Airborne Snow Radar Data Simulation With Deep Learning and Physics-Driven Methods

Masoud Yari, Oluwanisola Ibikunle, Debvrat Varshney, Tashnim Chowdhury, Argho Sarkar, John Paden, *Senior Member, IEEE*, Jilu Li, *Senior Member, IEEE*, and Maryam Rahnemoonfar

***Abstract*—Monitoring properties of ice sheets in polar regions is one of the main challenges in glaciology. There is a large amount of heterogeneous radar data from the polar regions that have been gathered through expensive missions. However, retrieving meaningful information from this large volume of data is still a great challenge. With the advancement of machine learning techniques in recent years, many scientists are eager to take advantage of these algorithms and techniques to explore and mine Arctic and Antarctic data. These advancements, however, have happened mainly in the area of supervised learning where the models are data hungry and require large amounts of annotated data. Generating simulated data can be an effective and inexpensive approach to provide large labeled datasets for training machine learning models. In this work, we explore two approaches to simulate arctic snow radar echogram images, namely a radar scattering physics based approach combined with some statistical measures and a purely data-driven approach based on a conditional generative adversarial network. Using several image comparison metrics, we analyze the utility of both methods for the purpose of simulating echograms. Our results show that the physics simulator generates images with good structural similarities, while the purely data-driven approach achieves better textural similarities for simulated image. Finally, we also show that by augmenting our real dataset by the simulated echograms, we can improve our deep learning model for tracking internal layers of snow.**

***Index Terms*—Generative adversarial networks (GANs), remote sensing, simulation, snow radar.**

I. Introduction

MELTING polar ice sheets are contributing to an increase in global sea-level with a rapid growth rate over the past several decades [1]. Sea-level rise may lead to coastal flooding that affects millions of livelihoods around the world. In this regard, precise estimation of annual snow layer thickness over time is very important for understanding snow precipitation on the ice sheets. It is particularly needed to model the dynamics of ice sheets and project their future mass balance. One way of calculating the snow thickness is through radar surveys, which reveal annual accumulation layers hidden beneath the ice surface. These annual accumulation layers are caused by changes in the snow structure or density, which results in a reflection that is sensed by the radar [2], [3]. The shallow snow layers have been dated using ice cores and the dominant layers shown to occur on an annual basis. Combined with appropriate snow density models or actual on-site measured density profiles, annual snow accumulation rates can be inferred from the layers in radio echo sounding data. Many thousands of line-km are covered in the radar surveys across Antarctica and Greenland. The massive volume of data products produced requires automatic techniques, which can process and extract the relevant snow layer thickness information.

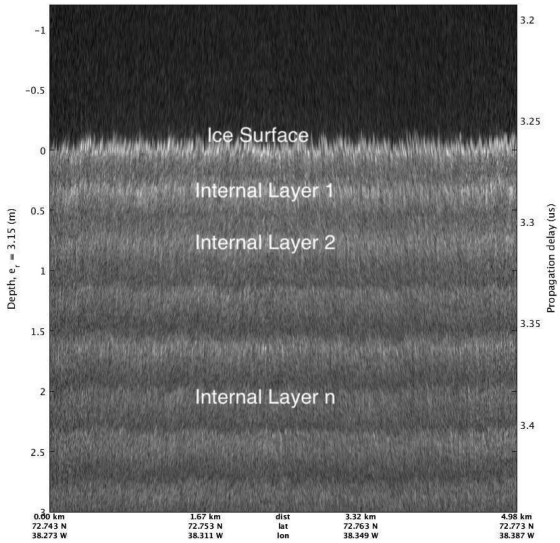

Fig. 1. Snow radar echogram image.

As part of the National Aeronautics and Space Administration (NASA) Operation IceBridge campaign, the Center for Remote Sensing of Ice Sheets (CReSIS) at the University of Kansas flew the Snow Radar [4] over Greenland to measure shallow layers of snow with fine vertical resolution. The collected data have been processed to produce radar echograms, which reveal the annual radiostratigraphy of snow as shown in Fig. 1.

To extract information needed by the science community such as melt rate, layer geometry, and annual snow accumulation, from the radar data, there is a need for accurate tracking of annual layers in radar echograms. However, manual tracking of multiple layers in thousands of echograms generated from

Manuscript received May 31, 2021; revised August 13, 2021, September 27, 2021, and October 26, 2021; accepted November 2, 2021. Date of publication November 19, 2021; date of current version December 6, 2021. This work was supported by NSF BIGDATA Awards (IIS-1838230, IIS-1947584), NSF HDR Institute Award (OAC-2118285), IBM, and Amazon. *(Corresponding author: Maryam Rahnemoonfar.)*

Masoud Yari, Debvrat Varshney, Tashnim Chowdhury, Argho Sarkar, and Maryam Rahnemoonfar are with the Computer Vision and Remote Sensing Laboratory (Bina Lab), University of Maryland, Baltimore County, MD 21250-0001 USA (e-mail: yari@umbc.edu; debvrav1@umbc.edu; tchowdh1@umbc.edu; asarkar2@umbc.edu; Maryam@umbc.edu).

Oluwanisola Ibikunle, John Paden, and Jilu Li are with the University of Kansas, Lawrence, KS 66045 USA (e-mail: ibikunle.oluwanisola@ku.edu; paden@ku.edu; jiluli@ku.edu).

Digital Object Identifier 10.1109/JSTARS.2021.3126547

multiple flights and campaigns is both onerous and ineffective, with the better alternative being an automated tracking approach using machine learning or deep learning algorithms.

Supervised learning methods and, in particular, recent advances in convolutional neural networks (CNNs) [5] have shown promising results in processing large datasets and understanding images. These algorithms are trained to learn complex features from the input images and extract information from them. With increase in dataset size, the networks learn better and perform very well on tasks such as image classification [6], [7], object detection [8], [9], semantic segmentation [10], [11], and image denoising [12], [13].

Like many other scientific fields, the lack of annotated data, or the difficulty in obtaining them, is the main barrier for implementing supervised learning methods. These challenges lead to an insufficient amount of correctly labeled data for deep learning models, which hampers the supervised learning process [14].

Some semisupervised methods have been used to track the snow layers; however, many of these tracked layers have either missing or incomplete labels, which consequently creates a great amount of uncertainty in the learning process. Therefore, it is highly desirable to create a large set of synthetic data that can be used as supplementary training data.

Traditionally, one way to create synthetic data is to understand and simulate the physical processes or the radar scattering; in this case, the processes responsible for snow accumulation layer creation. The works in [3] and [15]–[18] and a host of other works detail the underlying snow accumulation, firn compaction, and densification processes, which give rise to the layer reflections in the echograms. Instead of directly modeling the geophysical properties of the snow, we use a sizable amount of manually tracked snow radar data collected over Greenland to extract statistical measures from the data in order to parameterize simulated random scattering processes to closely approximate the actual underlying physical processes. These simulated processes can then be used to produce enough synthetic annotated data to augment the training data needed for a deep learning algorithm. We hereafter will refer to this method as the physics-driven (PHY) method or physics simulator. However, models that are trained based on these types of annotated data, using some kind of supervised learning approach, tend to approximate the maximum likelihood function, which basically amounts to estimation of the same physical function via a data-driven approach. Purely data driven approaches, on the other hand, have demonstrated great success in generating artificial imagery in recent years. In particular, generative adversarial networks (GANs) [19] are successfully used in simulating near-realistic fake images. This approach can be helpful in the scientific domain for generating synthetic data, so that they can be used for various types of supervised learning tasks. For example, Rahnemoonfar generated synthetic radar depth sounder images [20] and snow radar images [21] using GANs. However, the GANs only reproduce the probabilistic distribution of the original data, and not the physical or biological signals which are inherent in it [20], [22], [23].

The prime goal of this work is to propose data simulation methods that generate near-real echograms, and to use them to enhance the performance of internal layer tracking models by augmenting the training dataset. Our work in this article is organized as follows. We first discuss related works in Section II. We then propose two methods, namely a PHY method (see Section III) and a purely data-driven method (see Section IV).

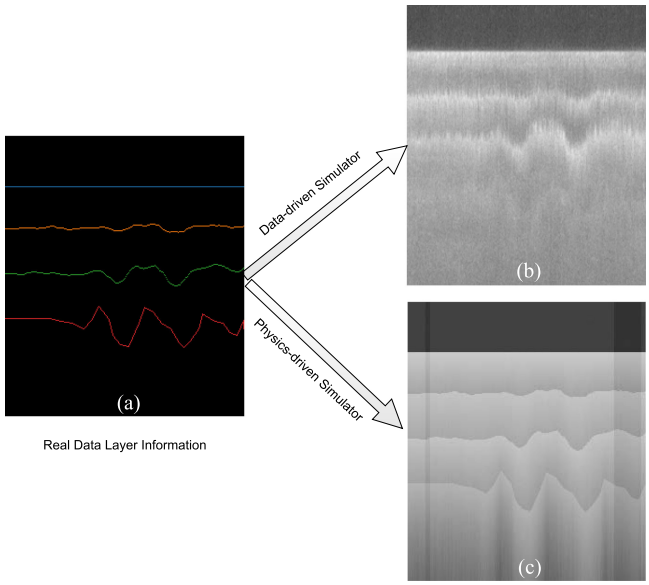

Fig. 2. First experiment involves using real radar data to generate simulated data using a data-driven approach and a PHY method. We use the new simulated data to improve our layer tracking model. (a) Sample layer information obtained from a real radar echogram; (b) output of the data-driven simulator; (c) output of the PHY simulator.

In Section V, we explain our benchmark datset and compare the performance of the two methods on our benchmark real dataset. In our experiment, we generate synthetic data with both data-driven and PHY approaches using the layer data in our benchmark dataset (see Fig. 2). We compare the results against the real echograms using some rigorous metrics, which we describe in Section V-C.

In Section VI, we use the PHY method to generate synthetic layer information, we then use both data-driven and PHY methods to generate the corresponding synthetic images (see Fig. 3). In the same section, we also report our quantitative results after training our deep learning model on our benchmark dataset augmented by simulated data. We also compare the quantitative results with previous results in [24] and [25]. Finally, Section VIII concludes this article.

II. Related Works

A. Image Simulation Through PHY Methods

The need for glacialogically constrained simulators has been reported in several past works. A few of these simulators have been developed albeit for different purposes than internal layer tracking as most of the current simulators focus on lower frequency sounding and not on the snow radar's microwave frequencies.

Leuschen *et al.* [26] developed a simulator for future exploratory missions to Mars for identifying subsurface interfaces marked by dielectric contrast, which can be up to a few kilometers beneath the planet's surface. Their model included a set of standard crustal models for chosen geological regions using a frequency domain algorithm that modeled physical and geological conditions such as frequency dispersion, dielectric layering, and ohmic loss.

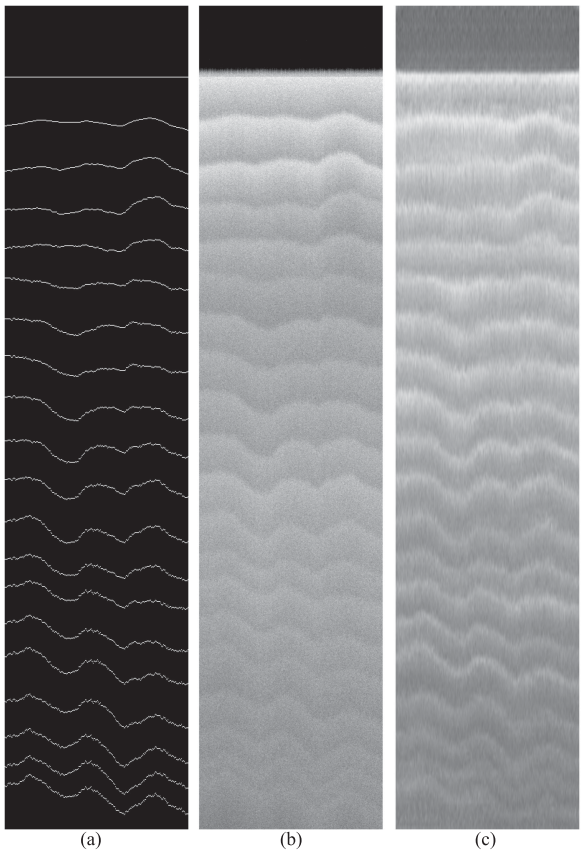

Fig. 3. (a) Sample simulated layer information; (b) output of the PHY simulator; (c) output of the data-driven simulator.

Gerekos *et al.* [27] created a simulator model that expanded existing models from surface-only or two-layers to an arbitrary number of subsurface geological layers. Using Snell's law and ray tracing, their simulator model used a coherent, facet-based approach to linearly sum the phase contribution of the reflecting or refracting rays from each facet at the receiving antenna using Huygen's principle. However, their work is more suited for lower frequency systems that can view the layers as smooth facets as opposed to being more volumetric in nature as they are at the microwave frequencies used by the snow radar.

The work in [28] developed glacialogically constrained electromagnetic models for the interaction of radar waves in firn and tested these against *in situ* data from a snow firn core. Their physical model for snow firn focused on off-nadir surface clutter migration and volume scattering from air-filled pores in the inhomogeneous material of the firn. They, however, used a quasispecular reflection model for the interfaces between layers of different density because their model was for high frequency (HF) and very high frequency (VHF) radars well below the frequencies used by the Snow Radar.

These works and similar papers focus on designing radar simulators for spaceborne and orbit sounding radars that typically operate in the HF and VHF range, while the aim of our proposed physics method is to create a simulator that represents airborne snow radar data operating in the 2–8 GHz frequency band [4], [29]. It should be noted here that our proposed physics method is based on a convolutional model from satellite radar altimetry [37]. Although full wave electromagnetic simulation methods such as the finite-difference time-domain (FDTD) technique can more accurately reproduce the electromagnetic scattering physics, they are computationally very expensive and require extensive knowledge of the permittivity and conductivity fields.

B. Image Simulation Through GANs

GANs [19], proposed by Goodfellow *et al.*, are deep learning based generative models. A GAN model consists of two submodels, a generator model and a discriminator model. The generator outputs synthetic images and the discriminator classifies the images as real (actual data) or fake (synthetic images from the generator). The generator model is dependent on the discriminator while discriminator updates are independent. Among different types of GAN architectures, conditional GAN (cGAN) has gained popularity. While basic GANs generate images from random distributions unconditionally, cGANs learn the desired mappings under conditional settings. For example, an unconditional GAN generates random digits based on the training while a cGAN generates digits in a range specified by the user. cGANs were first introduced to generate more meaningful images by resembling the data distribution for a given condition. Many noticeable research works have been conducted based on conditional constraints in adversarial nets. In [30], Mirza *et al.* have introduced a cGAN on the MNIST dataset for synthetic digit image generation. Denton *et al.* [31] have implemented the Laplacian Pyramid framework on top of a cGAN for natural image generation. The models that condition on images have been improved for many applications such as image generation from a map generated from a structure-GAN [32], predicting future frames [33], [34], and image generation from various scene attributes [35]. Isola *et al.* [36] investigate conditional adversarial networks for image-to-image translation problems.

For radar images, there have been several implementations of GANs for simulating synthetic images. Authors in [37] implement a GAN architecture named image despeckling generative adversarial network (ID-GAN) to generate the speckle noise in SAR images where speckle noise is a multiplicative noise present in SAR imagery. Ao *et al.* present a dialectical GAN in [38], which generates high-quality SAR images in order to improve the quality of SAR images and reduce the cost of their generation. In [20], Rahnemoonfar *et al.* use a cycle-consistent adversarial network as a data augmentation technique to generate synthetic radar images, part of which are used during training of a neural network. A CycleGAN [39] based architecture is implemented in [21] to generate synthetic snow radar images. Cycle-GANs have shown impressive results for radar image generation, but capturing the texture and contrast of real images more accurately needs further investigation.

In general, the deep learning approach cannot simulate all the radar characteristics (e.g., Doppler). Thus, the generated images are potentially valid only for a specified set of radar system parameters matching those in the GAN training set.

III. Physics Simulator

To simulate echogram images with complete labels necessary for data hungry deep learning algorithms, we sought to identify and reproduce prominent physical processes already reported in the literature that govern the layering and backscatter of the snow radar. Our approach is data driven in the sense that we simulate the underlying random processes and parameterize

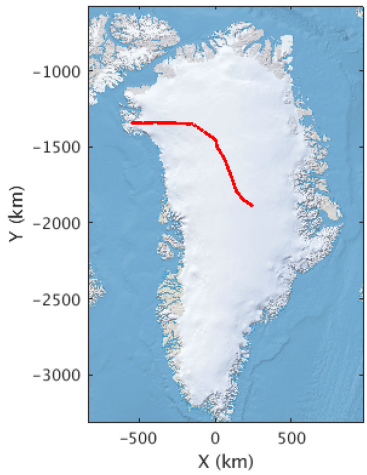

Fig. 4. Image of Greenland showing CReSIS flight line used to estimate image statistics.

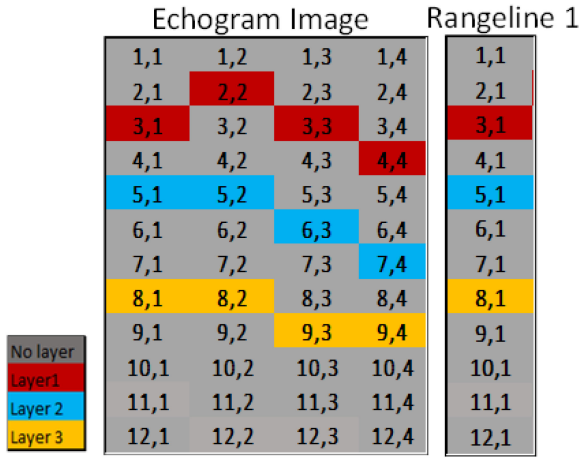

Fig. 5. Echogram image with the row and column index for each pixel indicated. Three layers are highlighted and the first range line indices are shown on the right.

TABLE I
KEY PARAMETERS OF THE SNOW RADAR USED DURING DATA COLLECTION

	Snow Radar Parameters
Bandwidth	2-8 GHz
Pulse duration	250 μ s
PRF	2 kHz
Transmit power	100 mW
Intermediate frequency range	62.5 -125 MHz
Sampling frequency	125 MHz
Range resolution	$\sim$ 4 cm
Along-track footprint	14.5 m

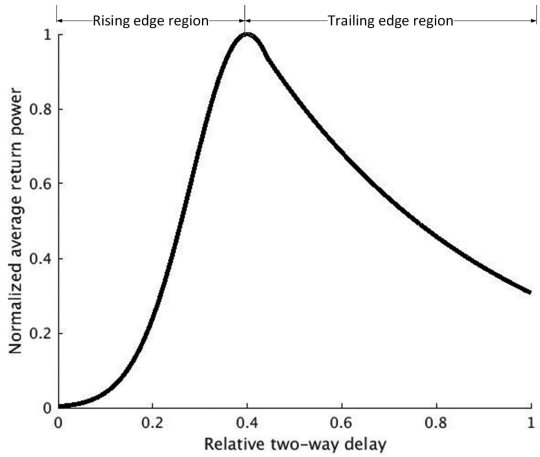

Fig. 6. Illustration of the normalized average or expected backscatter power from a single snow layer showing the fast rising edge and slow decay for the falling edge.

them based on estimated statistics from snow radar backscatter data collected during one of our flights over Greenland, as shown in Fig. 4. Table I lists some of the parameters of the snow radar used. This dataset (hereafter referred to as the "sample dataset") covers a 1200 km long flight line from central to north-west Greenland providing varying snow accumulation conditions. These data (after stacking and other postprocessing) represent 200 000 rangelines. It should be noted that most of this data is from the dry snow zone of Greenland where minimal snow melting occurs even in the summer and there are many detectable internal layers in the snow radar imagery.

The echogram images are data matrices where each pixel represents the nonnegative scattering power received from the targets in that pixel offset and scaled to the 8-bit range 0 to 255. Fig. 5 shows a depiction of an echogram data matrix with discretized bins (pixels) in both vertical (fast-time) and horizontal (slow-time) dimensions. The figure depicts an echogram with four rangelines each having three layers. Each pixel in the figure is denoted by its row (range bin) and column (range line) indices. A range line is a column in the echogram and usually contains scattering from the surface (red pixels in figure) and one or more internal layers (blue and yellow pixels in figure). A pixel corresponding to a snow layer is usually distinct due to the increased backscatter or higher image value caused by dielectric contrasts in the snow at that pixel location. We define layers to be single valued so that in each column of the image the layer takes on one and only one value. Snow layers tend to be nearly flat with smooth variations in the vertical or row dimension.

Using manually tracked snow layers from the sample dataset, we set out to reproduce the image's random processes. The layer scattering responses generally follow the shape predicted by the convolution models used in surface altimetry [40], [41]. The shape starts with a fast rising edge followed by a slower, exponential like, decay with the tracked layer centered on the peak. This behavior is shown in Fig. 6. Moore and Williams [41] show that the expectation of the power detected waveform for the surface can be modeled as the convolution of several constitutive elements such as the height distribution of the layer and the pulse limited footprint and expected backscatter. We assume the Born approximation so that the interaction or multipath between layers can be ignored. We model the set of layers as the superposition of each layer independently generated and then incoherently summed. This is shown in Fig. 7.

Following the convolution model, we simulated the response by convolving a Gaussian waveform, which approximates the height distribution of the layer, and an exponential decay waveform, which approximates the combined pulse limited area and layer backscatter roll-off. The peak of the Gaussian is aligned with the tracked layer location. The combined return for a range

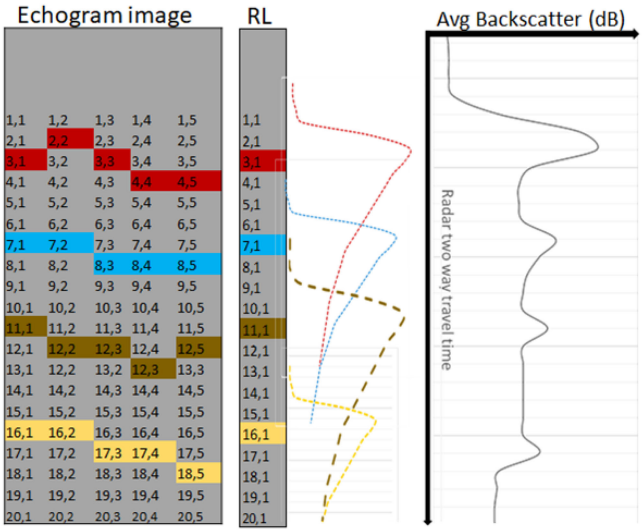

Fig. 7. Illustration of the average or expected backscatter for a range line as a linear superposition of each of the individual layers.

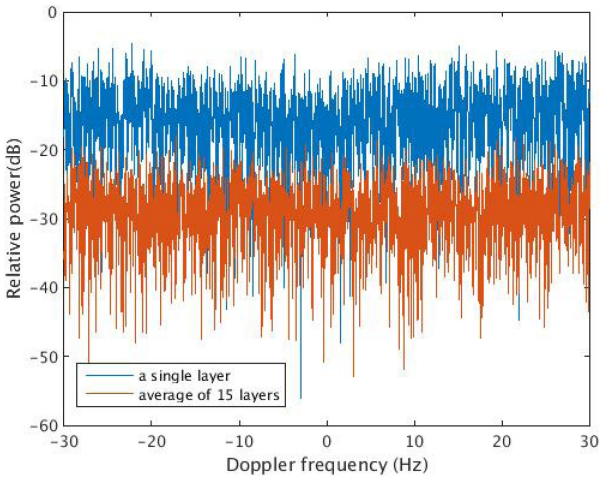

Fig. 8. Doppler spectra for a snow layer and the average of multiple layers. The Doppler spectrum is equivalent to the along-track wavenumber spectrum.

line is the linear superposition of these convolved waveforms—one for each layer in that range line.

Snow radar scattering tends to be incoherent because scatterers that form a layer vary randomly throughout the snow volume that constitute the layer. Therefore, range lines typically have minimal to no phase correlation with those in neighboring range lines. This incoherent backscattering assumption, which is the basis of the above convolution model, is supported by the Doppler spectra along snow layers. In general, the spectrum is broad without distinct coherent peaks. An example of the spectra from the first layer and the average of the first 15 layers below the surface is shown in Fig. 8 over a distance of 5 km after the aircraft elevation variations have been compensated for. The histogram of the snow radar data in the sample dataset, which includes along-track incoherent averaging, shows that the distribution of the peak power along a layer fits a Chi-square distribution. Using the scaled superposition waveform as the mean power, a Gaussian random process was created to simulate the expected power returns from each range line. As described later in this section, we then power detect and incoherently averaged multiple range lines to create the Chi-square distribution.

To parameterize the layer power generation, the mean power for layer l, denoted, m_l, is found by taking the average power of the bin that is manually tracked across all range lines where the layer is defined. The mean power represents the backscatter received by the radar for each layer and, therefore, encapsulates backscatter cross section, attenuation, and other affects. The width of the layers was also found. We estimated the width of a layer by calculating the range bins it took for the peak power to decline to $1/e$ of the layer peak return for all well-defined and tracked layers in the sample dataset. Based on the resulting histogram of the estimated layer width, we approximate the width of the layer peaks, d_l, by a uniform distribution between 10 and 15 range bins or rows.

The estimated along track mean peak power, m_l, for each layer is used to scale the convolved exponential and Gaussian waveform and the resulting range line signal power is given by

$$P_s(x) = \sum_{l=1}^{L} m_l \exp \frac{-(x-\mu_l)^2}{2d_l} \circledast u(x) \exp\left(-\alpha_l x\right) \quad (1)$$

where P_s is the expected backscatter power waveform for each range line; x is the fast time pixel index; u is the unit step function; L is the number of layers in the range line; m_l is the mean peak power for layer l, μ_l is the location or row of layer l and its generation will be discussed in Section V, d_l is the width of layer l pulled from a uniform distribution from 10 to 15 rows, and α_l is the exponential decay rate of the layer, respectively. The values of m_l, d_l, and α_l are estimated from the sample data.

To estimate the decay rate of each layer, all range lines in the sample data are grouped into K groups of 20 consecutive range lines each. Each group was then incoherently averaged in the along-track dimension to produce a single filtered range line per group. Note that the data are already incoherently averaged and decimated by five during the process to generate the echograms so the total number of incoherent averages is 100. This ensemble of K filtered range lines is then used to find the backscatter peak power $P_{\text{peak,lk}}$ and the minimum power between this and the next peak $P_{\text{min,lk}}$ for each filtered range line $k \in 1, \ldots, K$ and each layer $l \in 1, \ldots, L$. $P_{\text{peak,lk}}$ generally corresponds to the location of the layer since the tracked layer follows the peak power. Using the range bin distance between the peak and the minimum for each layer and filtered range line, b_{lk}, we compute the estimated decay rate for each layer as follows:

$$\alpha_l = 1/K \sum_{k=1}^{K} (-1/b_{\text{lk}}) * \log(P_{\text{min,lk}}/P_{\text{peak,lk}}). \quad (2)$$

The noise power, P_n, is constant and is estimated from the sample dataset by estimating the power of the received signal before the surface return under the assumption that there are no targets above the surface so that only thermal noise is present.

The complex signal and additive noise are both pulled from additive white complex circular Gaussian noise, which is then scaled by the estimated signal and noise power described earlier. The distributions before scaling have zero mean and unit variance so that

$$s(x) \sim \mathcal{N}(0,1) \in \mathbb{C}, n(x) \sim \mathcal{N}(0,1) \in \mathbb{C}. \quad (3)$$

After the complex weighted range lines are generated, they are power detected. The power detected range line with signal and noise is

$$P(x) = |s(x)\sqrt{P_s(x)} + n(x)\sqrt{P_n}|^2. \tag{4}$$

The distribution of the power detected signal follows an exponential distribution. The final step in the simulator is to incoherently average $M = 100$ range lines together followed by decimation in along-track by M. This results in a Chi-squared distribution with $2M$ degrees of freedom. This is done in the data processing to reduce signal fading which helps produce smoother and better delineated layering in the images.

IV. Simulating Data With cGAN

There exists a vast literature about generating near real imagery in the context of deep neural networks. In our work, we consider the image-to-image translation approach introduced in [36]. The main motivation for conditional cGAN is to generate high-quality realistic images upon certain conditions.

cGANs learn a mapping from observed image x and random noise vector z, to output image y. The objective function of the cGAN then can be defined as follows:

$$\mathcal{L}_{\text{cGAN}} = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z)))] \tag{5}$$

where G and D are generator and discriminator, respectively. In comparison with the objective function of the unconditional GAN

$$\mathcal{L}_{\text{GAN}} = \mathbb{E}_{y}[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(z)))] \tag{6}$$

we can see that the discriminator does not observe x.

In (5), G is supposed to minimize the objective function against an adversarial D that tries to maximize it. The generator is supposed to control the discriminator and not to deviate "too far" from the ground truth. We can achieve that by adding a Lagrange multiplier using the L_2 or L_1 distance. Here, we use L_1 since it encourages more pronounced edges. Therefore, our final objective can be expressed as

$$G^* = \arg\min_G \max_D \mathcal{L}_{\text{cGAN}}(G,D) + \lambda \mathcal{L}_{L1}(G) \tag{7}$$

where

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x,z)||_1]. \tag{8}$$

In this work, for the generator we implemented a U-Net [42] architecture, which allows skip connections to be added between the layers of similar size in the encoder and decoder. A typical encoder–decoder based generator progressively downsamples input images until the bottleneck layer. After the bottleneck layer, the process is reversed and the features maps are upsampled to the original size. In the image translation problem, a lot of information is shared between the input and output. In this scenario, a proper sharing of this information across the network is crucial. The skip connections in the U-Net allow the later layers in the decoder to use the features learned in earlier layers in the encoder. On the other hand, we used a convolutional PatchGAN as the discriminator, which can model HF structure. Compared to a regular GAN discriminator, the main advantage of PatchGAN is that while a regular discriminator decides whether an image is fake or real based on a single scalar value, PatchGAN penalizes the structure and makes the decision at the scale of patches.

V. Experimental Results

One of our tasks in this article is to compare the performance of the PHY and cGAN simulation methods. In this section, we will compare the two methods and describe our qualitative as well as quantitative comparison results. The benchmark datasets, which we use for comparison and our other experiments, are described in Section V-A. In the following, in Section V-B, we will explain our simulated data. In the last Section V-C, we will introduce the evaluation metrics we used for comparing numerical similarity between the simulated and real radar images, and will present our comparison results.

The ultimate goal of developing simulation methods is to augment our labeled dataset by generating new real-like images and their corresponding internal layers. Therefore, we plan red to generate new simulated data, and later use them to enhance the performance of our supervised learning model. We discuss this in later sections of this article; however, in the Section V-A, we will describe the dataset we will use for this purpose as well.

A. Dataset

Our dataset consists of 2620 radar echograms that were captured in the year 2012 and are publicly available on the CReSIS website. There are some semiautomatic techniques proposed recently, such as [3], [43], which are able to track internal layers of ice for a given noisy radar sounder image; however, they fail to track all the layers and in many cases can only partially track the layers. Fig. 9(a) is a sample of our radar images. Fig. 9(b) shows the corresponding annotated data generated by a semiautomatic method, where some internal layers are not tracked at all, and some are partially tracked. As discussed in [24], we have used the output labels from semiautomatic techniques as a starting point for training a deep learning model.

For our experiments, we divide the data to training and test sets. What we will refer to from now on as the *benchmark training dataset* or simply the *training set* consists of 2360 pairs of radar images and corresponding semiautomatic annotations.

The rest of 260 pairs of radar images and the corresponding annotations makes our *benchmark test dataset*, which we will refer to as the *test set*. In both training and test sets, the image sizes vary, and the number of internal layers in each image can be different.

We use the test set for quantitative comparison in this section. However, we have manually completed the partially annotated internal layers in all of our test data images. Fig. 9(c) presents an example of a fully annotated image. The number of internal layers in about 57% of our manually corrected test data is less than 6 layers. The highest number of layers is 34 across all 260 images.

For more information about the data, we refer the reader to CReSIS website[1] and [4], [29], [44]–[46]. The train and test datasets are discussed in [24] as well.

B. Data Simulation

To compare the two methods used to simulate radar data, we generate two sets of simulated data using the annotations of the test dataset. This will allow us to compare the fake data, which the simulators generate, with the real data. To train the cGAN model, we use our training set of 2360 images. We have kept the

[1] https://data.cresis.ku.edu/data/misc/koenig_TC2016_ML

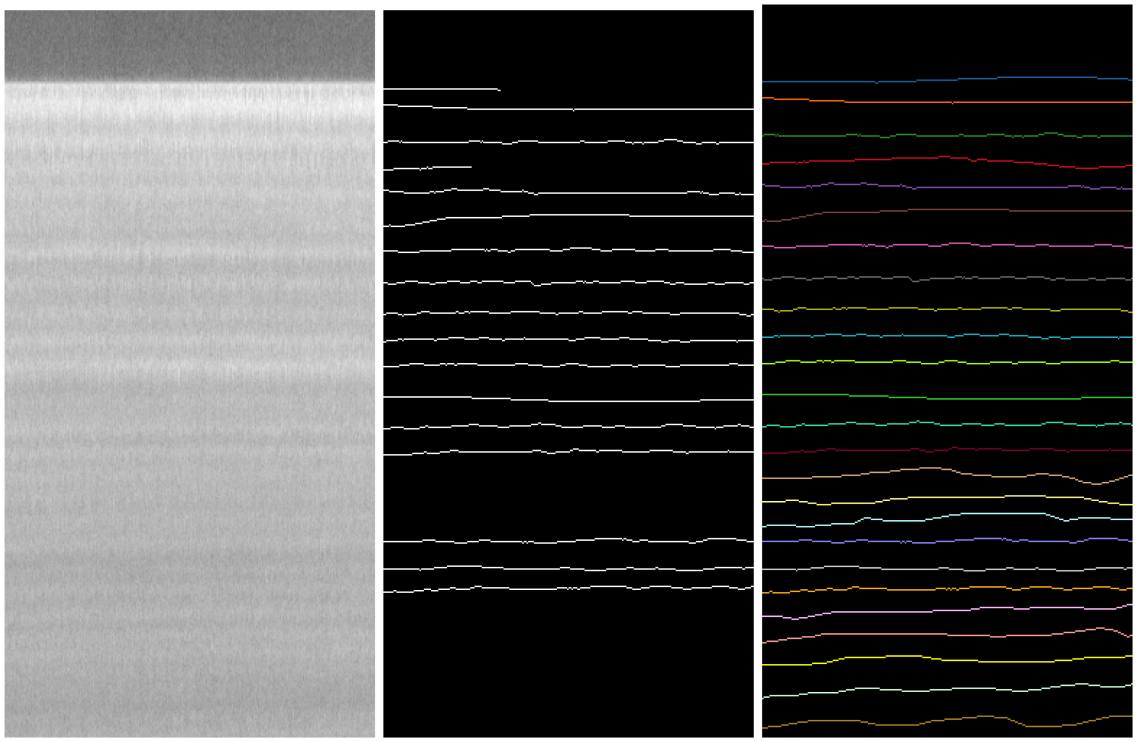

Fig. 9. Sample test data. (a) Radar echogram. (b) Semiautomatic annotation. (c) Manually corrected annotation.

same standard hyperparameters for training as described in [36]. After training, we use the trained model to generate 260 fake images from the binary mask of labels of our test set. For the PHY simulation, we obtain another set of 260 simulated images by applying the physics simulator on the labels of our test set. In Section V-C, we will use these two datasets for comparing the two methods.

For the purpose of augmenting our real dataset and training a layer tracker model (see Section VI), we also created a second set of "entirely new" simulated radar sounder images. This is done by first simulating "new" layer annotations similar to real radar layer geometries as seen in the sample dataset (please refer to Section III) and then applying the physics and cGAN simulators separately to generate new sets of radar images. Given the spatial span of the sample dataset, its layer geometry is representative of how snow accumulation varies over space and with depth. Therefore, we sought to model the layer thickness process for each layer using the sample dataset layer statistics.

Using the manual annotations available for the sample dataset, we estimated each layer's thickness (snow accumulation) over space by computing the difference between successive layers across all the rangelines. The histogram of the data shows that the probability density function of the accumulation for each layer is approximately a Gaussian process with slowly varying along-track mean accumulation. This suggests that, although there are local variations in each layer thickness from rangeline to rangeline, there is also a slowly varying trend in the average layer thickness over space. This varying along track mean thickness corroborates the fact that there are different accumulation rates over Greenland and that snow accumulation can vary a lot between different locations. Therefore, we regrouped the dataset using the thickness of the first layer into four groups of accumulation zones: shallow, medium, high, and very high accumulation zones so that each simulated image created from the synthetic layers belongs to one of these accumulation zones. Another important process is the correlation between the layer thickness of different layers at a particular location. This is seen qualitatively in an echogram image such that the layers often share a similar trend in their layer geometry and this can be attributed to the weather and topography of the imaged location, which tend to be consistent over time as each layer is deposited at a site. Therefore, to create layers with similar geometry to that in real data, these processes need to be incorporated in the model.

To model the HF local variations that exist from rangeline to rangeline, the power spectral density of each layer thickness after subtracting the along-track mean in each zone was computed. Similarly, to capture the slowly varying along-track accumulation mean that describes how the mean thickness varies across space, we computed the mean thickness and variance for each zone. The computed mean thickness of each zone represents the mean accumulation of the layers. Lastly, to partly model the similar trend that exists in the thickness of the first layer and subsequent layers and also the geometry of layers in an echogram, we normalized the layer thicknesses of all the layers relative to the thickness of the first layer to simulate correlation between the layers in an echogram. This was done for all four zones and the distribution of the normalized layer thickness is approximately Gaussian and we estimated the mean and variance of each.

Concretely, to simulate the layers in an echogram, the number of layers is randomly chosen from a uniform distribution

TABLE II
QUANTITATIVE SIMILARITY OF THE SIMULATED RADAR IMAGES WITH RESPECT TO THE REAL RADAR IMAGES, CALCULATED OVER THE TEST SET

Metric		cGAN	Physics simulator
SSIM	Min	0.39	0.31
	Mean	0.52	0.51
	Max	0.63	0.63
PSNR	Min	10.82	6.49
	Mean	23.1	16.61
	Max	29.6	23.25
RMSE	Min	8.44	17.55
	Mean	21.78	42.72
	Max	73.40	120.86

between 5 and 40. The mean thickness of the first layer is then generated from a uniform distribution based on the distribution of the first layer in the sample dataset and this determines the accumulation zone of the simulated echogram. Next, the mean thickness of each of the deeper layers is computed by multiplying the mean thickness of the first layer with the multiplicative factor drawn from the normalized layer thickness distribution of the accumulation zone. To capture the underlying random process governing the thickness of each layer, we fit a filtered Gaussian random process whose power spectral density matches that of the layer thickness process for each of the layers. Hence, to simulate a layer's thickness, we add the estimated mean layer thickness and the variance from the filtered Gaussian random process.

We conclude the generation of a new layer annotation set by accumulating the simulated thickness starting from the surface to the deepest layer.

The layer annotation is then passed as input to both the cGAN model (after training) and the physics based simulator to create simulated radar images.

C. Comparison of Simulation Algorithms

We now examine the synthetic images generated by both methods qualitatively, and analyze them in terms of certain quantitative metrics that we will describe in this section.

A sample of images simulated through both the physics simulator and cGAN, using a real image annotation, are shown in Fig. 10. From this figure, we see that the layers simulated through the physics simulator [see Fig. 10(d)] are able to capture the actual curvature of the labeled data [see Fig. 10(b)] better, whereas the images generated by the cGAN [see Fig. 10(c)] can capture the texture and contrast of the real radar images [see Fig. 10(a)] better. This can also be seen in the quantitative similarity between the simulated images and the real radar images, as tabulated in Table II, and noted in plots of Fig. 11.

We have conducted some quantitative comparisons using three metrics. The results on the whole set of our test data, consisting of 260 images, are presented in Table II and Figs. 11 and 12. The metrics used are root mean square error (RMSE), peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM) for PHY and purely data driven (cGAN) methods.

The RMSE of an image y with respect to a reference image x is calculated based on the following formula:

$$\text{MSE} = \frac{\sum_{i=1}^{N}(y_i - x_i)^2}{N} \tag{9}$$

assuming that x and y have same dimensions; here x_is and y_is are pixel values in x and y, respectively, and N is the total number of pixels of x. Furthermore, root mean squared error (RMSE) is defined as $\sqrt{\text{MSE}}$. A box plot for RMSE values across the entire test set and for the two methods is shown in Fig. 11(a).

To evaluate our results quantitatively, we include two other metrics, namely SSIM and PSNR.

The PSNR and signal to noise ratio (SNR) are commonly used in signal processing as image quality metrics. PSNR is expressed as

$$\text{PSNR} = 10\log_{10}\left(\frac{\text{peakval}^2}{\text{MSE}}\right) \tag{10}$$

where MSE is the MSE between an image and the reference image (with which the image is compared), and peakval is commonly the peak value of the range of the image data type; for instance, the peakval is 255 for an image of data type uint8. PSNR is most commonly used to measure the quality of reconstruction of compressed images. When comparing two images, the PSNR tends to infinity as the MSE tends to zero. It means that for identical images the PSNR is going to be infinite. Therefore, the more similar the two images are, the higher the PSNR value would be. Generally speaking, a higher PSNR indicates that the reconstruction quality is high.

Another metric used is the SSIM [47], which is a quality assessment index. It depends on three terms, namely the luminance term $l(x, y)$, the contrast term $c(x, y)$, and the structural term $s(x, y)$, where x and y represent the original and reference image. The overall index is a multiplicative combination of these three terms. The structural similarity index is expressed as

$$\text{SSIM}(x, y) = l(x, y)^{\alpha} c(x, y)^{\beta} s(x, y)^{\gamma} \tag{11}$$

where

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

and

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

and

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

here $\mu_x, \mu_y, \sigma_x, \sigma_y$, and σ_{xy} are the local means, standard deviations, and cross-covariance for images x, y. The constant C_1 is a small constant introduced in both the denominator and numerator to avoid instability when $\mu_x^2 + \mu_y^2$ is very close to zero. The constants C_2 and C_3 are introduced in a similar manner (see [47]). Here, $\alpha > 0$, $\beta > 0$, and $\gamma > 0$ are parameters used to adjust the relative importance of the three components. For simplicity, we set $\alpha = \beta = \gamma = 1$ and $C_3 = C_2/2$ in this article.

The SSIM attempts to model the structural change of an image by comparing small windows or subsamples in the image to compare the luminance, contrast, and structure of the two images. This metric gives us a robust measure of the perceived changes in the image. The closer the SSIM is to 1.0 the higher the quality of the image. To make the comparison more tangible, we calculate the SSIM as a function of depth for the whole dataset for both methods. The common depth among all our benchmark data is about 260 pixels and the snow surface is on average

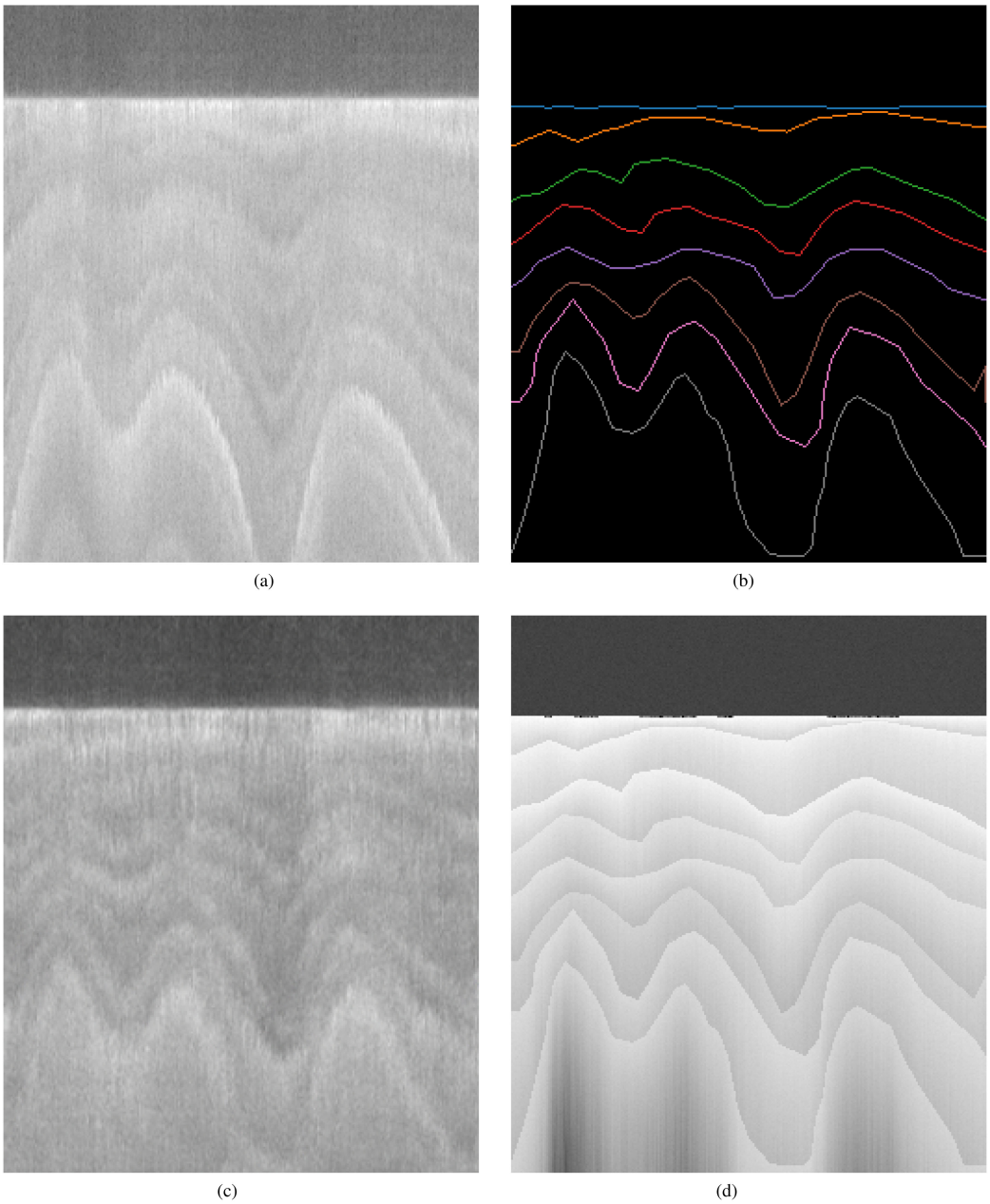

Fig. 10. Sample of a synthetic radar echogram simulated by cGAN (c) and physics simulators (d) using the label data (b) of a real radar image (a). The cGAN generator uses a binary mask of what is shown in (b) as input. (a) Real data. (b) Layer annotation. (c) GAN simulated data. (d) Physics simulated data.

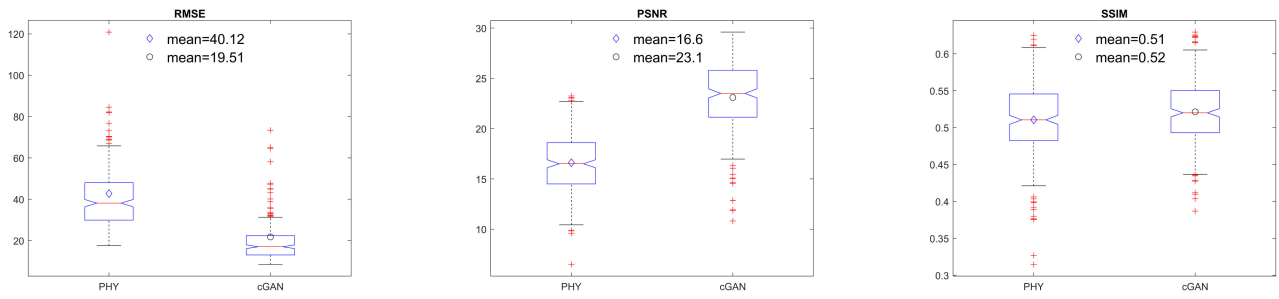

Fig. 11. Statistical comparison of three metrics for the whole test dataset. The metrics are root mean-square error (RMSE), PSNR, and SSIM for PHY and data driven (cGAN) methods.

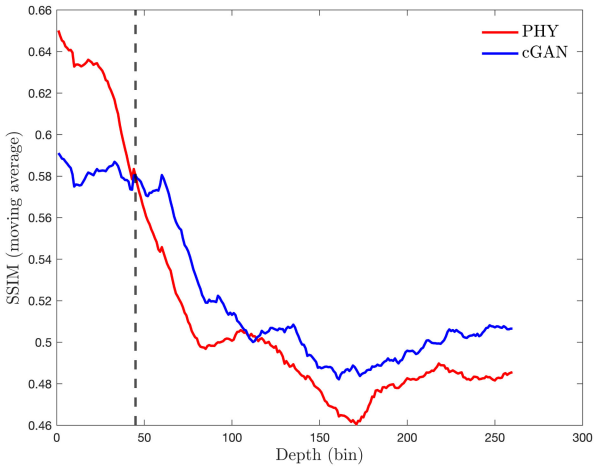

Fig. 12. SSIM as a function of depth. The vertical line denotes the approximate bin location of the surface return.

about 50 pixels below the top of the image. Fig. 12 shows the 50-moving average of the SSIM as a function of depth. It is apparent that that the PHY method does a better simulation above the surface, since there is almost no texture above the surface, while cGAN tends to add unnecessary texture. However, cGAN performs better below the surface.

From Table II, we see that the SSIM of physics simulator is close to the cGAN even though the former's output does not possess the same texture and contrast as the real radar images. This is primarily because the physics simulator replicates the exact curvature of the training labels, and is, hence, able to match the layer curvatures present in the real radar images. On the other hand, cGAN shows higher PSNR values than the physics simulator since cGAN is able to learn the pixel intensities and their gradients from real radar training images. Moreover, cGAN gives a lesser RMSE than the physics simulator, highlighting the former's usefulness in reconstructing images.

VI. Model Training and Predictions

As mentioned previously, one objective of generating synthetic data is to build a larger and more robustly annotated dataset. This dataset is going to be used in training deep learning models for tracking the internal annual accumulation layers. In our previous works [24], [25], we have proposed a deep learning model for tracking internal annual accumulation layers, using real data. As discussed in the aforementioned works, to improve the accuracy of prediction of our model, we need many more fully annotated data. In this section, we will use a combination of our real data and the newly simulated data to train our deep neural network.

Our approach to generate simulated data for this experiment is explained in Section V-A. We first generated about 1500 pairs of simulated radar images and the corresponding annotated data (or labels) using the PHY simulator; we call this dataset the physics-generated data (PD). Next we used the cGAN simulator to generate another set of simulated data. To this end, we used the 1500 "labels" from PD, which was synthetically generated based on the statistics of the sample dataset, and generated corresponding near-real simulated data using the (pretrained) cGAN generator. We call this dataset the cGAN-generated data (GD).

In this work, we have conducted the following experiments:

1) we train our deep learning model on real training data (RD) and PD, and test it on our (real) test set;
2) we train our deep learning model on RD and GD, and test it on our (real) test set;
3) we train our deep learning model on RD and PD and GD, and test it on our (real) test set.

For training our deep CNN, we have kept the same hyperparameters as in [24].

Here, we report three metrics for evaluating the performance of our internal layer tracker; namely, the optimal dataset scale (ODS) or best F-measure on the dataset for a fixed scale, the optimal image scale (OIS) or aggregate F-measure on the dataset for the best scale in each image, and the average precision (AP) on the full recall range (equivalently, the area under the precision-recall curve) [48]. For evaluation purposes, we apply a standard postprocessing nonmaximal suppression technique to obtain thinned edges.

Table III shows the evaluation results of our experiments. The evaluation is performed using the same test dataset (our test set) as in [24]. The first set of three columns in Table III presents the evaluation results for tracking the internal layers after the model is trained on real snow radar data. The second set of three columns shows the quantitative evaluation results for the layer tracking model trained on augmented training set by synthetic radar data generated by the cGAN. With the same pattern, the rest of the table shows the evaluation results for the layer tracking model trained augmented training set by synthetic radar data generated by the physics simulator, and finally the evaluation results after training the model on augmented training set by RD, PD, and GD. We use the fuse as our final predictions, the best results are in bold fonts.

VII. Discussions

As reported in Table III, the addition of the GD and the PD can improve all outputs of our multiscale deep learning model. This justifies the notion that augmenting the training set of deep neural networks with well annotated synthetic ground truth improves the algorithm's performance on radar data.

Our multiscale deep learning model is explained in [24] and [25] where the notion of side-output is described as well. Briefly speaking, the main property of the side-outputs is that the later side-outputs present the coarser structure of the image, while the earlier side-outputs represent the low-level details of the image. We see that at high resolutions, the simulated data captures the features that help the neural network distinguish layer containing pixels from nonlayer pixels resulting in the better performance of the deep neural network. However, given that the fifth side output is worse, this suggests that the added simulated data when downsized to very low resolution misleads the network.

The experiment of augmenting data with the physics simulated echogram reports the best final prediction (the fuse output), in addition to good improvement for all other side-outputs.

This confirms that the surface altimetry convolution model correctly captures the underlying processes responsible for layering information in echogram images, which results in a better performance of the algorithm. This also confirms the adequacy of the model as a good alternative to other rather computationally expensive simulation models, such as the FDTD model.

However, it has to be noted that the PHY "labels" for synthetic radar images are produced by imitating the geometries of the layers of real radar echograms in the sample dataset. Therefore,

TABLE III
EVALUATION OF OUR MULTISCALE DEEP LEARNING MODEL [24] TRAINED ON VARIOUS AUGMENTED DATASETS

	RD [24]			RD & GD			RD & PD			RD & PD & GD		
	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
Side 1	0.320	0.465	0.261	0.516	0.601	0.520	0.393	0.499	0.329	0.473	0.559	0.432
Side 2	0.763	0.779	0.760	0.755	0.778	0.736	0.734	0.742	0.706	0.740	0.745	0.719
Side 3	0.796	0.824	0.786	0.810	0.847	0.798	0.797	0.831	0.784	0.798	0.835	0.788
Side 4	0.732	0.769	0.645	0.775	0.811	0.708	0.769	0.810	0.719	0.769	0.813	0.717
Side 5	0.512	0.572	0.399	0.437	0.469	0.205	0.430	0.464	0.207	0.437	0.466	0.204
Fuse	0.815	0.854	0.815	0.825	0.874	0.819	**0.829**	0.876	0.813	0.825	**0.877**	**0.821**

the synthetic layer output of the physics generator depends on diversity of the layer geometries in the sample dataset used to set the parameters of the physics simulator—in this sense, the generator is data-driven. As noted earlier, most of the images in the current sample dataset are from the dry-snow zone in central Greenland. We note that this is a limitation in the geometry of layers produced in the synthetic dataset, which does not capture all the possible layer geometries such as those from percolation, wet snow, and ablation zones.

In general, the test data for evaluation of the deep learning method come from all over Greenland including zones with rapidly varying accumulation conditions and the ice sheet margins where significant melt occurs. These real test data have greater variation in their layer geometries including sharper changes and high slopes than those generated by the simulators. Therefore, simulating layers with these geometries will potentially improve the performance of the deep neural network.

It is also important to note that the experiment of augmenting data with the cGAN simulated echogram reports the best results for almost all of the metrics for the side-outputs; this confirms that the GD can capture the details and noise in images better, as we expected.

VIII. CONCLUSION

Our contribution in this work includes the following:

1) developing the first PHY algorithm for simulating the internal layers of ice sheets;
2) generating near-real snow radar images using purely data driven algorithms;
3) extensive experimental results on data driven and PHY simulators;
4) deep quantitative and qualitative comparison of the two aforementioned methods and their effects on improving the training of neural networks.

In this work, we implemented novel approaches to deal with the lack of fully annotated dataset, which is a major challenge in applying supervised deep learning algorithms, particularly to radar data. We explored two approaches to create "near-real" synthetic radar images using (i) a data-driven cGAN based approach and (ii) a PHY approach, using the glaciological relationship between snow layers, which we called the physics simulator (PHY). To the best of authors' knowledge, this is the first work that uses both cGAN and physics-based approach using the surface altimetry convolution model to create simulated radar echograms with snow accumulation layers.

We find that the physics simulator is able to learn the layer curvature better than GANs, whereas the latter can learn the texture and contrast variations better. This results in cGAN method showing better SSIM, PSNR, and RMSE than the physics simulator.

Using the physics simulator layer generator, we created new layers that match the physical properties (probability density function and power spectral density) of real layers and used them as input to the PHY and cGAN methods to create simulated echograms.

We showed that by adding simulated data to existing radar dataset, the performance of a deep learning model can be improved. In two of our experiments, we expanded our train dataset by about 50% by adding simulated data, and recorded improvements in two major metrics. To further improve these results, we note possible improvements to the current simulated dataset, which does not contain all the accumulation conditions that exist in real radar data. Expanding the sample dataset from which the physics simulator is parameterized will improve the simulated echograms to include layers with nonstationary mean and high curvatures similar to those seen in real radar echograms. Filling this gap is the subject of our future research.

REFERENCES

[1] A. Shepherd *et al.*, "A reconciled estimate of ice-sheet mass balance," *Science*, vol. 338, no. 6111, pp. 1183–1189, 2012.

[2] B. Medley *et al.*, "Airborne-radar and ice-core observations of annual snow accumulation over thwaites glacier, west antarctica confirm the spatiotemporal variability of global and regional atmospheric models," *Geophys. Res. Lett.*, vol. 40, no. 14, pp. 3649–3654, 2013.

[3] L. S. Koenig *et al.*, "Annual Greenland accumulation rates (2009-2012) from airborne snow radar," *Cryosphere*, vol. 10, no. 4, pp. 1739–1752, 2016.

[4] E. Arnold *et al.*, "Cresis airborne radars and platforms for ice and snow sounding," *Ann. Glaciol.*, vol. 61, no. 81, pp. 58–67, 2020.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, *arXiv:1409.1556 [cs.CV]*.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: https://doi.org/10.1145/3065386

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2016.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[11] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.

[12] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2self with dropout: Learning self-supervised denoising from single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1890–1898.

[13] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[14] D. Varshney, M. Rahnemoonfar, M. Yari, and J. Paden, "Deep ice layer tracking and thickness estimation using fully convolutional networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2020, pp. 3943–3952.
[15] R. J. Arthern, D. G. Vaughan, A. M. Rankin, R. Mulvaney, and E. R. Thomas, "In situ measurements of antarctic snow compaction compared with predictions of models," *J. Geophys. Res.: Earth Surf.*, vol. 115, no. F3, pp. 2–8, 2010.
[16] M. M. Herron and C. C. Langway, "Firn densification: An empirical model," *J. Glaciol.*, vol. 25, no. 93, pp. 373–385, 1980.
[17] M. Hörhold, S. Kipfstuhl, F. Wilhelms, J. Freitag, and A. Frenzel, "The densification of layered polar firn," *J. Geophys. Res.: Earth Surf.*, vol. 116, no. F1, pp. 1–4, 2011.
[18] N. Reeh, "A nonsteady-state firn-densification model for the percolation zone of a glacier," *J. Geophysi. Res.: Earth Surf.*, vol. 113, no. F3, pp. 2–6, 2008.
[19] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[20] M. Rahnemoonfar, J. Johnson, and J. Paden, "Ai radar sensor: Creating radar depth sounder images based on generative adversarial network," *Sensors*, vol. 19, no. 24, 2019, Art. no. 5479.
[21] M. Rahnemoonfar, M. Yari, and J. Paden, "Radar sensor simulation with generative adversarial network," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 7001–7004.
[22] L. Yang, D. Zhang, and G. E. Karniadakis, "Physics-informed generative adversarial networks for stochastic differential equations," *SIAM J. Sci. Comput.*, vol. 42, no. 1, pp. A 292–A317, 2020.
[23] U. A. Ciftci, I. Demir, and L. Yin, "FakeCatcher: Detection of synthetic portrait videos using biological signals," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2020.3009287.
[24] M. Rahnemoonfar, M. Yari, J. Paden, L. Koenig, and O. Ibikunle, "Deep multi-scale learning for automatic tracking of internal layers of ice in radar data," *J. Glaciol.*, vol. 67, no. 261, pp. 39-48, 2021. [Online]. Available: https://doi.org/10.1017/jog.2020.80
[25] M. Yari, M. Rahnemoonfar, and J. Paden, "Multi-scale and temporal transfer learning for automatic tracking of internal ice layers," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 6934–6937.
[26] C. Leuschen, S. Clifford, and P. Gogineni, "Simulation of a surface-penetrating radar for mars exploration," *J. Geophys. Res.: Planets*, vol. 108, no. E4, pp. 5–13, 2003.
[27] C. Gerekos, A. Tamponi, L. Carrer, D. Castelletti, M. Santoni, and L. Bruzzone, "A coherent multilayer simulator of radargrams acquired by radar sounder instruments," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7388–7404, Dec. 2018.
[28] R. Culberg and D. M. Schroeder, "Firn clutter constraints on the design and performance of orbital radar ice sounders," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6344–6361, Sep. 2020.
[29] F. Rodriguez-Morales *et al.*, "Advanced multifrequency radar instrumentation for polar research," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2824–2842, May 2014.
[30] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.
[31] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," in *Adv. Neural Inf. Process. Syst.*, vol. 28., 2015.
[32] X. Wang and A. Gupta, "Generative image modeling using style and structure adversarial networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 318–335.
[33] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," in *Proc. 4th Int. Conf. Learn. Representations*, 2016.
[34] Y.-H. Kwon and M.-G. Park, "Predicting future frames using retrospective cycle GAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1811–1820.
[35] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," 2016, *arXiv:1612.00215*.
[36] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.
[37] P. Wang, H. Zhang, and V. M. Patel, "Generative adversarial network-based restoration of speckled SAR images," in *Proc. IEEE 7th Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, 2017, pp. 1–5.
[38] D. Ao, C. O. Dumitru, G. Schwarz, and M. Datcu, "Dialectical gan for sar image translation: From sentinel-1 to terrasar-x," *Remote Sens.*, vol. 10, no. 10, 2018, Art. no. 1597.
[39] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.
[40] G. Brown, "The average impulse response of a rough surface and its applications," *IEEE Trans. Antennas Propag.*, vol. AP-25, no. 1, pp. 67–74, Jan. 1977.
[41] R. K. Moore and C. Williams, "Radar terrain return at near-vertical incidence," *Proc. IRE*, vol. 45, no. 2, pp. 228–238, Feb. 1957.
[42] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
[43] J. A. MacGregor *et al.*, "Radiostratigraphy and age structure of the Greenland ice sheet," *J. Geophys. Res.: Earth Surf.*, vol. 120, no. 2, pp. 212–241, 2015.
[44] B. Panzer *et al.*, "An ultra-wideband, microwave radar for measuring snow thickness on sea ice and mapping near-surface internal layers in polar firn," *J. Glaciol.*, vol. 59, no. 214, pp. 244–254, 2013.
[45] S. Gogineni, J. Yan, D. Gomez, F. Rodriguez-Morales, J. Paden, and C. Leuschen, "Ultra-Wideband Radars for Remote Sensing of Snow and Ice," in *Proc. IEEE MTT-S Int. Microw. RF Conf.*, 2013, pp. 1–4.
[46] J.-B. Yan *et al.*, "Airborne measurements of snow thickness: Using ultrawide-band frequency-modulated-continuous-wave radars," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 2, pp. 57–76, Jun. 2017.
[47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[48] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

Masoud Yari received the Ph.D. degree in applied mathematics from Indiana University, Bloomington, IN, USA, in 2008.

He is currently a Research Professor with NSF-HDR iHARP Institute, University of Maryland, Baltimore County, Baltimore, MD, USA. His research interests include mathematical foundations of machine learning, computer vision, partial differential equations, and dynamical systems.

Oluwanisola Ibikunle is currently working toward the Electrical Engineering Ph.D. degree in deep learning algorithms for remote sensing data with the University of Kansas, Lawrence, KS, USA.

He is a part of the Radar Data Processing Group, Center for Remote Sensing of Ice Sheets (CReSIS), Lawrence, KS, USA. He currently works on developing signal processing and machine/deep learning algorithms to solve the echogram internal layer detection and tracking problem.

Debvrat Varshney received the B.E. degree in electronics and instrumentation from the Birla Institute of Technology and Science (BITS) Pilani, India, in 2014, and the M.Sc. degree in earth observation from the joint collaboration between the University of Twente, Enschede, The Netherlands, and the Indian Space Research Organization (ISRO), Bengaluru, India, in 2019. He is currently working toward the Ph.D. degree in artificial intelligence (AI) from the University of Maryland Baltimore County, Baltimore, MD, USA.

He is an experienced Software Engineer. He is building deep learning algorithms for remotely sensed images with the University of Maryland Baltimore County. His research interests include semi-supervised learning and physics guided neural networks to efficiently process large climate datasets.

Tashnim Chowdhury received the B.S. degree in electrical and electronic engineering from the Chittagong University of Engineering and Technology, Chittagong, Bangladesh, in 2013 and the M.S. degree in electrical engineering from The University of Toledo, Toledo, OH, USA, in 2016. He is currently working toward the Ph.D. degree in information systems from the University of Maryland Baltimore County, Baltimore, MD, USA.

His research interests include deep learning, machine learning, semantic segmentation, few shot learning, meta learning, and Bayesian learning.

Argho Sarkar received the B.S. degree in applied statistics from the University of Dhaka, Dhaka, Bangladesh, in 2018. He is currently working toward the Ph.D. degree in information systems from the University of Maryland, Baltimore County, Baltimore, MD, USA.

His research focuses on developing algorithms for multimodal applications such as visual question answering and image captioning for medical and climate issues.

John Paden (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from the University of Kansas (KU), Lawrence, KS, USA, 2003 and 2006, respectively, studying radar signal and data processing for remote sensing of the cryosphere.

After graduating, he joined Vexcel Corporation, Boulder, CO, USA, a remote sensing company, where he was a Systems Engineer and a SAR Engineer for three and a half years before rejoining the Center for Remote Sensing of Ice Sheets (CReSIS), KU, in early 2010 to lead the radar signal and data processing efforts. He is currently a Research Faculty Member with CReSIS with a courtesy appointment with the Electrical Engineering and Computer Science Department.

Dr. Paden was a recipient of the American Geophysical Union Cryosphere Early Career Award in 2016. He was also the recipient of three NASA group achievement awards for radioglaciology work.

Jilu Li (Senior Member, IEEE) received the B.S. and M.S. degrees in aeronautics from Northwestern Polytechnical University, Xi'an, China, in 1985 and 1988, respectively, and the Ph.D. degree in aeronautics from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1992, and the M.S. degree in aerospace engineering, and the M.S. and Ph.D. degrees in electrical engineering from the University of Kansas, Lawrence, KS, USA, in 2003, 2006, and 2009, respectively.

From 1992 to 1994, he was a Postdoctoral Research Fellow with the Beijing Aerospace Automatic Control Institute, Beijing, China. From 1994 to 2000, he was a Senior Engineer with the Beijing Institute of Control Engineering, Beijing, China. From 2009 to 2010, he was a Senior Engineer with LinkQuest Inc., San Diego, CA, USA. Since 2011, he has been a Research Faculty Member with the Center for Remote Sensing of Ice Sheets (CReSIS), University of Kansas. His research interests are in developing advanced signal and array processing algorithms for processing and interpreting ice sheet radio echo sounding data. He has been participating in field experiments in Antarctica, Greenland, and Alaska, since 2011.

Dr. Li is a Member of the International Glaciology Society.

Maryam Rahnemoonfar received the Ph.D. degree in computer science from the University of Salford, Manchester, U.K., 2010.

She is currently an Associate Professor, the Director of NSF-HDR iHARP Institute, and the Director of Computer Vision and Remote Sensing Laboratory (Bina lab), University of Maryland, Baltimore County, Baltimore, MD, USA. Her research interests include deep learning, computer vision, data science, AI for social good, remote sensing, and document image analysis. Her research specifically focuses on developing novel machine learning and computer vision algorithms for heterogeneous sensors such as Radar, Sonar, Multispectral, and Optical. Her research has been funded by several awards including NSF BIGDATA, Amazon Academic Research, Amazon Machine Learning, Microsoft, and IBM.