsulate the different types of surface features I aim to identify over the GrIS with an MLP-based classifier. What distinguishes the two different crevasse classes is in 5.1a, one can generally see where the primary surface was before fracturing, as these crevasses are relatively fresh forming consistent vertical fractures directly perpendicular to the surface. This phenomenon is indicative of the first phase of deformation when smooth ice fractures do to underlying glacial movement. In comparison, Figure 5.1b reflects a more complex pattern where the shape of these

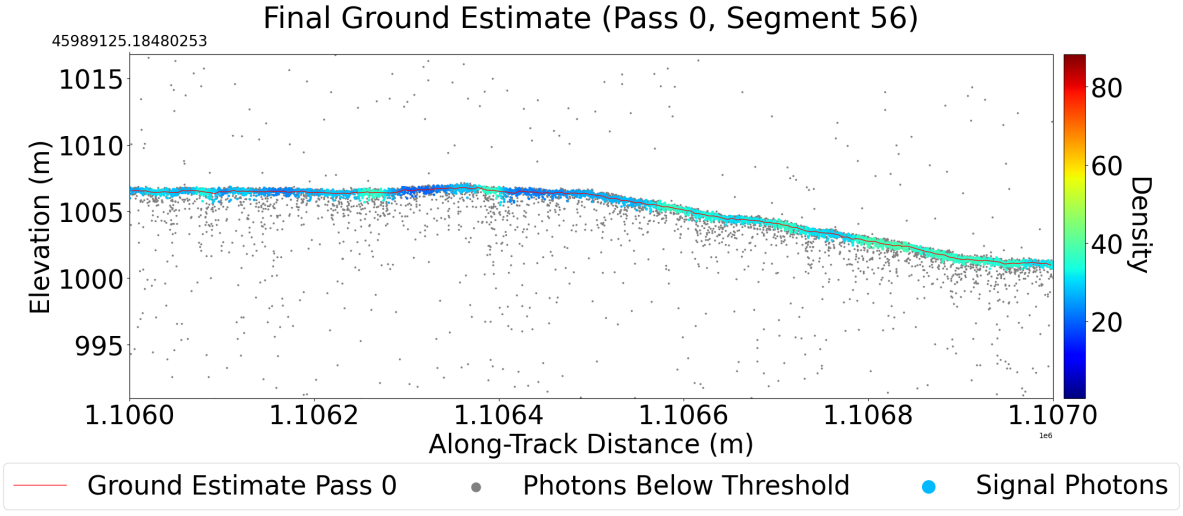
Figure 5.1: 1km segments of DDA-ice interpolated ground estimates over signal photons: **(a)** is an example of class **0 - Single-event / Surge-type**, **(b)** is an example of class **1 - Multi-event / Complex**, and **(c)** is an example of class **2 - Undisturbed Snow**. Granule info for **(a)** : ATL03_20210131212737_05941003_005_01.h5 (Negribreen Glacier System). Granule info for **(b)** and **(c)** : ATL03_20190511210825_06660305_005_01.h5 (Jakobshavn Isbræ Glacier).




(a) Surge-type Crevasses




(b) Complex Crevasses




(c) Undisturbed Snow

crevasses is not shallow and straight up and down, but rather they are deeper with wider openings and have less uniformity. This pattern is a result of the surface having undergone multiple phases of deformation and glacial movement, causing the once shallow, parallel crevasses to morph over time. The third class (Figure 5.1c) is included to distinguish smooth surfaces from deformed surfaces.

Since no public, labeled dataset currently exists which encapsulates different ice-surface deformation features sourced from LiDAR data, I created one myself for training and testing the DDA-MLP. I sourced most of the training data from my Greenland-wide DDA-ice results presented in Section 4.3. I derived most of the class 1 - multi-event / complex crevasse and some of the class 0 - single-event / surge-type crevasse pattern examples from Jakobshavn Isbræ which is a well-studied glacier in Southwest Greenland, with an constantly fast-moving ice-stream which presents unique morphological patterns. I also derived many of the class 0 examples from the Negribreen Glacier System in Svaldbard because it is an actively surging glacier with representative crevasse patterns [21]. The rest of my training examples came from iteratively flipping through 150m segments of data over the Greenland-wide DDA-ice results. The labeling process which I designed involves visualizing a plot of the DDA-interpolated ground estimate over the associated signal photons, for a 150m segment of data, and associating the segment with one of the three classes, or ignoring it based on data quality or feature scope. In order to limit the number of plots I needed to look through to create a usable dataset I used surface roughness as a filter because in cases of smooth ice (class 2) the roughness is low and when the surface is crevassed, the roughness is high. Therefore, when collecting examples for class 2 - undisturbed snow / smooth ice I only examined examples which had a $log(surface\_roughness) < 1$, and to collect examples for the other 2 crevasse classes I only examined examples which had a $log(surface\_roughness) > 2$. This roughness criteria greatly accelerated the process of labeling images, and highlights the potential value of using variograms as indicators of surface type. The final training dataset is represented in Table 5.3.

## 5.4 Architecture

The general architecture of DDA-MLP is a feed-forward MLP which takes as input features a pair of experimental residual variogram functions, one over the DDA-interpolated ground estimate, and another over the associated signal photons, combined into a single 1D vector. Each input feature reflects a morphological quantification for a 150m along-track segment of ice surface, and the model predicts which of the predefined surface-type classes the segment best aligns with.

The DDA-MLP consists of an input layer, 3 fully connected hidden layers, and an output layer. Fully connected means that every neuron from layer $i-1$ is mapped to every neuron in layer $i$ via individual weights and biases. The input layer receives the variogram input feature, and forwards the data linearly into the hidden layers. Each neuron in the input layer corresponds to a single value in the input feature vector. The hidden layers are then responsible for transforming the data through a series of nonlinear activations applied to summations of weights and biases (see Equations 5.1 and 5.2). The reason for including multiple hidden layers is such that the network can learn hierarchical representations of the data by extracting different features at each layer.

Since the DDA-MLP is designed to predict complex, dynamic ice-surface features using non-linear variograms, it is necessary to introduce some nonlinearity into the classification network. The nonlinear activation function employed in the DDA-MLP is the Exponential Linear Unit (ELU) which was originally presented by Clevert et al. in [5] as a way to speed up the learning process for neural networks and produce higher classification accuracies. The popular Rectified Linear Units (ReLU) activation function is zero for any input $< 0$, and linear for input values $> 0$, which produces a non-zero mean activation value, and this can cause a bias shift to the next layer, propagating bias forward through the network [5]. ELU allows for negative output values which pushes the mean activation value closer to zero [5], which is known to speed up the training process [5, 41]. It has also been shown to produce a more stable loss progression during training [5]. Equation 5.3 demonstrates the mathematical function for the ELU activation, and it should be noted here that the hyperparameter $\alpha$ controls the value at which an ELU saturates for negative values [5]. In

this case I use $\alpha = 1$ After trying different, well known activation functions available in PyTorch, such as ReLU, Leaky ReLU, Softplus, and Softmax, I determined that the ELU activation function produced the most stable results for the DDA-MLP.

Let $N$ be the number of neurons from the previous, fully connected layer, let $a_i$ be the input value and $w_i$ be the weight for the $i^{th}$ neuron in the previous layer. Let $b$ be the offset bias. Let $k(x)$ be the activation function. Let $z_{out}$ be the output value for the current neuron.

$$z = b + \sum_{i=1}^{N} a_i w_i \tag{5.1}$$

$$z_{out} = k(z) \tag{5.2}$$

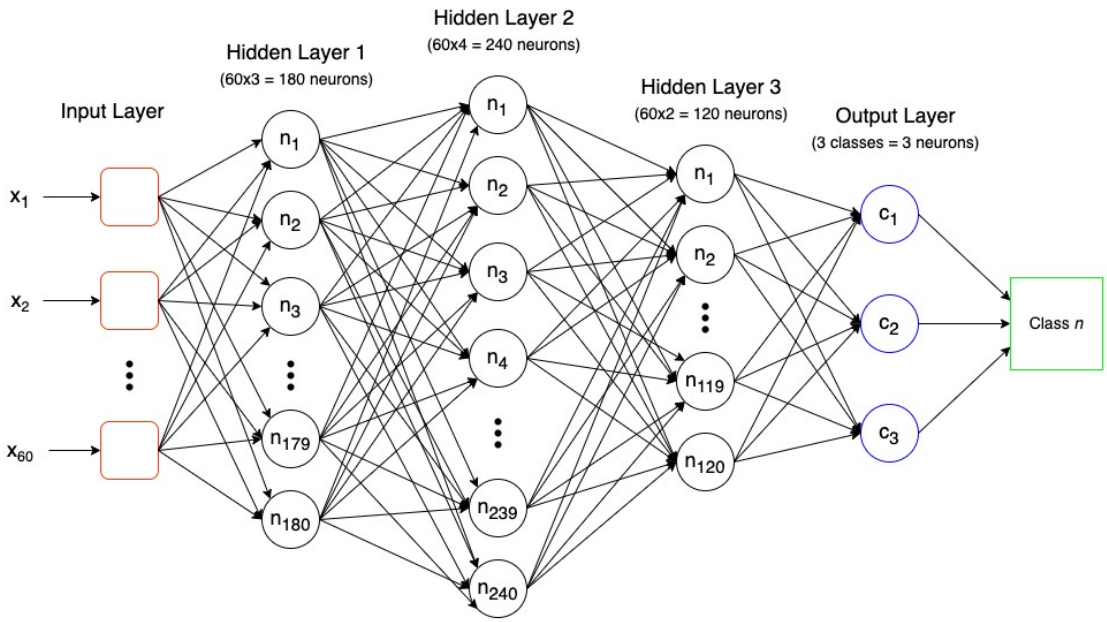
$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases} \tag{5.3}$$

Finally, after the ELU activation is applied to the output of the final hidden layer, a higher-level representation of the data is forwarded to the output layer. Since this is a multi-class classification problem, the output layer contains one neuron per class, which in this case is 3. Every neuron in the final hidden layer is mapped to each of the 3 neurons in the output layer as described in Equations 5.1 and 5.2. Each neuron in the output layer formulates a probability value for how likely the current training example is to belong to the class associated with it. The model makes its prediction based on the maximum probability value of the output neurons.

Determining the most appropriate number of hidden layers, and the number of neurons to include in each hidden layer, is a difficult problem which faces anyone wishing to leverage an MLP. Adding more hidden layers and neurons increases the complexity of the model which can be beneficial when learning from inherently complex data, but too much complexity and the model becomes susceptible to overfitting to the training data. Too few hidden layers can produce a model which lacks the complexity necessary to capture the full variance of the data, and will suffer from underfitting during training. For the DDA-MLP, I experimented with 1, 2, and 3 hidden layers

with varying number of neurons. I chose to scale the number of neurons for each hidden layer as a factor of the input feature vector length, which is 60. After heuristically testing numerous different combinations I settled on the best performing architecture of [3, 4, 2], which indicates 3 fully-connected hidden layers which contain 3, 4, and 2 times as many nodes as the input layer respectively. Table 5.4 highlights some of the different hidden layer combinations I tried and the corresponding minimum loss value achieved. See Section 5.5 for more details on determining the final DDA-MLP architecture during training. The overall architecture of the DDA-MLP can be visualized in Figure 5.2.

Figure 5.2: Architecture of the DDA-MLP with each layer having a different number of neurons, each layer is fully connected.




The last two model hyperparameters which must also be heuristically determined are the learning rate and batch size. Both affect the generalizability of the model. The model's learning rate controls the magnitude of updates to the weights and biases made during each iteration of the gradient descent optimization algorithm. A higher learning rate can cause the model to converge quickly, but also risks unstable training and getting stuck in local loss minima. On the other

hand, a lower learning rate implies much smaller and more incremental updates to the weights and biases leading to slower convergence, but can help avoid overfitting the model to the training data. Batch size refers to the number of input examples which get fed forward through the MLP before backpropagation is performed during training. With a larger batch size, the model sees a greater variety of examples for which to tune its weights and biases, which can improve the model's generalizability. Too large of a batch size relative to amount of training data can result in a poorly directed gradient during backpropagation. For the DDA-MLP, batch sizes of 2, 4, 8, 16, 32, and 64 were tested with the best performing size being 32 for the current configuration. Learning rates were tested between a range of $1 \times 10^{-7}$ and $1 \times 10^{-5}$, with the most optimal configuration using a value of $1 \times 10^{-6}$ . See Table 5.2 for a complete breakdown of the final model hyperparameters for the DDA-MLP.

Table 5.2: Final configuration of DDA-MLP hyperparameters

| Parameter | Value |
|---|---|
| variogram lag step value, $h_{step}$ | 5 |
| # hidden layers | 3 |
| # neurons per hidden layer | [180, 240, 120] |
| activation function | ELU |
| learning rate | $1 \times 10^{-6}$ |
| batch size | 32 |
| # training epochs | 1500 |

## 5.5 Training

Training an MLP is an optimization problem with the goal to optimize the model's internal parameters, like those in Equation 5.1, based on the training data. At each iteration of training, the DDA-MLP predicts the class of each training example and uses the cross entropy loss function to determine how far off its prediction was in order to update its parameters. Entropy was first introduced by Claude Shannon in [42] as a way to quantify the level of uncertainty of a random variable

Table 5.3: Number of hand-labeled training data examples per class used in training of the DDA-MLP.

| Class # | Class Name | # Examples |
|---|---|---|
| 0 | Single-event / Surge-type | 1496 |
| 1 | Multi-event / Complex | 1485 |
| 2 | Undisturbed Snow | 1494 |

based on the range of possible outcomes (See Equation 5.4). In the case of MLP-classification, the possible outcomes are the discrete classes and the probabilities are those which the model assigns to each output neuron. The DDA-MLP uses cross entropy loss, as outlined in Equation 5.5, as its loss criterion. The optimization problem is then for the model to learn an internal parameter set which minimizes this loss function, and to accomplish this the DDA-MLP employs stochastic gradient descent (SGD) via the Adam algorithm for first-order gradient-descent based optimization problems introduced by Kingma and Ba in [26]. During training, backpropagation, as defined in [12], involves computing the gradient of the loss layer by layer, starting from the output and moving backward towards the input layer. In this case, the Adam algorithm for SGD only computes the first-order gradient, and employs adaptive learning rates for parameters based on estimates of first and second order moments, and updates the parameters proportionally to the learning rate hyperparameter in the direction of steepest descent of the gradient [26]. The pseudocode for this process is presented in Figure 5.3.

$$H(X) = -\sum_{i=1}^{n} p_i log(p_i) \tag{5.4}$$

Where X is a discrete random variable, and $n$ is the number of possible outcomes with $p_i$ being the probability of the $i^{th}$ outcome.

$$Loss_{ce} = -\sum_{i=1}^{n} t_i log(p_i) \tag{5.5}$$

Where $n$= number of classes, $t_i$ is the truth label, and $p_i$ is the model-predicted probability for the $i^{th}$ class.

The DDA-MLP is trained on a hand-labeled dataset that was created iteratively as described in Section 5.3, with the training data class breakdown depicted in Table 5.3. For generalizability, the training data is randomly divided into 80% training data and 20% validation data before beginning the training process. The DDA-MLP also uses batch optimization, so each update to the model parameters is based on predicting over a batch, $> 1$, of training examples which improves the model's generalizability. This way the model is trained on 80% of the labeled data, and validated by applying the updates to the held-out validation data. The goal of the training process is for the loss to converge and to minimize the validation loss in order to ensure the model will be generalizable and able to predict on data it has yet to see. During each training epoch the following pseudocode reflects what is happening:

```
For batch_x in train_data:
     Y = model.predict(batch_x)
     loss = cross_entropy_loss(Y, Y_true)
     model.backprop(loss) # update model weights/biases based on loss criterion
     SGD.step() # 1 step of stochastic gradient descent to update model params

# predict on validation data based on model param updates from above
for batch_z in validation_data:
     Y = model.predict(batch_z)
     loss = cross_entropy_loss(Y, Y_true)
     total_loss += loss
```

One of the best ways to assess model performance is by the convergence of training and validation loss curves throughout the training loop. A strong predictive model should be generalizable to data it has not seen yet, so when the model updates its weights based on the training

Figure 5.3: The Adam SGD optimization algorithm for training neural network parameters, as presented in [26]. Here the step size parameter $\alpha$ is equivalent to the learning rate hyperparameter.

**Algorithm 1:** *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. $g_t^2$ indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With $\beta_1^t$ and $\beta_2^t$ we denote $\beta_1$ and $\beta_2$ to the power $t$.

**Require:** $\alpha$: Stepsize
**Require:** $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
**Require:** $f(\theta)$: Stochastic objective function with parameters $\theta$
**Require:** $\theta_0$: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  **while** $\theta_t$ not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep $t$)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  **end while**
  **return** $\theta_t$ (Resulting parameters)

data it should also improve the model's performance on the validation data. Ultimately, one would expect the training and validation loss curves to converge at similar rates if the model is indeed generalizable. Divergence, where the training loss curve diverges from the validation loss curve, would indicate the model is overfitting to the training data (See Appendix A.2 for examples of divergence). This can be caused by numerous factors such as a model which is too complex (i.e. too many hidden layers or neurons), a learning rate which is too high causing weight updates that are too large and leading the optimizer to get stuck in local minima, and a batch size which is too small causing overfitting to the training data due to over updating the model's weights. Each of these cases leads the model to learn a combination of weights that is too specific to the training data, and is thus not generalizable to unseen data. In Figure 5.4, the DDA-MLP's training and validation loss curves decrease at a similar rate without diverging, reflecting a strongly generalizable model. Examples where the model's hyperparameters caused a divergence in the training and validation loss curves, indicating overfitting to the training data, can be visualized in Appendix Section A.2.

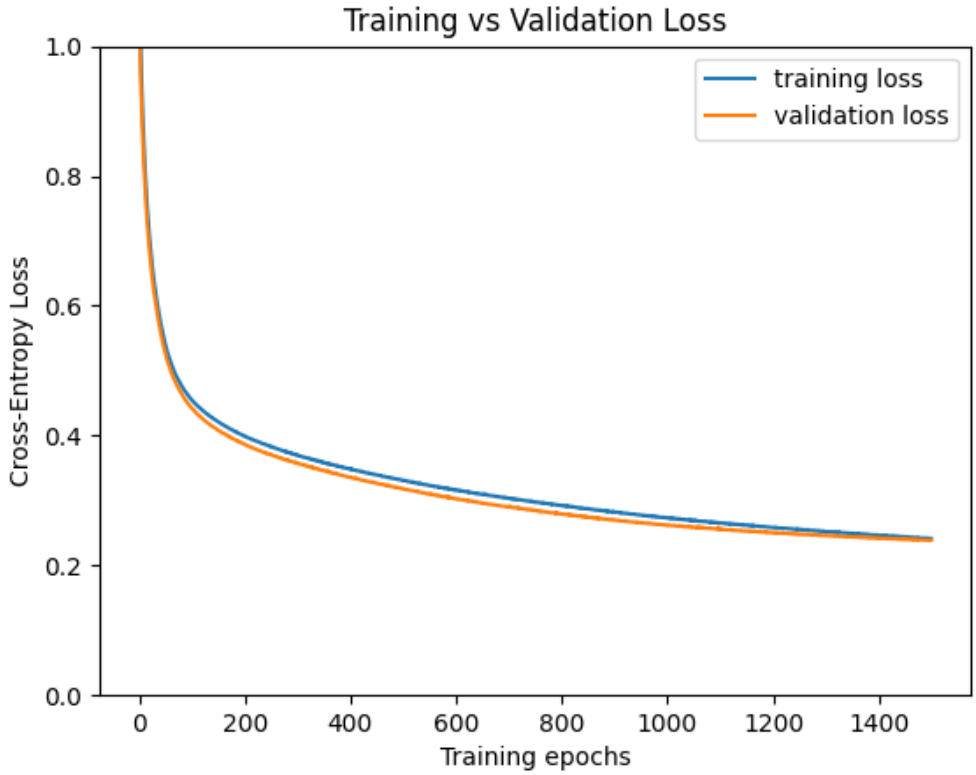
Table 5.4: A variation of DDA-MLP models trained with different hidden layer configurations train and the minimum validation loss achieved for each.

| Hidden Layer Shape # | Minimum Validation Loss Achieved |
|---|---|
| [2, 3] | 0.29 |
| [0.5, 2, 3] | 0.25 |
| [6, 9, 7] | 0.38 |
| [3, 4, 2] | 0.21 |

## 5.6 Results

The final configuration for the DDA-MLP, with the loss function reflected in Figure 5.4, was selected based on the convergence of such loss functions indicating a strong level of generalizability, and the minimum validation loss achieved was lowest for this configuration compared to all others which were tried (See Table 5.4). The selected configuration for the DDA-MLP was tested on a

Figure 5.4: Training and validation loss curves for the DDA-MLP using the cross entropy loss criterion and the Adam algorithm for SGD, as presented in [26].
Model hyperparameters: hidden layers = [3, 4, 2], learning rate = $1e-6$, batch size = 32, num epochs = 1500.



separate test dataset of 1513 hand-labeled examples. The model boasted an overall accuracy of approximately 65.2% with the per-class label breakdown illustrated in Table 5.5.

Table 5.5: Number of hand-labeled test data examples per class used in evaluation of the DDA-MLP.

| Class # | Class Name | # Examples |
|---|---|---|
| 0 | Single-event / Surge-type | 551 |
| 1 | Multi-event / Complex | 406 |
| 2 | Undisturbed Snow | 556 |

Table 5.6: **Actual (y-axis) vs. Predicted (x-axis)** Confusion matrix for the DDA-MLP performance on hand-labeled test data.

| | 0 | 1 | 2 | Accuracy % |
|---|---|---|---|---|
| 0 | 274 | 255 | 22 | 49.73 |
| 1 | 204 | 202 | 0 | 49.75 |
| 2 | 40 | 6 | 510 | 91.73 |

### 5.6.1 Application to Greenland

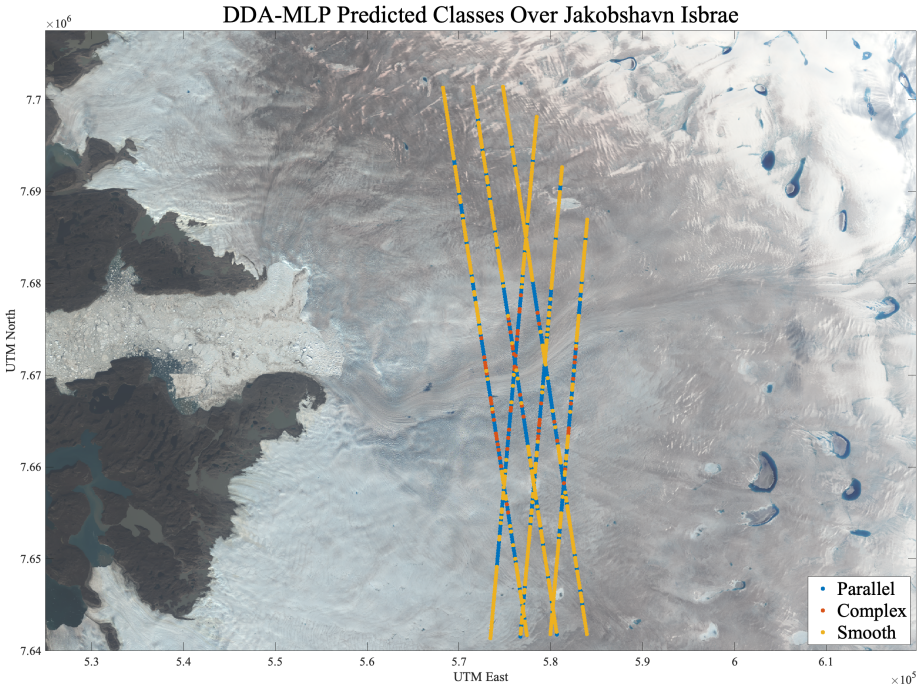
A good way to qualitatively assess the value of the DDA-MLP in glaciological terms is to apply a trained model to classify ground segments over Greenland, and see if it highlights certain processes that are known to exist. Firstly, I wanted to visualize the predicted classifications over Jakobshavn Isbræ Glacier overlaying a satellite image in order to see if the model was roughly distinguishing crevassed areas from smooth areas, and further if it was able to distinguish the two different crevasse types found on the glacier. This visualization is found in Figure 5.5. Next, I aimed to visualize the DDA-MLP predictions on the scale of the entire GrIS. To do this, I took a small subset of results from the Summer 2021 Greenland-wide DDA-ice results, and computed the residual variogram functions over the interpolated ground height and signal photons to create input features for the DDA-MLP. Then, I fed these through the pre-trained model to get class predictions and projected the results over the GrIS. This visualization is found in Figure 5.6.

### 5.6.2 Interpretation of Results

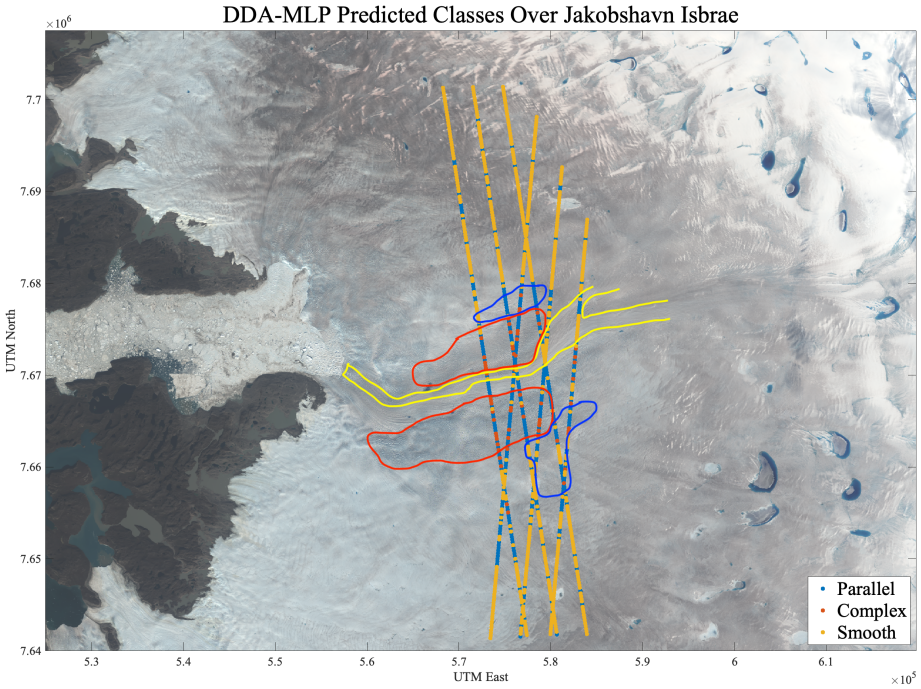
The DDA-MLP accuracy results over test data presented earlier in this section highlight the potential of such a method for ice-surface morphology classification. The confusion matrix and accuracy statistics in Table 5.6 reflect a model which can accurately distinguish between crevassed and smooth surfaces, but struggles to separate the two crevasse classes from each other. However, the approach serves as an interesting proof-of-concept for using a physically informed neural network to classify surface deformation features from LiDAR data.

Ultimately, a glaciological interpretation is needed to fully assess the current value of the physically-informed DDA-MLP. In order to conduct such an evaluation I chose to visualize the model's predictions over Jakobshavn Isbræ in Figure 5.5 because I already have significant domain knowledge on the morphological structure of the glacier. This figure further demonstrated the model's ability to separate smooth ice from crevassed ice as the Jakobshavn ice stream flows through the middle of the image from right to left creating somewhat of a ripple effect of crevasse patterns

Figure 5.5: DDA-MLP predicted classes over **Jakobshavn Isbræ Glacier**. **(a)** is the outputted classification map, and **(b)** is the same map with annotated crevasse zones. In **(b)** the yellow annotated zone is over the center of the ice stream where the ice is known to be smooth, the red annotated zones are the shear zones where complex crevasse patterns exist and the blue annotated zones are where new crevasses have recently opened on the margin of the glacier. Date range for ICESat-2 data: 6/15/2021 to 9/15/2021. Landsat8 image: LC08_L1TP_009011_20210823_20210901_02_T1

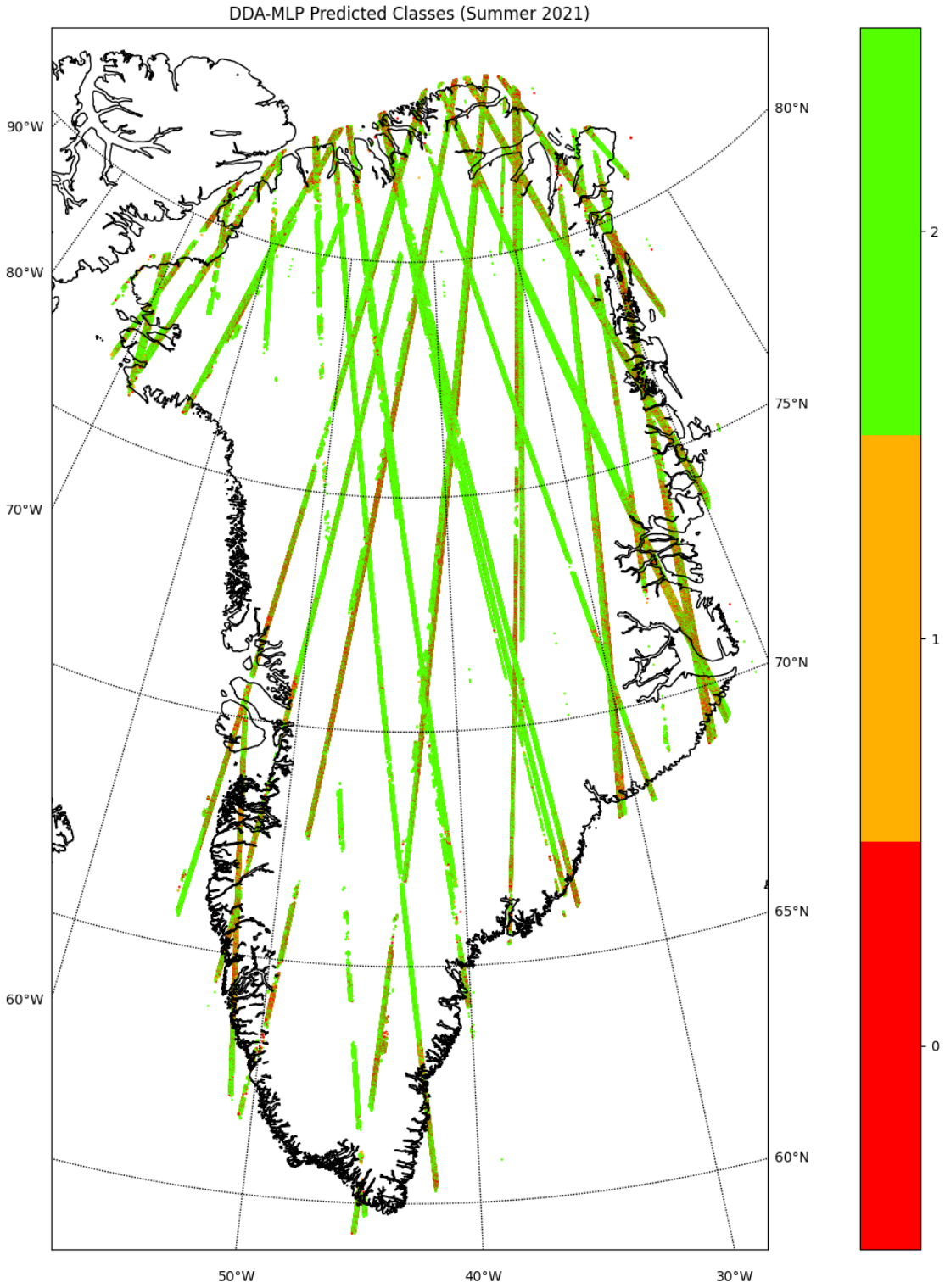



(a) Crevasse Classification Map




(b) Annotated Crevasse Zones

Figure 5.6: Greenland-wide DDA-MLP predicted classes projected over the Greenland Ice Sheet. **0** = single-event / surge-type crevasses, **1** = multi-event / complex crevasses, and **2** = undisturbed snow. Date range for ICESat-2 data: 6/15/2021 to 9/15/2021.



extending from the ice stream. The upper and lower areas of the figure, further from the central ice stream, are areas with few crevasses and mostly smooth ice, as identified by the DDA-MLP. The annotated zones in Figure 5.5b are the result of crevasse pattern zonation of Jakobshavn Isbræ first presented by Mayer and Herzfeld in [32]. The yellow zone reflects the center of the ice stream where the ice is constantly flowing at high rates creating a smooth surface depressed into the glacier. The model does a decent job of identifying these smooth areas in the middle of heavily crevassed areas. The red zones are those of the shear margin of the ice stream which have undergone numerous phases of deformation due to the constantly fast-moving ice stream, primarily forming complex crevasse patterns, like those expected in class 1 - multi-event / complex. The model is able to identify these areas as crevassed surfaces, and most of the class 1 predictions fall inside of these shear zones which demonstrates the potential of such a method. Finally, the blue zones in Figure 5.5b are those just outside the shear zone which have experienced far less stress from the ice stream, but have still produced crevasse fields which reflect more of a surge-type structure. The model at least does a fairly decent job of correctly identifying these zones, even if it tends to over classify crevassed areas as class 0 - single-event / surge-type.

From Figure 5.6 one can extend this interpretation to the entire Greenland Ice Sheet. With class 2 - undisturbed snow reflected in neon green it makes sense that most of the predictions, especially towards the center of the ice sheet, are this color because other than the margins, Greenland is smooth. There are definitely some discrepancies present where the model is predicting crevasses, red and orange, throughout an entire track across the center of the ice sheet. However, looking at the results around the margin of the ice sheet in Figure 5.6, specifically the Eastern most area, one sees more red and orange predictions indicative of crevassed areas, which one would expect to see. From the Greenland-wide surface roughness plots presented in Figure 4.1, it is evident that the marginal areas reflect most of the deformation present throughout the ice sheet. This is reinforced, to an extent, from the results in Figure 5.6, where the DDA-MLP predicts surface deformation at much higher concentrations throughout the margins compared to the central area of the ice sheet.

# Chapter 6

# Discussion and Conclusion

Overall, the approach demonstrated in this research presents a cloud-based, multi-stage pipeline for automated, high-resolution glacial analysis leveraging proven methods for cryospheric analysis and advanced computing techniques. LiDAR is a rather underutilized source of earth observation data, but the active nature of the sensor presents certain advantages over other remote sensing techniques [44]. The Greenland-wide surface height and roughness products presented in Section 4.3 provide an efficient and effective method for large-scale, LiDAR-based cryospheric topography and morphology analysis, with capabilities to incorporate high-resolution satellite imagery for a more small-scale regional study, as presented in Section 4.3.1. This method provides the ability to visualize macro scale changes of topography of the GrIS and track year-over-year changes of specific areas of interest throughout the course of the ICESat-2 mission which began in 2018 and will continue collecting data until 2025 [is this correct?]. The approach will also be applicable to any future missions using high-repetition rate LiDAR sensors.

The impressive ability to take a high-resolution, computationally intensive, surface-height interpolation algorithm like the DDA-ice, and extend it efficiently to the scale of the entire GrIS presents a novel approach to glacial morphological analysis without sacrificing efficiency for resolution. Using this approach, the community can leverage the full capabilities of ICESat-2 for cryospheric analysis, as its freely available datasets can eliminate some of the previous need for costly airborne data collection missions. For example, since ICESat-2 repeats its orbit every 91 days scientists studying a specific region in the cryosphere, such as a surge glacier, can analyze the

evolution of the topography seasonally in an automated fashion. Therefore the community gains access to previously unprecedented frequency and longevity of such high-resolution observations.

The DDA-MLP results presented in Section 5.6 serve as an interesting proof-of-concept for an MLP-based classification network for extracting surface morphology features from LiDAR data in an automated fashion. The idea for the DDA-MLP is to build on the Greenland-wide surface roughness maps in Figure 4.1 in order to add a layer of depth and specificity to the identification of the morphological processes present on the ice sheet. The surface roughness maps are quite effective at highlighting the process margin of the ice sheet, which validate the potential for using spatial variograms for ice-surface morphological analysis, and the DDA-MLP is intended to learn connections between these morphological characteristics and specific ice-dynamic processes. Even though the classification network is still in early stages of development it already demonstrates the ability to separate smooth ice from crevassed ice and from the results over Jakobshavn Isbræ in Figure 5.5 it shows potential for generally identifying areas of complex crevasse patterns resulting from numerous deformation phases. Most importantly, since the classification model is physically informed it lends itself to be more interpretable to the scientific community, thus making it potentially more valuable than a more black box approach such as a Convolutional Neural Network.

## 6.1 Future Work

The methodology for producing Greenland-wide surface elevation and roughness products presented in this research could be of great value to the cryospheric community. The pipeline could be packaged into a tool and deployed on the NASA Explore IaaS for the community at large to use. The potential benefit of this approach would be that researchers could use the tool for any glacier or region in the cryosphere to analyze and track different ice-surface features without being forced to fully understand the underlying algorithms, or even worse, develop their own approach from the ground up. Ultimately, this would remove significant burdens of algorithm development and data collection from the researcher, and allow them to focus their energy on the science. It is also inevitable that more servers, more VMs, and an overall stronger compute infrastructure (i.e. better

HPC technology) will only improve the efficiency and make it more accessible for the community. There is also potential for portaging a tool like this to a language like C instead of Python would significantly improve the parallel computing efficiency because of how Python's global interpreter lock (GIL) limits the parallel compute capabilities.

A significant challenge for the DDA-MLP approach was creating a representative training dataset from scratch for the three different ice-surface feature types identified in this work. Defining distinct classes of ice-surface types over Greenland is something that requires significant domain knowledge and field observation data, as it can be difficult to distinguish surface types from the photon point-cloud alone. More refinement of the training data along with the inclusion of other types of complementary data, such as satellite imagery or field observations, could lead to a much more robust training dataset which would serve to improve the current DDA-MLP classification results. With a more accurate classification model there is strong potential for integrating this methodology to improve current efforts in ice-dynamic modeling. Over time, as more data is observed and more domain knowledge is integrated, more ice-surface deformation patterns are due to present themselves. Such patterns can then be associated with specific ice-surface features which are representative of different dynamic processes. With such continued observations the connection between ice-surface signatures and underlying glacier dynamics will become stronger. Therefore, the modeling of ice-dynamic processes can be improved upon by integrating the physical manifestations of such processes in the form of distinct ice-surface deformation features.

Another major obstacle and potential area for improvement is the development of more representative variogram functions, specific to the ice-surface features under investigation. Many factors go into determining an ideal residual variogram function, as defined in Equation 2.8, such as, 1) how large of an ice segment to sample from 2) which value to use for lag, $h$, and potentially how many lag steps to use for the function, and 3) how many data points, in this case geolocated photons (or the $x_i$'s), must be present in a window of data to compute a valid variogram function. For 1), this could vary based on geographic area or surface type or it could be a static length, and much experimentation could be done on different surface types with the inclusion of field observations to

improve the variogram function. Similarly, there is the question in 2) of whether one should vary the lag value or keep it static, and furthermore there is a question of how many lag steps to use on a fixed or variable window of data to produce a spatially representative variogram function. These factors could use more experimentation with the inclusion of more data over different surface types which would improve the fit of such a residual variogram function based on the preliminary results presented in this work (Section 5.6). While formulating such a function for dynamic ice-surfaces proves to be quite difficult, it has already been shown that residual variogram functions can be valuable quantifications of spatial structure as demonstrated in the surface roughness representation of Greenland in Section 4.3. The overall quality of the classification method presented in this paper relies on spatially representative residual variogram functions, therefore by incorporating more data and experimentation into determining an ideal parameter set this method can certainly evolve from a proof-of-concept to an applicable method for classifying surface types from geolocated photon data.

The proof-of-concept results in this research demonstrate significant potential for an automated, LiDAR-based feature detector that could be extended to adjacent application areas where LiDAR data is available. The DDA-MLP approach could then be extended to any different types of surface topography classification if optimal domain-specific variogram models are developed. One potential source of value that this classification method presents is in conjunction with a satellite imagery-based ice-surface classification task. Over crevassed ice surfaces, ICESat-2 data provide a uniquely accurate representation of the depth dimension because the LiDAR sensor is able to return geolocated photons at extremely high resolution below the surface. However, the LiDAR sensor only picks up one dimension along track which limits the ability to use the data to generalize over a wider spatial area. In combination with satellite imagery this depth dimension provides a unique way to more accurately estimate surface morphology. With both imagery and LiDAR, one could compute a residual variogram function over a 2-dimensional area (i.e. not just along track) and include LiDAR data points to get the depth of different deformation features, producing a variogram function more representative of the actual surface. This demonstrates the immense po-

tential for a physically-informed ice-surface classification method built upon the concepts presented in this work.

## Bibliography


[1] María Alonso and J Malpica. Satellite imagery classification with lidar data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 38, 05 2012.

[2] Emery D. Berger, Sam Stern, and Juan Altmayer Pizzorno. Triangulating python performance issues with scalene. 2022.

[3] K. M. Brunt, T. A. Neumann, and B. E. Smith. Assessment of icesat-2 ice sheet surface heights, based on comparisons over the interior of the antarctic ice sheet. Geophysical Research Letters, 46(22):13072–13078, 2019.

[4] Ute Christina Herzfeld and Oliver Zahner. A connectionist-geostatistical approach to automated image classification, applied to the analysis of crevasse patterns in surging ice. Computers & Geosciences, 27(5):499–512, 2001.

[5] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus), 2016.

[6] O. Eisen, W. D. Harrison, and C. F. Raymond. The surges of variegated glacier, alaska, u.s.a., and their connection to climate and mass balance. Journal of Glaciology, 47(158):351–358, 2001.

[7] M.W Gardner and S.R Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric Environment, 32(14):2627–2636, 1998.

[8] Vitor C. F. Gomes, Gilberto R. Queiroz, and Karine R. Ferreira. An overview of platforms for big earth observation data management and analysis. Remote Sensing, 12(8), 2020.

[9] Katie L. Grant, Chris R. Stokes, and Ian S. Evans. Identification and characteristics of surge-type glaciers on novaya zemlya, russian arctic. Journal of Glaciology, 55(194):960–972, 2009.

[10] Emmanuel Gringarten and Clayton V. Deutsch. Teacher's aide variogram interpretation and modeling. Mathematical Geology, 33:507–534, 2001.

[11] J. T. Harper, N. F. Humphrey, and W. T. Pfeffer. Crevasse patterns and the strain-rate tensor: a high-resolution comparison. Journal of Glaciology, 44(146):68–76, 1998.

[12] Robert Hecht-Nielsen. Iii.3 - theory of the backpropagation neural network**based on "non-indent" by robert hecht-nielsen, which appeared in proceedings of the international joint conference on neural networks 1, 593–611, june 1989. © 1989 ieee. In Harry Wechsler, editor, Neural Networks for Perception, pages 65–93. Academic Press, 1992.

[13] Matthew Hertz and Emery D. Berger. Quantifying the performance of garbage collection vs. explicit memory management. SIGPLAN Not., 40(10):313–326, 2005.

[14] Matthew Hertz and Emery D. Berger. Quantifying the performance of garbage collection vs. explicit memory management. page 313–326, New York, NY, USA, 2005. Association for Computing Machinery.

[15] Ute C. Herzfeld. Vario functions of higher order – definition and application to characterization of snow surface roughness. Computers & Geosciences, 28:641–660, 2002.

[16] Ute C. Herzfeld. Master of the obscure—automated geostatistical classification in presence of complex geophysical processes. Mathematical Geosciences, 40(5):587–618, 2008.

[17] Ute C. Herzfeld, Garry K.C. Clarke, Helmut Mayer, and Ralf Greve. Derivation of deformation characteristics in fast-moving glaciers. Computers & Geosciences, 30(3):291–302, 2004.

[18] Ute C. Herzfeld, Garry K.C. Clarke, Helmut Mayer, and Ralf Greve. Derivation of deformation characteristics in fast-moving glaciers. Computers & Geosciences, 30(3):291–302, 2004.

[19] Ute C. Herzfeld, Matthew Lawson, Thomas Trantow, and Thomas Nylen. Airborne validation of icesat-2 atlas data over crevassed surfaces and other complex glacial environments: Results from experiments of laser altimeter and kinematic gps data collection from a helicopter over a surging arctic glacier (negribreen, svalbard). Remote Sensing, 14(5), 2022.

[20] Ute C. Herzfeld, Brian McDonald, and Alexander Weltman. Bering glacier and bagley ice valley surge 2011: crevasse classification as an approach to map deformation stages and surge progression. Annals of Glaciology, 54(63):279–286, 2013.

[21] Ute C. Herzfeld, T. Trantow, M. Lawson, J. Hans, and G. Medley. Surface heights and crevasse morphologies of surging and fast-moving glaciers from icesat-2 laser altimeter data - application of the density-dimension algorithm (dda-ice) and evaluation using airborne altimeter and planet skysat data. Science of Remote Sensing, 3:100013, 2021.

[22] Ute C. Herzfeld, Thomas M. Trantow, David Harding, and Philip W. Dabney. Surface-height determination of crevassed glaciers—mathematical principles of an autoadaptive density-dimension algorithm and validation using icesat-2 simulator (simpl) data. IEEE Transactions on Geoscience and Remote Sensing, 55(4):1874–1896, 2017.

[23] Regine Hock and Peter Jansson. Modeling Glacier Hydrology, chapter 170. John Wiley & Sons, Ltd, 2006.

[24] Jason R. Janke. Using airborne lidar and usgs dem data for assessing rock glaciers and glaciers. Geomorphology, 195:118–130, 2013.

[25] Barclay Kamb, C. F. Raymond, W. D. Harrison, Hermann Engelhardt, K. A. Echelmeyer, N. Humphrey, M. M. Brugman, and T. Pfeffer. Glacier surge mechanism: 1982-1983 surge of variegated glacier, alaska. Science, 227(4686):469–479, 1985.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[27] MP Kodde, Norbert Pfeifer, BGH Gorte, Thomas Geist, and Bernhard Höfle. Automatic glacier surface analysis from airborne laser scanning. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 36(3/W52):221–226, 2007.

[28] Teng Li, Baogang Zhang, Wen Xiao, Xiao Cheng, Zhenhong Li, and Jian Zhao. Uav-based photogrammetry and lidar for the characterization of ice morphology evolution. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13:4188–4199, 2020.

[29] Lori Magruder, Thomas Neumann, and Nathan Kurtz. Icesat-2 early mission synopsis and observatory performance. Earth and Space Science, 8(5), 2021.

[30] Lori A. Magruder, Kelly M. Brunt, and Michael Alonzo. Early icesat-2 on-orbit geolocation validation using ground-based corner cube retro-reflectors. Remote Sensing, 12(21), 2020.

[31] Thorsten Markus, Tom Neumann, Anthony Martino, Waleed Abdalati, Kelly Brunt, Beata Csatho, Sinead Farrell, Helen Fricker, Alex Gardner, David Harding, Michael Jasinski, Ron Kwok, Lori Magruder, Dan Lubin, Scott Luthcke, James Morison, Ross Nelson, Amy Neuenschwander, Stephen Palm, Sorin Popescu, CK Shum, Bob E. Schutz, Benjamin Smith, Yuekui Yang, and Jay Zwally. The ice, cloud, and land elevation satellite-2 (icesat-2): Science requirements, concept, and implementation. Remote Sensing of Environment, 190:260–273, 2017.

[32] H. Mayer and U.C. Herzfeld. A structural segmentation, kinematic analysis and dynamic interpretation of Jakobshavns Isbræ, West Greenland. Zeitschrift für Gletscherkunde und Glazialgeologie, 37(2):107–124, 2001.

[33] R. Millan, J. Mouginot, A. Derkacheva, E. Rignot, P. Milillo, E. Ciraci, L. Dini, and A. Bjørk. Ongoing grounding line retreat and fracturing initiated at the petermann glacier ice shelf, greenland, after 2016. The Cryosphere, 16(7):3021–3031, 2022.

[34] Taewon Moon, Seojung Hong, Ha Young Choi, Dae Ho Jung, Se Hong Chang, and Jung Eek Son. Interpolation of greenhouse environment data using multilayer perceptron. Computers and Electronics in Agriculture, 166:105023, 2019.

[35] Matthias Müller, Lars Bernard, and Johannes Brauner. Moving code in spatial data infrastructures – web service based deployment of geoprocessing algorithms. Transactions in GIS, 14(s1):101–118, 2010.

[36] Thomas A. Neumann, Anthony J. Martino, Thorsten Markus, Sungkoo Bae, Megan R. Bock, Anita C. Brenner, Kelly M. Brunt, John Cavanaugh, Stanley T. Fernandes, David W. Hancock, Kaitlin Harbeck, Jeffrey Lee, Nathan T. Kurtz, Philip J. Luers, Scott B. Luthcke, Lori Magruder, Teresa A. Pennington, Luis Ramos-Izquierdo, Timothy Rebold, Jonah Skoog, and Taylor C. Thomas. The ice, cloud, and land elevation satellite – 2 mission: A global geolocated photon product derived from the advanced topographic laser altimeter system. Remote Sensing of Environment, 233:111325, 2019.

[37] J. F. Nye. The mechanics of glacier flow. Journal of Glaciology, 2(12):82–93, 1952.

[38] Charles F. Raymond. How do glaciers surge? a review. Journal of Geophysical Research: Solid Earth, 92(B9):9121–9134, 1987.

[39] Brice R. Rea and David J. A. Evans. An assessment of surge-induced crevassing and the formation of crevasse squeeze ridges. Journal of Geophysical Research: Earth Surface, 116(F4), 2011.

[40] W. G. Rees and N. S. Arnold. Mass balance and dynamics of a valley glacier measured by high-resolution lidar. Polar Record, 43(4):311–319, 2007.

[41] Nicol N. Schraudolph. Centering Neural Network Gradient Factors, pages 207–226. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[42] C. E. Shannon. Prediction and entropy of printed english. Bell System Technical Journal, 30(1):50–64, 1951.

[43] Liam S Taylor, Duncan J Quincey, Mark W Smith, Celia A Baumhoer, Malcolm McMillan, and Damien T Mansell. Remote sensing of the mountain cryosphere: Current capabilities and future opportunities for research. Progress in Physical Geography: Earth and Environment, 45(6):931–964, 2021.

[44] Marco Tedesco. Remote sensing and the cryosphere, chapter 1, pages 1–16. John Wiley & Sons, Ltd, 2015.

[45] Thomas Trantow and Ute C. Herzfeld. Crevasses as indicators of surge dynamics in the bering bagley glacier system, alaska: Numerical experiments and comparison to image data analysis. Journal of Geophysical Research: Earth Surface, 123(8):1615–1637.

[46] Mathias Vuille, Bernard Francou, Patrick Wagnon, Irmgard Juen, Georg Kaser, Bryan G. Mark, and Raymond S. Bradley. Climate change and tropical andean glaciers: Past, present and future. Earth-Science Reviews, 89(3):79–96, 2008.

[47] Benjamin Walker and Laura Ray. Multi-class crevasse detection using ground penetrating radar and feature-based machine learning. In IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, pages 3578–3581, 2019.

[48] Cheng Wang, Xiaoxiao Zhu, Sheng Nie, Xiaohuan Xi, Dong Li, Wenwu Zheng, and Shichao Chen. Ground elevation accuracy verification of icesat-2 data: a case study in alaska, usa. Opt. Express, 27(26):38168–38179, 2019.

[49] Rebecca M. Williams, Laura E. Ray, James H. Lever, and Amy M. Burzynski. Crevasse detection in ice sheets using ground penetrating radar and machine learning. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7(12):4836–4848, 2014.

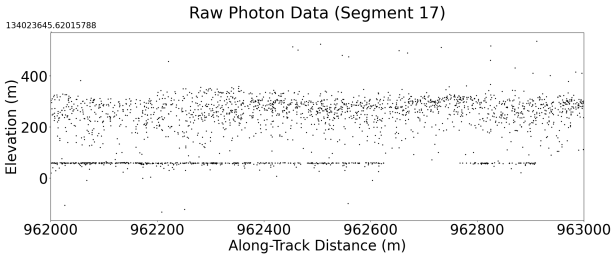
[50] Cheng Xiang, S.Q. Ding, and Tong Heng Lee. Geometrical interpretation and architecture selection of mlp. IEEE Transactions on Neural Networks, 16(1):84–96, 2005.

# Appendix A

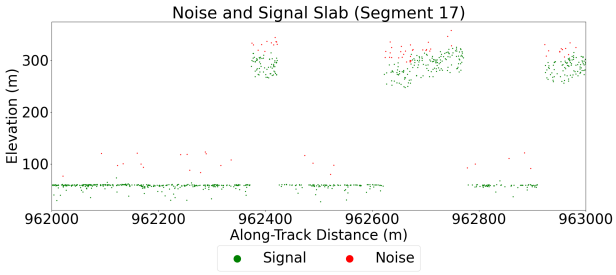
## Supplemental Figures

### A.1 Cloud Obstructed Data

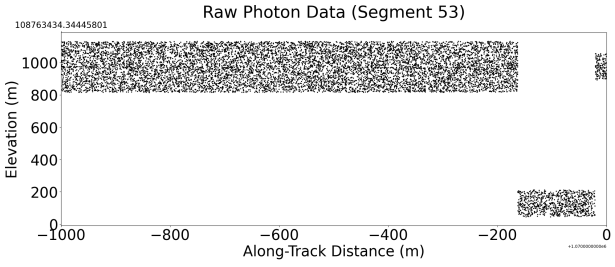
Figure A.1: Geolocated photon point-cloud data from ATL03 (**(a)** and **(c)**), and the corresponding DDA-ice signal/noise slabs (**(b)** and **(d)**), where the sensor has been obstructed by cloud cover and/or high background noise scattering. All data sourced over Greenland. ICESat-2 granule info: for **(a)** and **(b)**: ATL03_20220401044511_01391503_005_01.h5 beam: gt1l (strong), for **(c)** and **(d)**: ATL03_20210612200124_12221103_005_01.h5 beam: gt2l (strong).



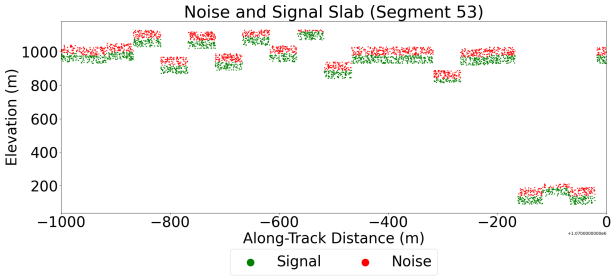

(a) Raw Photons




(b) Signal / Noise Slabs




(c) Raw Photons




(d) Signal / Noise Slabs

## A.2 Training Loss Divergence Examples

Figure A.2: Loss curves for a model with a significantly smaller batch size compared to the selected DDA-MLP model causing divergence between the training and validation loss curves after only 394 training epochs. Model hyperparameters: hidden layers = [3, 4, 2], learning rate = $1 \times 10^{-6}$, batch size = 8, num epochs = 2000.

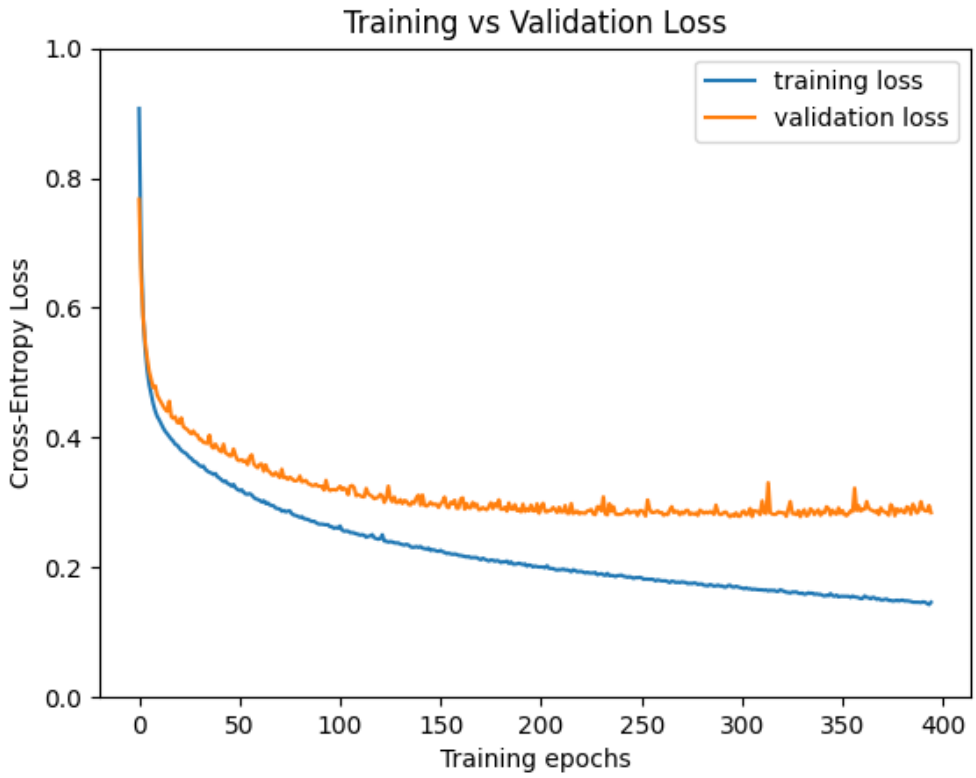


Figure A.3: Loss curves for a significantly more complex model compared to the selected DDA-MLP model causing divergence between the training and validation loss curves. Model hyperparameters: hidden layers = [30, 54, 32], learning rate = $1 \times 10^{-6}$, batch size = 32, num epochs = 2000.

Figure A.4: Loss curves for a model with a higher learning rate compared to the selected DDA-MLP model causing divergence between the training and validation loss curves. Model hyperparameters: hidden layers = [3, 4, 2], learning rate = $5 \times 10^{-5}$, batch size = 32, num epochs = 2000.