An Interpretable, Satellite-Based Convection Detection Algorithm Created by an Explainable Boosting Machine and Refined using Domain Knowledge

— DRAFT —

Nathan Mitchell, Lander Ver Hoef, Imme Ebert-Uphoff, Kristina Moen, Kyle Hilburn, Yoonjin Lee, Emily J. King

[Also need affilitations matched to authors]

Department of Statistics, Colorado State University, Fort Collins, Colorado

Cooperative Institute for Research in the Atmosphere, Colorado State University, Fort Collins, Colorado

Department of Mathematics, Colorado State University, Fort Collins, Colorado

Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, Colorado

ABSTRACT: Explainable Boosting Machines (EBMs) are extensions of Generalized Additive Models that have been shown to achieve comparable performance on some applications to modern machine learning (ML) techniques while retaining the interpretability of glass-box methods. The overall goal of this work is twofold: (1) explore the use of EBMs, in combination with feature engineering, to obtain interpretable, physics-based machine learning algorithms for meteorological applications; (2) illustrate these methods for the application of detecting deep convection in the upper troposphere from satellite imagery.

Specifically, we seek to simplify the complex process of convection detection by combining two frameworks: the use of mathematical methods to extract key features, such as the texture of clouds, and EBMs. Our EBM model focuses on the binary classification task of predicting regions as either convective or non-convective utilizing channels of the Advanced Baseline Imager (ABI) sensor of the Geostationary Operational Environmental Satellite 16 (GOES-16), namely channel 2, visible imagery, and channel 13, infrared imagery. The mathematical framework of Gray-Level Co-occurrence Matrices (GLCMs) is used to extract texture information from the visible imagery, and flags from the Multi-Radar/Multi-Sensor (MRMS) system data are used as ground truth.

Once trained, the EBM's feature functions, which represent the strategies used by the model for its predictions, were examined and altered as necessary to more closely match strategies used by domain scientists to detect convection. This process allowed us to utilize domain knowledge regarding the physics of convective systems to inform changes to these feature functions which, in turn, improved the model's predictions.

The result of our efforts is a fully interpretable ML algorithm that was developed in a human-machine collaboration. Note that the altered model performs generally better than the unaltered model. While the final model does not yet reach the accuracy of more complex neural network approaches for this application, it performs well and represents a significant step towards building fully interpretable ML algorithms for this and other meteorological applications.

1. Introduction

The ability to detect regions of mature convection based on satellite imagery is a complex task. This task has been taken on multiple times before and, in recent years, has been approached through the lens of increasingly complex machine learning models. In this paper, we seek to accomplish the same task but with a simplified approach. The driving forces behind this simplification are clever feature engineering and the fully interpretable Explainable Boosting Machine (EBM) machine learning algorithm. With these two aspects in mind, our goals are two-fold: we seek to (1) advertise the ML technique of EBMs and its unique capabilities to the broader atmospheric science community and (2) examine how such a model can be harnessed to predict regions of mature convection from satellite imagery.

a. Target Application - Detecting Convection in Satellite Imagery

Being able to detect and monitor convection is important because of its connection to severe weather, which includes hazards from hail, strong winds, and tornadoes. Severe deep convection also poses risks to aviation, in particular from turbulence. Severe storms are a large and growing contributor to billion-dollar disasters in the United States. In their original analysis, Smith and Katz (2013) found that severe storms from 1980-2011 contributed 10% to the cost and 32% of events. In a more recent analysis by NOAA National Centers for Environmental Information (NCEI) (2024), the contribution to the cost by severe storms has increased to 17% over the period 1980-2023, and severe storms are nearly 50% of the events.

The gold standard for detecting and monitoring convection is ground-based radar, however much of the world lacks ground-based radar coverage either due to resource limitations associated with the cost of building and maintaining such a radar network or due to physical limitations such as radar beam blockage by terrain. This motivates the development of satellite-based products to provide information in radar data void regions, and the latest generation of geostationary satellite imagers are ideally suited to this task. For example, the GOES-R Series Advanced Baseline Imager (ABI) (Schmit et al. 2017), has a spatial resolution ranging from 0.5 to 2 km and a temporal refresh rate ranging from 0.5 to 15 minutes. These scales are capable of observing convection; visible imagery from ABI in Fig. 1(a) shows cloud top bubbling associated with deep convection. Multi-spectral information enhances situational awareness with infared imagery in Fig. 1(b) highlighting cold

cloud-top temperatures associated with deep convection. Together, this information from ABI can identify areas of radar-indicated convection in Fig. 1(c).

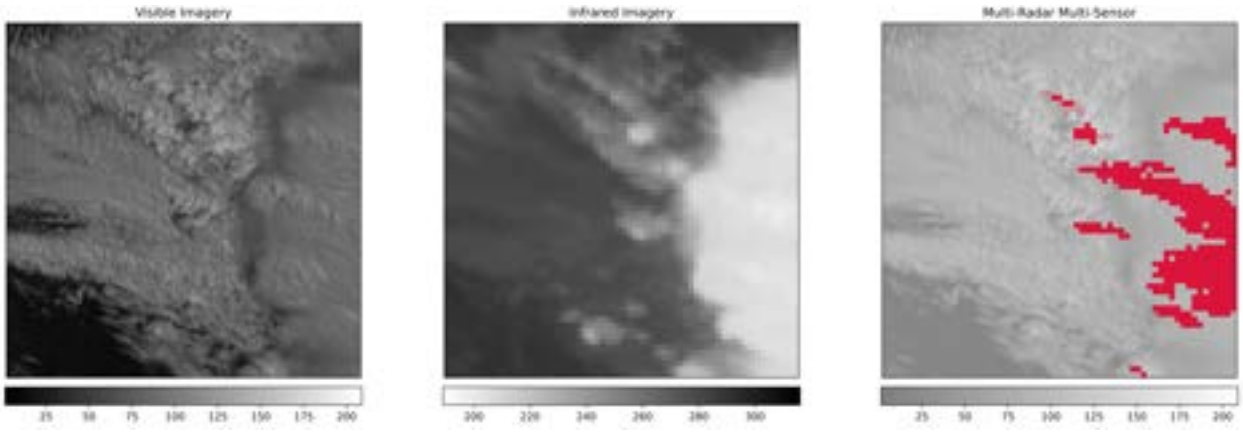
Machine learning provides a natural choice for extracting convection information from satellite imagery because of its ability to learn these statistical relationships and make use of the information content in gradients and spatio-temporal patterns. For example, Veillette et al. (2018) derives Vertically Integrated Liquid, Lee et al. (2021a) derives precipitation type, and Hilburn et al. (2021) derives composite radar reflectivity using convolutional neural networks. A limitation with convolutional neural networks is that it can be difficult to understand how the network makes the predictions. Explainable AI approaches such as Layerwise Relevance Propagation (LRP) (Ebert-Uphoff and Hilburn 2020) can be used to interrogate particular pixels and provide explanations for what channels and spatial features made the largest contributions to the prediction. However, this does not fully explain the strategies behind those contributions and does not provide insights for how the network might perform in other situations.

This motivates the development of machine learning approaches that build the explainability into the model right from the start, providing a so-called interpretable machine learning model. Hilburn (2023) attempted this by replacing a neural network with linear regression and by including inputs involving spatial derivatives and pooling operations. This approach was able to reveal additional prediction strategies that were not obvious using LRP. However, the data preparation and model training were cumbersome using this approach. In this research, we explore the more efficient and interpretable-by-design approach of Explainable Boosting Machines.

b. Strategies to Detect Convection

Approaches to convection detection range from simple techniques that use numerical thresholds to complex deep learning models. These methods, however, all tend to approach the problem in the same way, namely by identifying overshooting tops. The product of strong atmospheric instability and subsequent vertical updrafts, overshooting tops can be identified from satellite-based imagery in multiple ways by considering the relationship they have with their surroundings.

We highlight two primary strategies to identify overshooting tops:



(a) (b) (c)

35 FIG. 1. Examples of (a) Visible Imagery (Channel 2), (b) Infrared Imagery (Channel 13), (c) and Multi-Radar
36 Multi-Sensor (MRMS) scenes, each from 5 May 2021.

- Strategy 1 seeks to identify overshooting tops by considering the elevation difference seen between the overshooting tops and the surrounding cirrus anvil clouds by numerically quantifying this displacement using a proxy such as cloud-top temperature.

- Strategy 2 involves numerically quantifying texture within regions (typically represented by small groups of pixels) of visible imagery to then isolate the bumpy, textured overshooting tops from their flatter, less-textured anvil cloud companions.

In this work, we combine these two strategies by considering both cloud-top temperature and texture.

c. Existing approaches

Thresholds are a classic approach to identifying convective regions. Fixed values of brightness temperature can be used as a threshold to determine which regions represent overshooting tops and which regions represent the surrounding anvil (Ai et al. 2017). Which threshold value should be chosen, however, depends greatly upon the context in which the threshold is to be used (e.g. the season of interest), implying there does not exist a one-size-fits-all threshold value that can be chosen to reliably separate overshooting tops from their surroundings. A natural extension of this method utilizes a difference between brightness temperature values, namely those from the water

vapor absorption channel and the atmospheric infrared window channel (Ai et al. 2017). The same issue regarding chosen threshold values exists with this approach.

While brightness temperature is a common metric, another approach uses visible (VIS) channels to inform predictions based on calculated texture values. For example, as a small part of their identification process, Bedka, Kristopher et al. (2019) used imagery from a VIS channel to calculate the standard deviation of the brightness of a cloud within a 5 x 5 pixel box. If this value was sufficiently high, its status as a cumulus cloud (which was determined from a previously defined measure) was retained. Otherwise, it was re-classified as being cirrus. Their algorithm did not stop at that step, but we use this initial step to motivate the idea that extracted texture can be thresholded at specific values to separate texture into desirable, informative groups.

Moving on to the machine learning algorithms, we first note a clustering approach by Berendes et al. (2008). They used the Standard Deviation Limited Adaptive Clustering procedure to detect cumulus clouds (among other categories, including overshooting tops) based on a combination of both VIS and infrared (IR) data. Berendes et al. (2008) created a new feature used in their algorithm by calculating a "grey level difference vector" and extracting the contrast measure from the VIS channel used—a measure which they describe as being useful for identifying both cumulus and stratocumulus clouds.

An approach by Bedka and Khlopenkov (2016) sought to mimic how a human analyst might identify convection within cloud-top imagery. The first step in their algorithm dealt with the identification of cirrus anvil clouds that often surround the overshooting tops of interest. Next, a pattern recognition scheme was used in order to detect said overshooting tops within the anvils. Finally, a logistic regression model was fit to identify the probability of a region being part of an overshooting top. The three features used to train the model all took into account temperature differences between the overshooting top itself and various other measures (Bedka and Khlopenkov 2016). We found this method's human-like approach to be a desirable property for a convection detection algorithm.

Finally, we consider an approach to convection detection using neural networks. We focus on the use of a convolutional neural network (CNN) first developed by Lee et al. (2021a) but used again, with some updates, in Bansal et al. (2023). CNNs are a promising tool for this application as they were developed with image-based tasks in mind and excel at image-based pattern recognition

tasks. A second reason for us to focus on the work by Lee et al. (2021b) is familiarity, since some of the authors of this paper were involved in that work as well.

Their CNN was trained on satellite-based reflectance and brightness temperature data. An important result from Lee et al. (2021a) with respect to our research is that the CNN was able to learn—on its own—that "bubbling" was a "main feature" of convection. Much like the other studies mentioned, being able to accurately detect the presence of overshooting tops, those with a "bubbly" texture, within cloud-top-based imagery is an important step toward predicting the presence of convection and mapping out regions as being convective or non-convective. While the performance of CNNs is impressive, they are very hard to interpret, i.e., we do not fully understand their reasoning. This motivates our search for simpler, more interpretable ML methods with comparable performance.

d. Black-box models, Explainable AI, and Interpretable AI

Neural Networks, more specifically CNNs, are designed to find patterns within image-based data. These methods have been shown to perform well at such tasks, but at a cost, namely their inherent lack of interpretability. Such models are known as being black-box. In high stakes applications—such as forecasting severe weather—this lack of interpretability can be a problem as forecasters may not know how the models arrived at their conclusions and failure modes of the model are hard to anticipate.

The field of Explainable AI (XAI) was developed to add explainability back into these black-box models after they have been trained (post-hoc explainations). Such tools include attribution maps, such as Shapley Additive Explanations (SHAP) (Shapley 1953) and Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al. 2016), but these models still inherently lack interpretability, and face many other challenges (Rudin 2019; Mamalakis et al. 2022, 2023).

Another approach to XAI involves the development of inherently interpretable models, i.e. models that do not require such post-hoc explanations in order to be interpreted. Opposed to their black-box counterparts, these models are known as being glass-box and are the type of model that we pursue here.

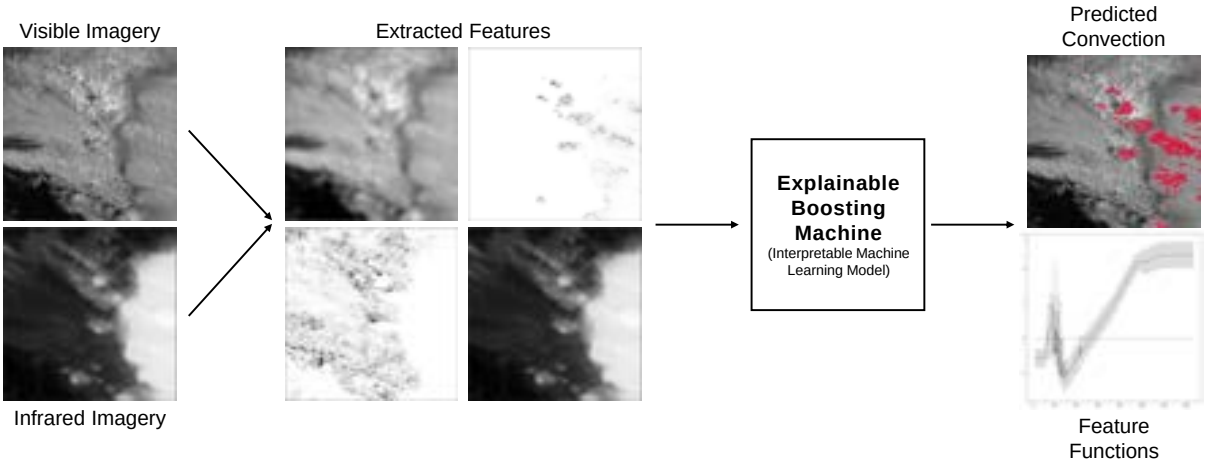


FIG. 2. Summarized approach showing VIS and IR channel data from 5 May 2021(a), our extracted features (b), the method used (c), and a finalized prediction as well as a feature function (d).

e. Toward an interpretable model

In our search for a high-performing yet more interpretable approach than CNNs, we used these existing approaches in Section bc to inform our own process. First, these studies demonstrated that VIS channel data can be used to extract texture-related information (in various ways) that can be leveraged to identify the distinct, bubbly cloud-top surfaces that are associated with overshooting tops and convective cells. Second, these studies demonstrated that IR channel data can be used to extract temperature-related information that can be leveraged to delineate between an anvil cloud and an overshooting top based on temperature differences. Finally, these studies demonstrated that machine learning algorithms including clustering algorithms, logistic regression, and CNNs are useful for image-based convection detection. Furthermore, we have seen that machine learning algorithms can be led to approximate human-based decision making processes.

Our approach combines these ideas into one framework and seeks to develop a convection detection algorithm that is both simpler and fully interpretable. To do this, we propose to combine the following steps:

1. Use VIS data for brightness information,

2. Use the mathematical framework of Gray-level Co-occurrence Matrices (Moen 2024) for texture extraction from VIS channel data,

3. Use IR channel data for temperature information,

4. Combine all of that information using the framework of Explainable Boosting Machines.

The first three steps above represent **feature engineering**, i.e., using domain knowledge and mathematical methods to extract key information from the imagery. The last step combines that information using an interpretable ML method to provide a prediction.

We summarize this approach in Figure 2.

As we will see in Section 4, the final model we developed based on this approach does not perform as well as the CNN in Bansal et al. (2023). We hope to close this gap with future improvements. Despite the gap, however, our model performs surprisingly well—especially considering its very limited number of features, the simplicity of the features used, and the fact that it is a fully interpretable model. We see this work as a promising first step toward the development of fully interpretable models, not only for convection detection but for numerous other potential applications that seek to extract information from meteorological imagery.

f. Introducing Explainable Boosting Machines

Explainable Boosting Machines (EBMs) are functionally similar to Generalized Additive Models (GAMs). At their core, these nonparametric models aim to estimate how each feature relates to the outcome of interest. These relationships are expressed as independent functions between the inputs and the outcome. These functions are the driving force behind GAMs and EBMs being interpretable as they allow the user to gain insight into the strategies the model learned. One major difference between GAMs and EBMs—as well as between EBMs and all other ML models—is that the learned strategies can be altered after the model has been trained *without* the need to retrain the model.

1) EBMs enable Post-Hoc Model Alteration

There are many different ways in which a model's interpretable nature can be harnessed during the post-hoc model assessment and the real-world decision-making processes. For instance, a researcher may determine some particular relationship that the model learned does not accurately reflect the real-world phenomenon it was attempting to model—something that could not have

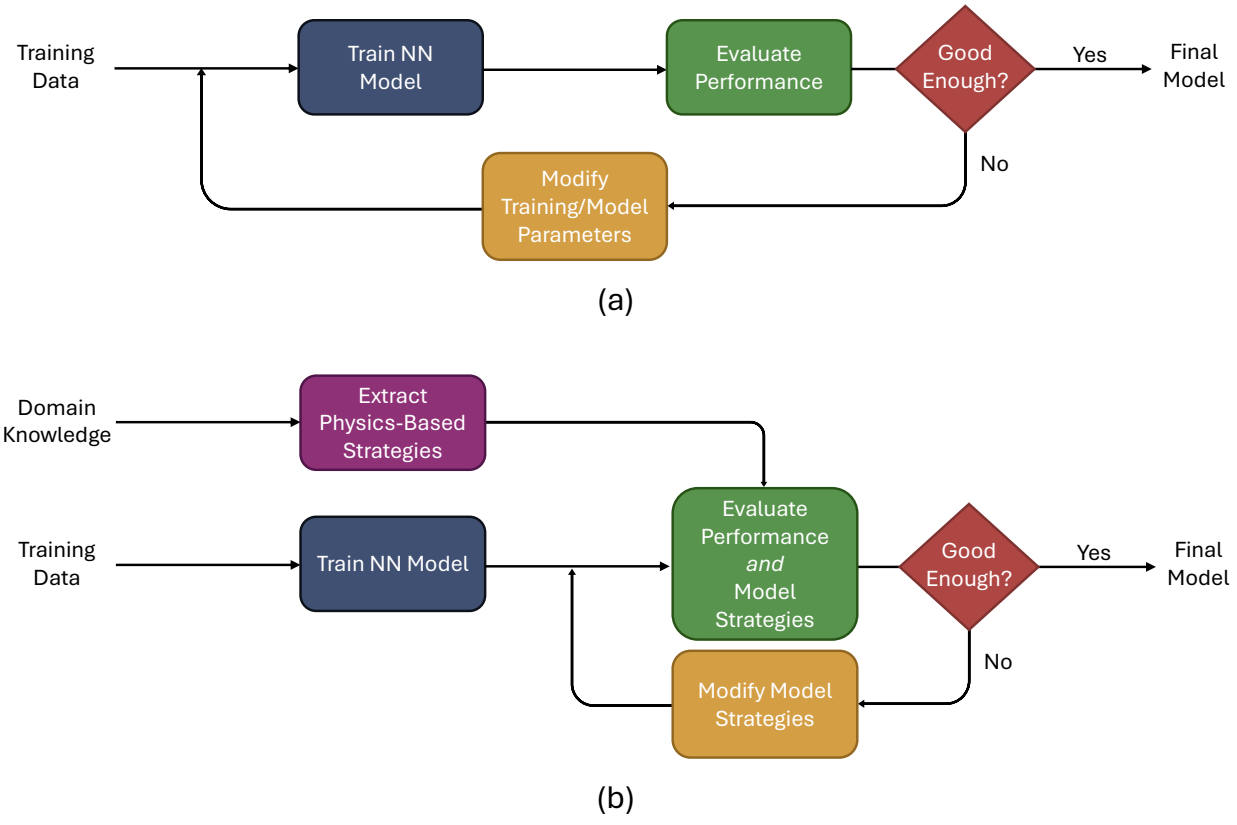


FIG. 3. Examples of two ML workflows—one for CNNs (a) and one for EBMs (b).

been done if a model that lacked interpretability was used—and choose to correct the error when implementing the learned information.

Furthermore, if any model besides an EBM had been used—even an interpretable one—that would be the extent of what could be done regarding an incorrect strategy without re-training the entire model. We note here, however, that re-training the model is a common practice often implemented when building machine learning models. Figure 3 shows two typical workflows, one for models such as CNNs (a), and one for EBMs (b). Though these workflows look similar, EBMs differ in what can be done about correcting inadequate performance. If a CNN's performance does not meet some desired threshold, the hyperparameters can be tuned and a new model can be trained. This loop continues until a desired model has been constructed and adequate performance is observed.

In contrast, even before the performance of the EBM is evaluated, the strategies that were learned by the model can be compared to the "expected" physics-based strategies that were derived

from knowledge of the system. This step is part of model evaluation and may lead to a deeper understanding of the model's performance. If the model does not perform well, the learned strategies *themselves* can be altered without the need to retrain the model. After the model has been updated, the strategies can be re-examined and the performance of the altered model can be evaluated. This loop, then, continues until a desired model has been constructed and adequate performance is observed.

Functionally, this loop is similar in nature to the loop seen in the CNN workflow. Each loop seeks to change the model to achieve better performance, but the way in which the EBM can be updated is far more informed than the way in which CNNs can be changed. There are a limited number of ways in which outside information about the system being studied can be embedded into a CNN (e.g. through feature engineering), but this limitation is not inherent to EBMs. In this work, we explore how domain knowledge can be used to inform changes made to a fully trained model to achieve better predictions.

2. Data and Feature Extraction

We re-use the data set developed by Lee et al. (2021a) because (1) the data set was available in-house and (2) using the same data allowed for direct comparison between to their CNNs.

The data was separated into (1) a training set—to train the EBM, (2) a validation set—to examine how the model performed and to inform any changes to be made to the feature functions and the intercept, and (3) a test set—to test the final model on unseen data. These datasets were comprised of "scenes," each scene being 64 x 64 (4,096) pixels. In total, we used 7,786 scenes—6,033 for training, 902 for validation, and 851 for testing.

These data were collected in three separate time periods over the course of three years over the central and eastern parts of the contiguous United States (CONUS). The training set spans data from May to August of 2019, the validation set from May to August of 2020, and the test set from May of 2021. Given these data were collected only during the northern hemisphere's summer months, one limitation of the proposed algorithm is that it has not seen data from any other time period and thus performance may suffer if it encounters such data. We note, however, that the summer months were selected because convection is a rare event yet much more common in the summer. If other months were selected, we would expect to see fewer convection examples.

Satellite imagery was obtained from GOES-16's ABI, more specifically Channel 2 (0.64 μm), as seen in Fig. 1(a) in its native 0.5 km resolution, and Channel 13 (10.3 μm), as seen in Fig. 1(b) in its native 2 km resolution. Convective flags from data provided by the MRMS system, as seen in Fig. 1(c), were used as a ground truth.

An obvious limitation of using visible imagery is that it is only useful during the day. As such, any visible imagery data taken from a time when the solar zenith angle was over 65° were removed from the dataset to ensure no "night" scenes were included. This threshold, however, allowed for some "edge cases," taken during twilight, to persist.

a. Developing Information-Rich input Features for Convection Detection

EBMs have been designed to provide interpretable ML algorithms for limited tasks, usually tasks that involve scalar values as predictors. Thus, to use this approach for our application, we had to build a bridge, namely we had to first extract scalar features from the imagery that have high information contents regarding the desired task (see Steps 1-3 in Section 1e).

We extracted four features from Channels 2 and 11 of the GOES imagery, using only spatial relationships, not temporal ones. While temporal information from image *sequences* may be useful to detect convection, see Lee et al. (2021a), we decided for this first study to only leverage spatial features. In future work we plan to explore adding temporal features to our approach, for example using [cite the temporal work here that Kristina is looking at (vines)].

Below we define the features we extracted. For each feature we also discuss how we expect it to relate to convection (e.g., positive / negative correlation) based on meteorological knowledge about convection. These relationships can later be compared to the relationships learned by the EBM model (Section b3), and can be used to correct the latter if needed.

1) Brightness Feature

The first feature, x_1, was derived from channel 2 imagery. Channel 2, also known as the "Red" band, provides high-resolution visible imagery. Visible imagery provides a measure of the intensity of the light being reflected back to the satellite by the clouds and the surface being imaged. When viewed as a collection of pixels, i.e., as a "scene" (such as those we have presented), textural information can be observed. For this first feature, we decided to remove this textural information

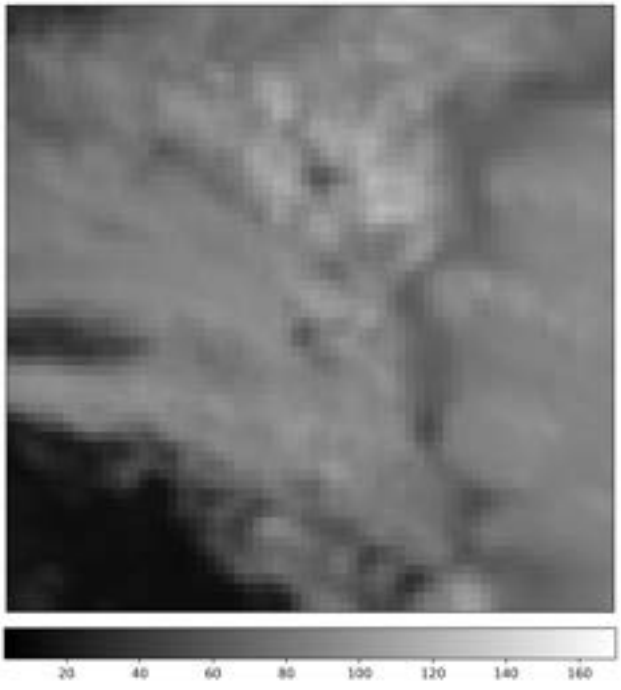


FIG. 4. Brightness Feature Example

by using a 9 x 9 box blur to instead get a measure of the overall brightness in larger regions. Thus, going forward, this feature will be referred to as the "Brightness" feature. A scene of the Brightness feature from our testing set can be seen in Fig. 4. Additionally, these data were resized from their original resolution, 0.5 km, down to a more coarse resolution, 2 km, to match the resolution of the subsequent three features. Nearest neighbor interpolation was used for this process.

Expected relationship to convection: Given this feature was intended to capture brightness, we had expectations surrounding how the model would handle it based on knowledge of the system. As noted by Mecikalski and Bedka (2006), cumulus clouds are often the brightest of the clouds observed in visible imagery, in part due to the fact that they reflect and scatter more light than other cloud types, i.e. they have high optical depth. Using this as a general principle, we inferred that the model would positively correlate high cloud brightness and the presence of convection.

2) WARM & COOL CONTRAST TILE FEATURES

With advancements in the spatial and temporal resolution of data produced by state-of-the-art satellite systems, such as the GOES system, the use of visible imagery to identify texture as a proxy for convection is a logical step. Thus, our next two features, x_2 and x_3, extract texture

information from the high-resolution visible imagery described previously. To do this, we have utilized the mathematical framework of Gray-Level Co-occurrence Matrices (GLCMs), utilizing techniques described more fully in Moen (2024). Following choices described in Moen (2024), we extracted the Contrast statistic—which measures the local variations within the GLCM tiles—for every scene within the visible imagery dataset. For every 4 x 4 pixel grid within the visible imagery, one GLCM tile was calculated, meaning the resolution decreased by a factor of four. This gave these two features a resolution of 2 km each, matching the Brightness feature.

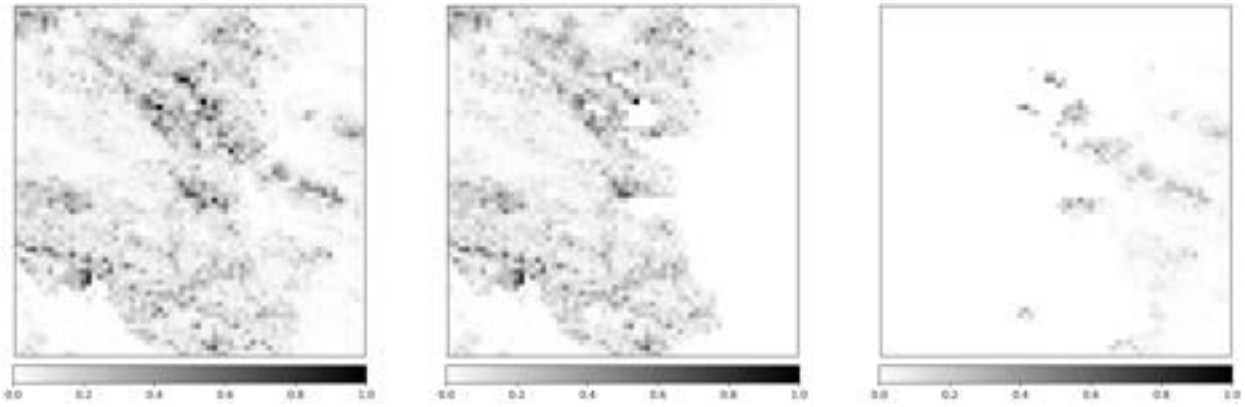
After a grid of "Contrast Tiles" had been calculated for each scene, the two distinct features were derived using cloud-top temperature information provided by channel 13. The first feature, x_2, known as the "Warm Contrast Tiles" feature, was created by retaining the contrast tiles that corresponded to pixels strictly above 250 K. The remaining contrast tiles, those that corresponded to temperatures at or below 250 K, were used to create the third feature, x_3, the "Cool Contrast Tiles" feature. For each feature, the values that were "removed" via the threshold were set to zero.

Similarly to Lee et al. (2021a), we chose a very generous and very warm temperature at which to threshold. As they discussed, many products utilize a much colder threshold to aid in the detection of deep convection. Because EBMs allow for relationships to be learned through the training process, we did not feel the need to set this decision boundary particularly high and instead allowed for the model to learn for itself where to draw the line. We did, however, still feel the need to include the threshold to coerce the relationship to align with our desired goal.

An example of the described thresholding process is shown in Figure 5.

One final alteration was made to these features. It can recognized that contrast tiles that represent a region with a"large" amount of contrast will take on correspondingly "large" values. There is an analogous relationship for regions of little contrast. This becomes a problem when we recognize that the contrast tiles do not inherently have an upper bound (though they do have a lower bound—zero). Because of this, we divided each contrast value by the largest contrast value, in effect forcing the values to exist between zero and one. Thus, regions that have the most contrast—or, for our purpose, we use "contrast" as a measure for "texture"—have corresponding values of one and regions with no texture have values of zero.

Of primary interest to us were the Cool Contrast Tiles. Because we expected for convection to occur at temperatures less than 250 K—and, as we will see, removed the possibility of convection



(a) (b) (c)

FIG. 5. Contrast Tiles threshold progression, starting with the calculation of the Contrast Tiles on the full image (a), the Warm Contrast Tiles themselves (b) and the Cool Contrast Tiles themselves (c). Scenes (b) and (c) represent the features used in the final model.

occurring above 250 K—we heavily relied on this feature to inform us of where the overshooting tops were—more generally where the cloud-top surfaces were most "bumpy," and, typically, regions of deep convection. The Cool Contrast Tiles, then, can be thought of as a conduit to "measure" convection. Though not our primary concern, the Warm Contrast Tiles were not removed as a feature to ensure no potentially valuable information was lost. From Nathan: this reasoning is shaky at best.... Fom Imme: LOL.

Expected relationship to convection: For the Warm Contrast Tiles, we expected a negative correlation with convection. Thus, we expected low values, close to zero, to indicate the presence of convective regions. For values larger than zero we expected to indicate the presence of non-convective regions. For the Cool Contrast Tiles, we expected a positive correlation. Namely, we expected areas with little texture, i.e., those corresponding to flatter clouds, to be associated with the presence of non-convection and for the bumpier, more textured regions to be associated with the overshooting tops and, thus, presence of convection.

3) INFRARED IMAGE FEATURE

The final feature, x_4, was derived from Channel 13, the "Clean" IR Longwave Window Band. Though we built information provided by this feature into the Cool and Warm Contrast Tiles

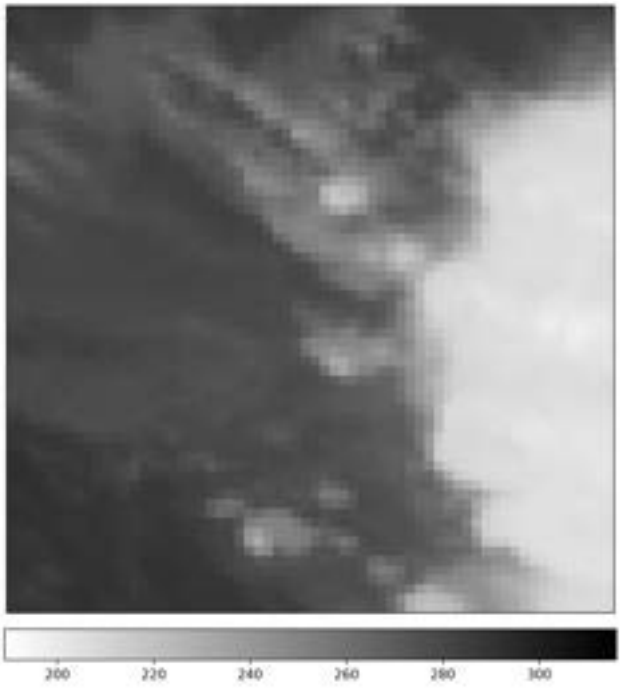


FIG. 6. Infrared Imagery Feature Example, taken 5 May 2021.

features, we additionally use the original data as a separate feature to provide more temperature information than just a simple threshold.

Infrared imagery provides information related to surface and cloud-top temperature by measuring the intensity of emitted radiation and converting it into a temperature. We consider temperature in Kelvin. A scene of the Infrared Imagery feature can be seen in Figure 6.

Expected relationship to convection: We can estimate the relationship between cloud-top temperature and convection by discussing the relationship between atmospheric temperature and cloud growth. We observe that, as elevation into the troposphere increases, temperature decreases. When temperature starts to stagnate, an area designated as the tropopause has been reached. Clouds will continue to grow until they reach the tropopause. At the tropopause, they will start to expand horizontally, forming the familiar anvil shape. But, because of strong updrafts, it is possible for clouds to grow past the tropopause. Such formations are known as overshooting tops. Thus, we expected a negative correlation between temperature and convection. The coldest observed temperatures, those corresponding to areas around the tropopause, were expected to be home to convection. As temperature increases, we expected to observe fewer and fewer convective cells.

b. Multi-Radar Multi-Sensor (MRMS) Data as Ground Truth

We framed the problem of convective detection to be one of binary classification for each pixel —any given pixel was classified as having/not having sufficient evidence for presence of convection. In order to get an estimate of which pixels within each scene could be classified as convective, we utilized data provided by the MRMS system.

The MRMS system provides a Precipitation Flag that aims to automatically classify surface precipitation into seven distinct categories: (1) warm stratiform rain, (2) cool stratiform rain, (3) convective rain, (4) tropical–stratiform rain mix, (5) tropical–convective rain mix, (6) hail, and (7) snow (Zhang et al., 2016). For the purposes of this study, these categories were combined to create two classes that we used as ground truth. This simplification involved combining categories (3), (4), (5), and (6) from the Precipitation Flag into one "convective" flag and the remaining categories into a "non-convective" flag (Lee et al. 2021a).

The MRMS system provides data at a high spatial resolution of 1 km. Thus, each pixel within the 128 x 128 grid was assigned a value of either 1 if the pixel was previously assigned the convective flag or zero if it had been assigned the non-convective flag. After this conversion, to match the 2 km resolution present within the predictors, the resolution of these data were halved. Nearest neighbor interpolation was used for this process.

In addition to the change in resolution, one final modification was made to the MRMS dataset. Similarly to the Warm and Cool Contrast Tiles features, a threshold of 250 K was imparted upon the MRMS data. All convective flags spatially corresponding to cloud tops that were found to be above 250 K were set to 0, leaving the final dataset used as a ground truth to contain only convective flags pertaining to regions in which the cloud tops were at or below 250 K. In doing this, we removed many instances of small, isolated clouds that featured convection. The developed anvils that featured mature convection, which were of primary interest for this application, were left behind.

Figure 7 shows visible imagery (i.e. the original scene) with the corresponding ground truth tiles overlaid. The tiles at full saturation correspond to the detected convection that was found to occur in a region of the scene that was below 250 K, while the more transparent tiles correspond to the detected convection that was found to occur in a region of the scene that was above 250 K. As mentioned, only the "cool" tiles were used to train the EBM. The "warm" tiles are included

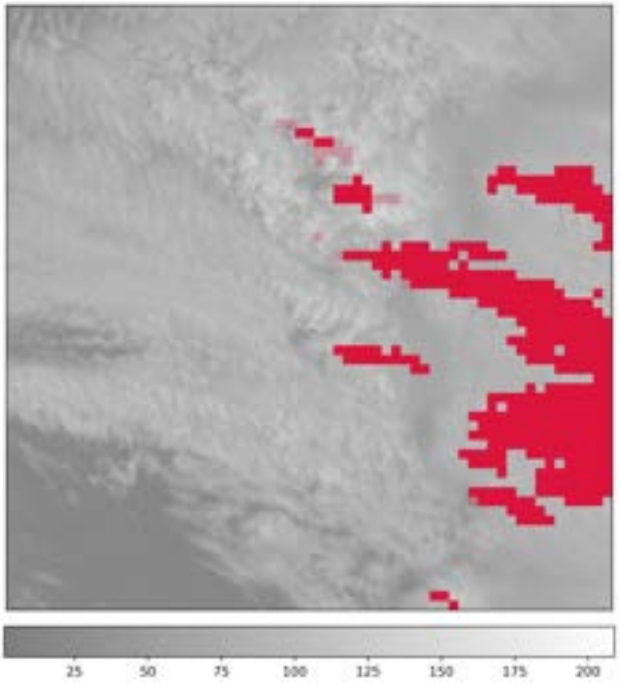


FIG. 7. MRMS "Ground Truth" Example, taken from 5 May 2021.

for reference purposes only. Additionally, the original scene is shown only for spatial comparison purposes.

With respect to the MRMS system's convective flags, we found that, within the training dataset, 512,710 pixels of the total 24,711,168 pixels—which, for reference, is roughly 2.075%—corresponded to pixels that were marked as being convective. Because of this imbalance, we utilized the Adaptive Synthetic Sampling (ADASYN) technique to over-sample the data to increase this percentage, but only just slightly. After oversampling, our final training dataset had a convective flag percentage very close to 3%.

3. Methods

We approached the detection of deep convection as an image-to-image translation problem. We consider how EBMs can be utilized to predict deep convective regions in the upper troposphere using data provided by GOES-16's ABI and how interpretable machine learning algorithms are able to be improved upon when domain knowledge can be incorporated into the architecture of the model.

a. Structural form of an EBM

As mentioned in Section 1f, EBMs are functionally similar to Generalized Additive Models (GAMs). In fact, this relationship goes deeper than a mere similarity. EBMs were built upon the GAM framework and were designed to be an improvement upon them by using modern machine learning techniques to learn the optimal strategies. Another way in which EBMs were able to achieve a boost in accuracy from their GAM predecessors was via the addition of automatically detected and included pairwise interaction terms (Lou et al. 2013)(Lou et al. 2012). The increased complexity of these models from GAMs does not impact their interpretable nature.

In general, EBMs take the form:

$$g(E[y]) = \beta_0 + \sum f_i(x_i) + \sum_{i \neq j} f_{ij}(x_i, x_j)$$

Here, each $f_j(x_j)$ represents a main effect feature function. Each $f_{i,j}(x_i, x_j)$, then, represents a feature function for each of the automatically detected and included pairwise interaction terms. Furthermore, β_0 represents the model's learned intercept. Finally, g represents the chosen link function, which allows for tasks such as regression or classification to be performed (Nori et al. 2019).

b. Feature functions

The bulk of this research dealt with feature functions and how they can be altered. As such, the key to understanding our methodology comes from an understanding of how these functions work. We will highlight the aspects we find to be relevant to their understanding below.

1) Feature functions, a detailed explanation

First, we note that EBMs slowly learn one feature function at a time—to ensure they are independent—for each included time. For our purposes, a "term" could be one of two different possibilities: one of the (potentially) many dimensions of the feature space (a "main effect" term) or an interaction between two dimensions (an "interaction" term). While the EBM framework supports the addition of interactions between more than two features, such interactions are not automatically detected nor included within the model. Because of this, and to ensure we maintained our desired level of interpretability, we opted to not include any interactions between more than

two features. Additionally, we note that that no modifications were made to the training algorithm used, though many modifications are possible.

Once a feature function has been learned, they function similarly to any other one- or two-dimensional function. Like any other function, feature functions transform inputs into outputs. The inputs are the values of the feature(s) of interest and the outputs, which, in this context, are called "scores," can be thought of as a way to measure the relationship the input has to the outcome of interest. Higher score values indicate a strong association while lower score values indicate a weaker association. Scores of exactly zero indicate a lack of an association. In our binary classification convection detection case, positive scores indicate that the input is associated with the presence of the event class, convection, and negative scores indicate that the input is associated with the presence of the non-event class, non-convection. Though this general description fits for both the main effect and interaction feature functions, further nuance is required to distinguish between the two.

Each main effect feature function is realized similarly to a traditional one-dimensional function. More specifically, however, they are *step* functions. For a categorical feature, the feature function would map each unique level of the feature to a unique score. The same general approach is taken for continuous features, though not every unique value of each feature is mapped to a score. Instead, the values of each continuous feature are discretized into "bins" that, together, describe the range of the feature. Each bin, then, is mapped to a unique score.

A similar approach, but extended into two dimensions, is taken for the interaction feature functions. Instead of each input being one value of the feature of interest, inputs are (x_i, y_i) *pairs*. The x value comes from the first feature included in the interaction and the y value from the second. The output, each z value, is the score. If there are n bins for the first feature and m bins for the second, there are nm total score values. In practice, however, we have observed that each feature that is part of an interaction is broken up into the same number of bins (i.e. there are n^2 scores). To keep interpretation simple, interaction feature functions are realized as heat maps, the third dimension provided by the scores being plotted as a range of color values.

Finally, we tie up one loose end. It may be the case that the training data does not encapsulate all possible values any given feature may take on in unseen data. The feature functions account for

this by adding in an additional bin that is designated role is to deal with unseen data. Additionally, feature functions are able to deal with "missing" data with an additional bin.

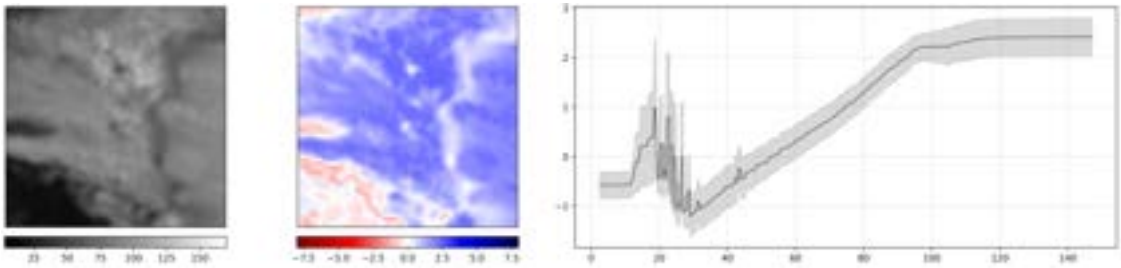
2) Additivity & Interpretability of EBMs

One defining feature of EBMs is their inherent interpretability. This interpretability, and, more specifically, what the developers of EBMs refer to as "intelligibility," is a direct result of their feature functions. EBMs are able to provide information about the independent, modular relationships they have found within the data though these functions. Furthermore, it can be easily seen how each feature contributes toward the final prediction. The feature functions are able to provide such information because, much like their GAM predecessors, they are additive in nature. Each feature function, each f_j and $f_{i,j}$, is *trained* independently, meaning each can be *interpreted* independently from the others.

We note here that other EBM implementations need not be as interpretable as the one we have built. In fact, EBMs support the use of many more than just four main effect features and four interaction terms. The inclusion of additional features may improve performance, though perfect performance was not our goal. Instead, we sought to build a model with a small enough number of terms such that each of their contributions to the final prediction could be analyzed and potentially modified. This EBM could have utilized dozens of features to improve performance as other implementations have done (Caruana et al. 2015), but, as the number of terms increases, the interpretability of the full model dwindles.

Overall interpretation decreases as the number of features increases because feature functions are best understood when visualized. Unlike other functions that may have a "closed form" expression that could be reasoned about, feature functions are a collection of two sets of values, the bins and their corresponding scores. Thus, the best way, though not *necessarily* the only way, to understand these functions is to plot them. Through this visualization, a more acute sense of the model's decision-making processes can be seen.

Figure 8 illustrates these processes for a main effect feature in a convection detection example. Fig.8(a) shows our Brightness feature, seen also in Fig. 4. Jumping to the right-hand side, Fig. 8(c) shows the function that was learned for the Brightness feature. The dark line in the center represents the learned function while the semi-transparent shaded regions above and below represent an error



(a) (b) (c)

FIG. 8. (a) Brightness, (b) Feature Importance, and (c) a Feature Function with shaded error bars to represent a measure of uncertainty.

estimate calculated during the training process. Here, the values seen on the x-axis are governed by the levels found within the Brightness feature. Though this function appears to be continuous, it is actually the culmination of over 1,000 discrete bins. Each bin is mapped to a score, which, here, represents how levels of brightness are associated to the presence of convection.

Something important to note about the realization of these feature functions is that, in this application, the x values represent the *observed* brightness values as they existed in the training dataset—roughly from 0 to 150. To achieve this, no alterations were made to the brightness values the EBM was trained on. Thus, for EBMs to function, it can be observed that, unlike other common prediction-based machine learning workflows, the data need *not* be normalized.

Finally, Fig. 8(b) can be thought of as a way to combine the scene itself, Fig. 8(a), and the associated feature function, Fig. 8(c), into one, concise plot. Visually, Fig. 8(b) *resembles* plot Fig. 8(a). Despite this, it is not just a simple recoloring of the scene. Instead, the feature function was "applied" to the brightness values of the scene to achieve a set of corresponding scores. These scores were then plotted. The "red" color indicates the presence of the non-event class, non-convection, and the "blue" color indicates the presence of the event class, convection. The darker the value of the color, the more the corresponding pixels signal their respective class, and, the lighter the value of color, the weaker that signal is. It follows, then, that the "white" shade indicates the associated pixels do not imply the presence of either class. Simply put, plots such as those in Fig. 8(b) can be thought of as a way to visually examine the "importance" of certain values, per scene.

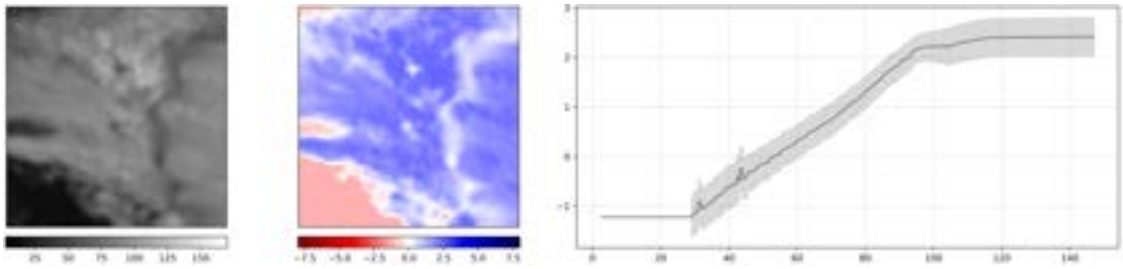
The ability to visually examine what the model has determined to be important toward predicting both the event and non-event classes is crucial for feature function interpretation and, toward our overall goal, altering the undesired strategies the model learned.

3) Feature Function Alteration

Once the feature functions have been trained and visualized, the next step in the EBM workflow is to examine the learned relationships, specifically noting any relationships that do not align with any notions of what they ought to look like. During the training process, there is no sense of a "right" or "wrong" strategy to be learned, only a standard performance metric to be optimized. Thus, depending on how clear the signal is between the features and the output, the number of strategies to be altered may vary considerably. If the relationships themselves look appropriate, one possible approach to alteration is to change the weight that one relationship has by scaling the scores of the desired feature function. In fact, this is a built-in functionality and requires little effort to enact.

Scaling, however, was not utilized in this application. Instead, we found that the model learned at least one spurious main effect relationship and felt the need to change it and another to better match what we expect to see by observing the physical characteristics of convective systems. To examine what might motivate an alteration, we can first look once again to Fig. 8, specifically at the Brightness feature itself in the lower left-hand corner. It can be seen that this region is devoid of clouds and, as there are no clouds to reflect back light, is visually dark.

If there are no clouds, there is no possibility of convection. By examining the feature importance plot in (b) in the same region, we observe that the EBM was unable to pick up on this. At some point in the training process, it likely saw evidence that some dark pixels may indicate the presence of convection. Because of this, parts of this region have been colored in as blue, indicating the presence of convection. Based on the knowledge of the physical properties of convective regions, we know that this is incorrect. The model seems to have learned an incorrect strategy for detecting convection. But, because EBMs can be altered after they have been trained, we do not have to accept this decision nor retrain the model in an attempt to coerce the model to learn the correct strategy. Instead, we can alter the *score* values (in the main effect feature function case, the y values) of the Brightness feature function to match what we expect out of the relationship.



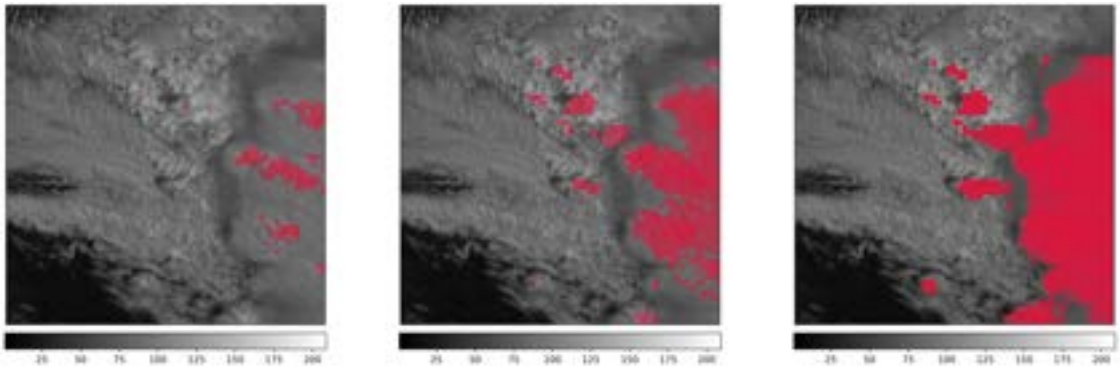
(a) (b) (c)

FIG. 9. (a) Brightness, (b) Feature Importance, corresponding to the Edited Feature Function (c).

We demonstrate what this process might look like in Figure 9, which shows the same scene as in Figure 8, but displays the effects of altering the corresponding feature function. Because we know that small brightness values tend to indicate the presence of the non-event class ('no convection'), we simply flattened out the feature function in this region for all values to the left of the minimum score value. Because the estimated error of the feature function is a product of the original (unedited) feature function, areas in which any feature function has been altered will not display a measure of error. We will continue this practice for all future visualizations of feature functions that have been altered. The effect of altering the feature function can easily be seen in the map of feature importance. Before, in Figure 8, the region in the lower left-hand corner had trace amounts of blue, indicating the potential presence of convective regions. With the feature function flattened out, this entire region has been turned red, indicating the presence of the non-event class ('no convection') instead.

This editing process, however, is not limited in scope to the feature functions. In fact, the intercept of the model can be edited as well. We note here that altering the intercept of the model does not impact the scores of the feature functions in any way. Instead, raising the intercept works to increase the overall score garnered by all the feature functions for each value of each feature and lowering it has the same, but opposite, effect. In a binary classification setting, raising the intercept will lead to the event class being predicted more often on average. Lowing the intercept will lead to the non-event class being predicted more often on average.

An example of this is shown in Figure 10. The red pixels featured within the scenes represent the predictions of three different models, the opacity of these pixels showing the confidence level of these predictions—darker, opaque red pixels indicating a very "sure" prediction and a less opaque, more transparent red pixel indicating a more "unsure" prediction. Of the three predictions shown,



(a) (b) (c)

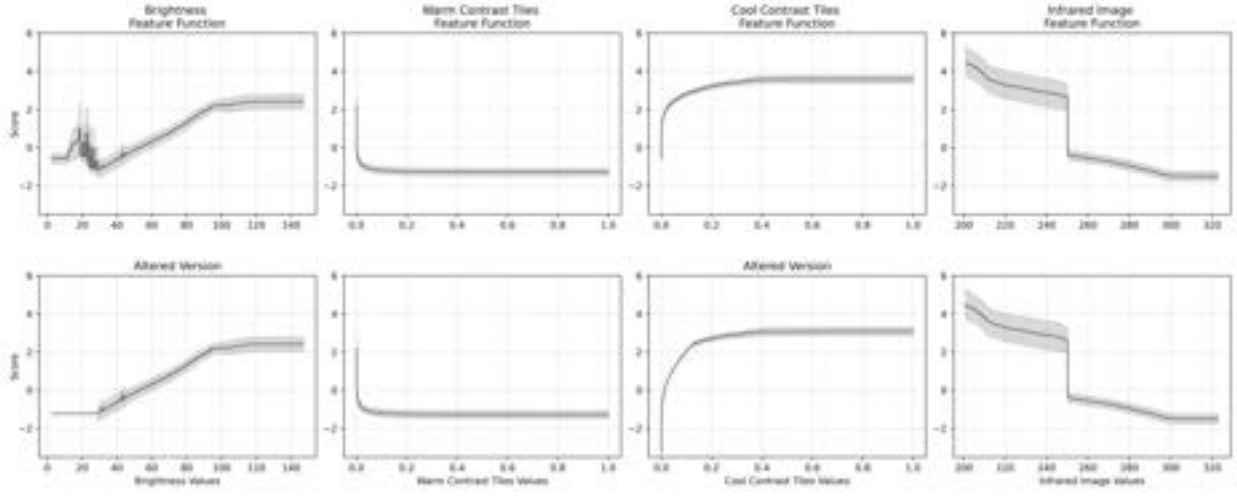
Fig. 10. Predictions of convective regions from the same model but with an increasing intercept. Intercept increased from (a) -21.02 to -18.77 (b) to -16.52 (c). the second lowest intercept, and (c) predictions from a model with

scene (a) represents the predictions of a model with the smallest intercept, scene (c) the predictions from a model with the largest intercept, and scene (b) the predictions of a model with an intercept equidistant between (a) and (c). It should be noted that the feature functions of the models used to generate these predictions were identical and represent the feature functions before they were altered in any way. The only alteration made to this model was made to the intercept.

Aside from the increased number of predictions, increasing the intercept also increased the confidence the model had in its predictions. When the intercept was relatively larger, the model had more "evidence" to use to suggest the presence of convection and was thus more "sure" in its predictions. This effect can be seen best when considering all three scenes in Fig. 10. From (a) to (b) to (c), the opacity of the predicted pixels seen in (a) increased from nearly translucent in (a) to slightly opaque in (b) to fully opaque in (c). It can be said that, as the intercept increases, pixels predicted as convective go from having a probability of near 0.5 of being convective to a probability of near one.

The first row of Figure 11 displays the original main effect feature functions learned by the model.

Of the four feature functions, we only felt the need to alter two of them. These were, as shown in the second row of Figure 11, the Brightness feature's feature function and the Cool Contrast Tiles feature's feature function. Given we have already discussed the changes made to the former and why those changes were made, we will discuss here the changes made to the latter.



(a) (b) (c) (d)

FIG. 11. Unedited and edited feature functions for all main effect features, (a) Brightness, (b) Warm Contrast Tiles, (c) Cool Contrast Tiles, and (d) Infrared Imagery—unedited functions in the first row and edited functions in the second.

The Cool Contrast Tiles feature's feature function, upon first visual inspection, appeared to be exactly what we had expected. As the values of this feature increased, i.e. as there was more texture detected within any given region, the score increased as well. Despite this, it can be observed that the minimum score value of this function occurred around zero, the score only increasing from there. This was a problem.

As we mentioned previously, we chose a very generous threshold of 250 K at which to separate our Contrast Tiles into Warm and Cool. This threshold was set low to allow for the EBM to learn the distinction within the cold regions. Instead, the relationship it learned, based on this function alone, was to predict convection whenever the temperature was below 250 K. Because we do not expect for every cold pixel to be convective, we had some work to do.

The first change we made was to pull the tail of the function down. As can be seen in (c), the minimum score value changed from roughly zero to roughly negative three. This change allowed for pixels that had low contrast values, those close to 0, to be assigned negative score values. Thus, even in a cold region, low texture indicates a lack of convection.

After lowering the tail, the curved region was flattened out. Because of this, values that had texture close to zero, but not quite zero, got assigned small score values. As the amount of texture increased, so too did the score. Visually, this relationship was made to be more "linear" in nature.

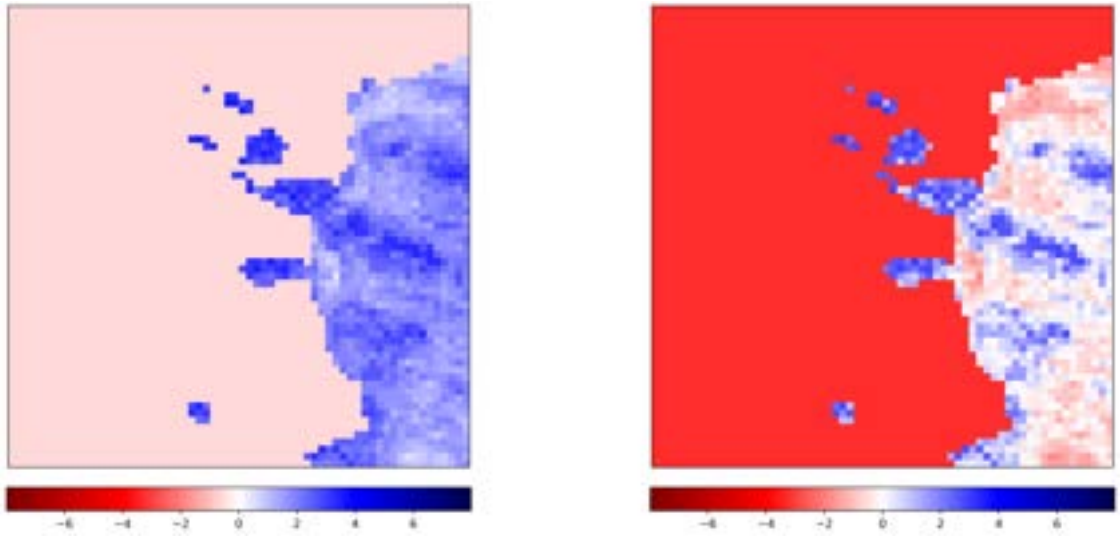


FIG. 12. Feature importance map for the Cool Contrast Tiles (a) before the model was altered and (b) after the model was altered.

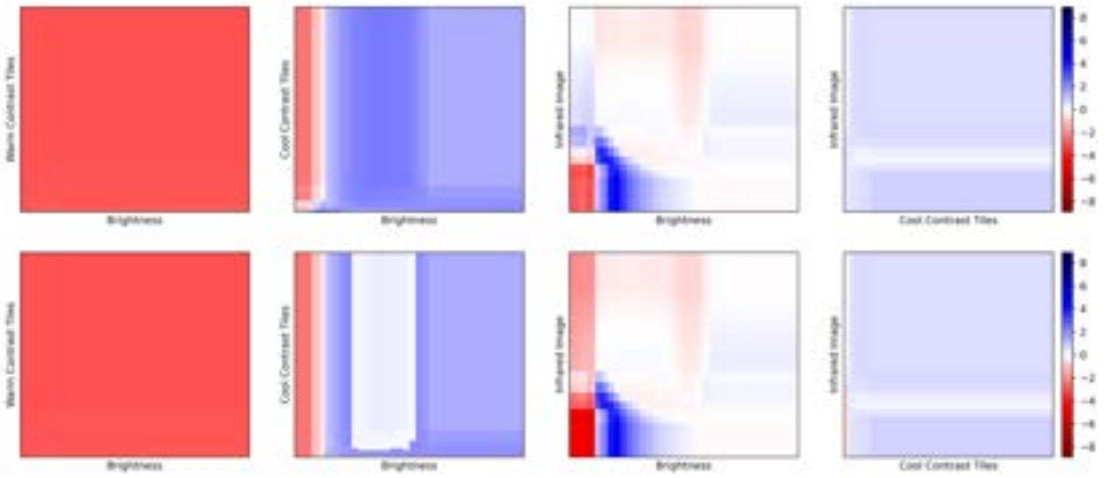
After flattening, the rest of the feature function was left alone and thus has visually retained its error bars.

To get a better sense of what this alteration did to the model, we present in Figure 12 the feature importance maps for the Cool Contrast Tiles from before (a) the function was altered and after (b) the function was altered from the same scene we have been following.

As can be seen, before the function was altered, the entire region of contrast tiles below 250 K were all given a non-negative association with the presence of convection. Only tiles with a value of zero, pixels excluded by the threshold, were given a negative score.

Only after the function had been edited can the distinction between minimal texture and maximal texture be seen. Here, only the regions with the most texture were given large, positive scores. As texture decreased, so too did the assigned score. Much of the cloud top, those regions without much texture, were given score very close to zero. Some of these regions were even given negative scores.

As we will see, the Cool Contrast Tiles feature was very "important" to our model, so we wanted to ensure that it was classifying texture in the ways in which we saw fit which, once again, was only possible because of our utilization of EBMs.



(a) (b) (c) (d)

FIG. 13. Feature functions for the four included interactions—the first row shows the feature functions before alteration and the second row shows their altered counterparts. Only feature functions (b), (c), and (d) were altered.

4) INTERACTION FEATURE FUNCTION ALTERATION

In addition to the four main effect features, four interaction terms were included within the model. These four interactions were automatically determined by the model framework to be the best interactions out of the six possible pairwise interactions between the four main effect features. For more information on the algorithm that carries this process out, visit Lou et al. (2013). Interactions between the following pairs were included in the mode: Brightness & Warm Contrast Tiles, Brightness & Cool Contrast Tiles, Cool Contrast Tiles & Infrared Image, and Brightness & Infrared Image.

The first row of Figure 13 displays the feature functions for each of the included interaction terms before the model was altered. The second row, then, shows the altered interaction feature functions. In accordance with the new form these functions take on, we have listed the name of the fist feature on the x-axis and the name of the second feature on the y-axis. Note that the placement of the feature names is not arbitrary. We made slight alterations to the feature functions in Fig. 13(b), (c), and (d). We will briefly discuss the alterations in this order by displaying the effects each alteration had to their feature importance maps in Figure 14. Fig. 14(a) and (b) show the feature importance maps for the interaction between Brightness and Cool Contrast Tiles, where Fig. 14(a) shows the importance corresponding to the unedited model and Fig. 14(b) shows the

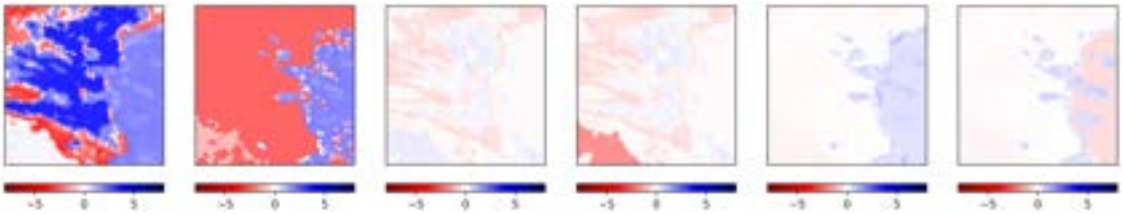


FIG. 14. Feature importance maps for the altered interaction feature functions. Importance maps for (a) unedited and (b) edited feature functions for the interaction between Brightness and Cool Contrast Tiles, (c) unedited and (d) edited feature functions for the interaction between Brightness and Infrared Image, and (e) unedited and (f) edited feature functions for the interaction between Cool Contrast Tiles and Infrared Image.

importance corresponding to the edited model. In editing this feature function, we took note of the incorrect strategy learned for areas that were above 250 K. Such areas were highly associated with the presence of convection (denoted by the dark blue color). We altered the feature function to instead imply a negative association, further emphasizing areas of interst (those below 250 K).

Next, Fig. 14(c) and (d) show the feature importance maps for the interaction between Brightness and the Infrared Image, where Fig. 14(c) shows the importance corresponding to the unedited model and Fig. 14(d) shows the importance corresponding to the edited model. The alterations made to this function were slight as this feature did not have much impact toward the final prediction. For this function, we ensured that areas without clouds received a negative score instead of a positive one as seen in Fig. 14(c).

Finally, Fig. 14(e) and (f) show the feature importance maps for the interaction between the Cool Contrast Tiles and the Infrared Image, where Fig. 14(e) shows the importance corresponding to the unedited model and Fig. 14(f) shows the importance corresponding to the edited model. For this function, we de-emphasized areas that, while cold, did not have much texture. In effect, we were able to highlight the areas within the Cool Contrast Tiles that had the most texture.

5) Feature Function Limitations

Before we discuss the results, there is a limitation inherent to EBMs and their feature functions we wish to discuss. Each feature function, for each main effect and each interaction, is static. They can be altered at any point, but when it comes time to make a prediction, they will give the same prediction for equivalent values, even if, to a human observer, those values do not mean the same

thing when viewed in the broader context of the entire scene. EBMs make pixel-wise predictions, meaning they do not get to observe the greater context as other modern machine learning methods do. Because of the simplicity of this method, we lose out on the ability to model more complex relationships that other methods may easily detect.

A visual example will be given as part of the result section, but we can envision a different example to motivate this idea. Say, for instance, we are observing a region of convection. In this scene, a shadow is cast onto the bubbly region. A human observer would be able to view this region in the context of the entire scene and classify it as being convective, despite the drop in brightness. The EBM, however, may not be able to make the same distinction. Because the feature functions inform the EBM that dark regions are typically associated with regions of non-convection, it may not have enough evidence from the other features to reach the threshold for predicting convection.

If the goal is to classify convection in particularly confusing scenes, an EBM may not be the right tool. With this in mind, we can move into discussing the results.

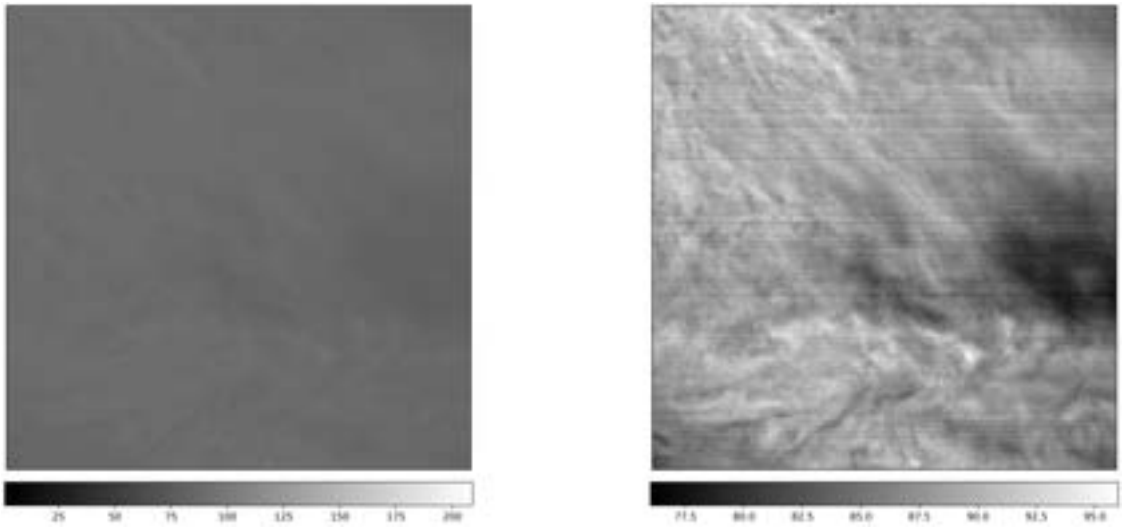
4. Results

We will discuss the results of our model is three parts. First, we will compare the overall performance of the original model to the performance of the edited model on our designated testing set. Next, we will provide case studies where we visually examine the performance of each model on four distinct scenes, each from our testing set. Finally, we will compare the results of our edited model to the results of a CNN on the same testing set. Before we get to the results, however, we must briefly discuss issues pertaining to the testing dataset.

a. Image Striping

There are some known issues regarding the data provided by the ABI aboard GOES-16. One such issue, specifically pertaining to visible imagery, has been referred to as "striping." This issue, caused by differences in how each detector responds to the light it is receiving, can be identified by horizontal stripes that persist across any given scene (Gunshor et al. 2020). As example of what this might look like taken from our testing dataset is given in Figure 15.

Fig. **??**(a) shows what the visible looks like when the color scheme ranges across all reflectance values (as all previous visible imagery scenes have been shown) while Fig. **??**(b) shows the scene



(a) (b)

59 FIG. 15. Visible Imagery example (5 May 2021) from testing set heavily affected by striping, (a) plotted on
60 the full range of reflectance values and (b) on the range seen within the individual scene.

with a color scheme limited in range to the reflectance values of the individual scene to enhance the striping detail that is far less visible in Fig. **??**(a).

This problem was not limited to our testing set. In fact, many scenes from both the training and validation sets contained scenes that featured striping. We found that, in most cases, if the model was found to be massively over-predicting (in some cases, the model predicted convection for nearly every pixel within the scene), striping was cause of such behavior. Because of the vast number of scenes featuring striping and the lack of a reliable algorithm available to us that could detect/remove said striping, we opted to leave such cases in the training and validation datasets. To ensure fair and accurate testing set results, however, we empirically removed scenes from the testing set that featured visible striping worse than the worst striping seen in the validation dataset. In total, 150 scenes were removed from the testing dataset. This left us with 701 scenes used for testing instead of the original 851 as outlined previously.

From Nathan: I'm considering adding in the prediction the model made from this scene... thoughts? Include the feature importance of the cool contrast tiles for reinforcement? Not strictly necessary but may be interesting to see. Let me know.

Metric	Definition
Probability of Detection (POD)	$\frac{\text{hits}}{\text{hits + misses}}$
False Alarm Ratio (FAR)	$\frac{\text{false alarms}}{\text{false alarms + hits}}$
Success Ratio (SR)	$\frac{\text{hits}}{\text{hits + false alarms}}$
Critical Success Index (CSI)	$\frac{\text{hits}}{\text{hits + false alarms + misses}}$

FIG. 16. All relevant metrics and their associated definitions.

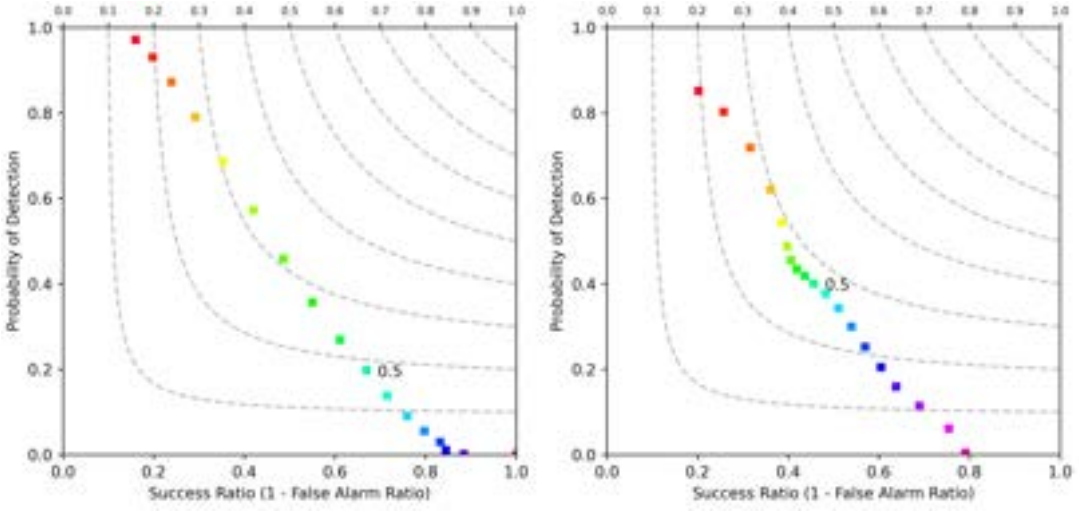
b. Part One: Overall Model Results

To get an overall sense of how well the EBM was able to perform on the testing set, we will examine standard performance-based metrics. First, we will examine the probability of detection (POD). Then, the false alarm ratio (FAR) and the success ratio (SR). We also briefly consider the critical success index (CSI). The corresponding formula for each of these metrics is given in Table 16.

To start, however, we will go back to the SR and the POD. Figure 17 shows these two metrics plotted against one another for (a) the unaltered model and (b) the altered model.

As discussed, the EBM framework gives a probability associated with each prediction of the event class, each probability corresponding to how "sure" the model was that is had correctly predicted any given pixel as belonging to the event class. Probabilities closer to one indicate the model is very sure it made the correct decision, and it follows that values closer to zero indicate the model was less sure in its prediction. For validating the model and in all subsequent predictions, we used a probability of 0.5 as our threshold—pixels predicted as convective counting only if the model was at least 50% sure in the prediction of that pixel.

Though we believe that a probability of 0.5 is a natural choice for a threshold, because EBMs can be altered down to their intercept, the choice of this threshold is largely arbitrary. If the threshold is increased, the feature functions and intercept can be scaled down. An analogous transformation could be made for a decreased threshold. From Figure 17, we note that our unaltered model had an SR of 0.669 and a POD of 0.198 and our altered model had an SR of 0.456 and a POD of 0.401. On their own, these metrics do not indicate stellar model performance. What we wish to note, however, is that, through the process of editing the model, we decreased the SR, which



61 FIG. 17. Performance diagrams (1 - FAR vs. POD) for (a) the unaltered EBM and (b) the altered EBM for
62 threshold values ranging from 0.05 to 0.95. Presented results were made at the marked threshold, 0.5.

measures the fraction of the hits to the total number of event-class predictions, but increased the POD, which measures the fraction of false alarms to the total number of event-class predictions. Roughly, Figure 17 tells us is that we increased the number of false alarms more than we did hits but decreased the number of misses.

Something interesting to note about the number of false alarms our model garnered is that 24 of the 222 scenes that predicted the presence of convection were the cause of roughly 45.10% of all false alarms. Each of these scenes featured more than 250 false alarms. These 24 cases can be broken down into three reasons why there was over-prediction. First, five of the scenes featured some striping that did not meet the chosen visual threshold. Next, a problem we discuss further in Case Study III, 17 of the scenes featured "confusing" texture (i.e. strong texture signals that did not necessarily indicate the presence of convection). Finally, two scenes featured both of these issues.

Furthermore, some of the scenes that featured numerous false alarms were part of "sequences" of scenes. In such cases, the scene did not change from scene to scene, but the feature that contributed to the over-prediction (such as a confusing texture) prevailed throughout the scenes, adding numerous false alarms to the total. We note here that this discussion of poorly performing

scenes was simply to give insight into what caused many of the false alarms that were seen. These scenes were not removed from the testing dataset.

c. Part Two: Case Studies

Next, we will consider four case studies. These examples can be broken down into two distinct categories. The first two studies will consider scenes in which the model performed well and the second two where the model did not perform well. Case I will consist of two scenes where, to our standards, the model accurately predicted the presence of convection well. Case II will consist of a scene were the model accurately predicted the presence of non-convection. Case III will consist of two scenes where the model over-predicted the presence of convection. Finally, Case IV will consist of one scene where the model under-predicted the presence of convection.

We believe that the visual examination of the model's performance leads to a better and more full understanding of the model's performance and work to further contextualize the discussed metrics. To further our point, we will list the relevant metrics pertaining to these individual scenes.

From Nathan: will need to actually reference/link to the appendix at some point.

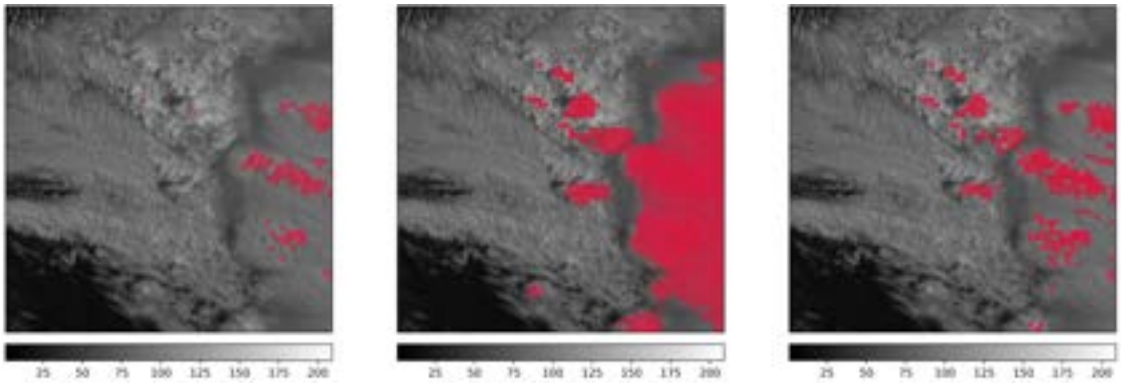
1) Case I

The first scene we present is the same scene we have been following throughout. Because of this, we will take advantage of the reader's familiarity with the scene to further contextualize something we made a brief mention to earlier. Throughout our discussion, we have made reference to how we altered the feature functions of our model as this was one of the primary areas of interest. A keen reader may remember, however, that the intercept of the model was altered as well. The original intercept of our model was -21.0168. To alter this, we added 2.25 to this value, leaving us with a final intercept of -18.7668.

From Nathan: do we need to mention somewhere that we won't be doing this for the other cases? Is that unnecessary?

A naive approach to editing an EBM may consist of simply altering the intercept to increase the number of predictions. We show in Figure 18 the consequences of such an action on our model.

The first scene, (a), shows the predictions of the EBM we trained before any alterations were made. We note here that there were very few predictions made and that the predictions that were



(a) (b) (c)

63 FIG. 18. (a) Predictions of the original, unedited model, (b) predictions made when only the intercept was
64 altered, and (c) predictions made when the intercept and the feature functions were altered.

made were not made with high probability (as noted by the low opacity). Next, (b) shows the effects of altering the intercept only. For this visualization, we altered the intercept by an addition of 2.25, the same value chosen for the final model. We note here that the model was more confident in the predictions it made in the areas in which we expected for convection to occur and less confident in the areas we wished it had not predicted convection. Finally, (c) shows the effect of altering the feature functions. We note that we were able to keep/discard the respective regions of interest while maintaining the confidence level by altering the feature functions.

Here, we make the case that, depending on the model, simply altering the intercept may not be enough to ensure accurate predictions for any given task. The "shape" the predictions take, as can be seen in Figure 18, may instead depend on the specifics of the feature functions.

Next, we can examine Figure 19. We note here that all subsequent examples will follow this same format.

Figure 19 gives a brief visual overview of the relevant information pertaining to this scene. In order from left to right, we have visualized the original visible imagery (a), the suggested ground truth given by MRMS data (b), the four derived features (c), and, finally, a map of the predicted convection (d).

In this scene, and in all subsequently presented scenes, a total of 4,096 predictions were made—one for each pixel within the 64 x 64 pixel grid. Each pixel predicted as being convective met the 0.50 threshold after the model had been altered. The opacity of the predicted convection,

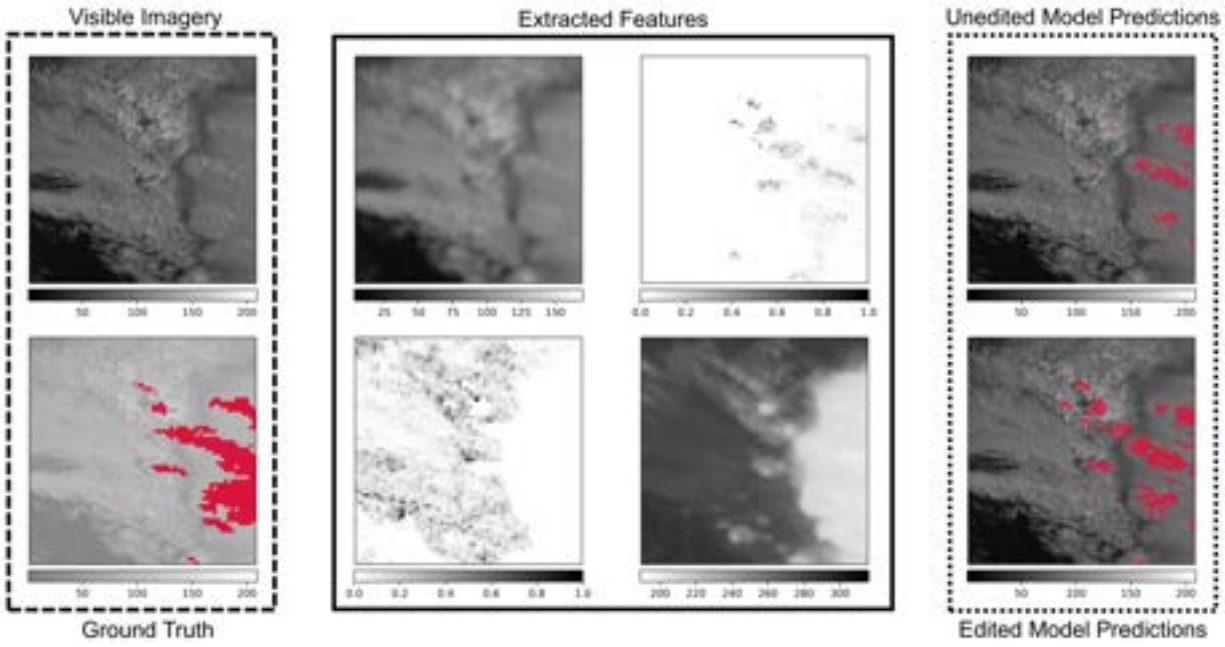


FIG. 19. EBM workflow stages for selected scene from the testing dataset. (a) Visible imagery and Ground Truth (5 May 2021) shown in, (b) Extracted Features, and (c) the performance of the Unedited and Edited models.

those tiles colored in red, represents the probability each prediction is actually convective—as the pixels become more opaque, the probability increases.

Before the model had been altered, this scene had an SR of 0.772 and a POD of 0.225. Out of the 4,096 predictions made, this scene featured 95 hits, 3,646 correct rejections, 28 false alarms, and 327 misses. After the model had been edited, this scene had an SR of 0.570 and a POD of 0.396. Breaking it down further, there were 163 hits, 3,551 correct rejections, 123 false alarms, and 259 misses. Altering the model resulted in an increase in the total number of false alarms, a fact that can be seen both in the metrics and by observing Figure 19. Despite this increase in false alarms, our altered model tended to predict convection reasonably. Though these may not be the metrics of a desirable convection-resolving model, we believe that visual examination provides a better account of model performance.

By comparing the original scene and the predicted convection, it can be seen that, despite the fact that the predictions do not numerically align with the MRMS data, they do align with where the most texture is featured within the visible imagery. In fact, many of the model's misses—particularly on

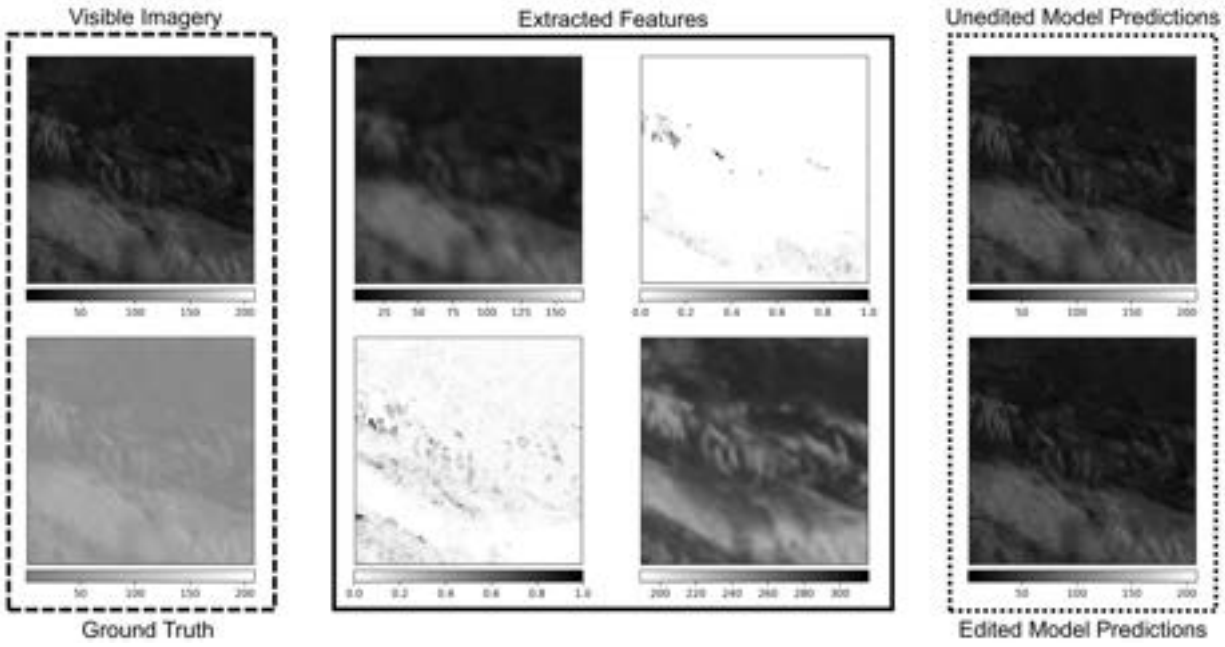


FIG. 20. EBM workflow stages for selected scene from the testing dataset. (a) Visible imagery and Ground Truth (5 May 2021) shown in, (b) Extracted Features, and (c) the performance of the Unedited and Edited models.

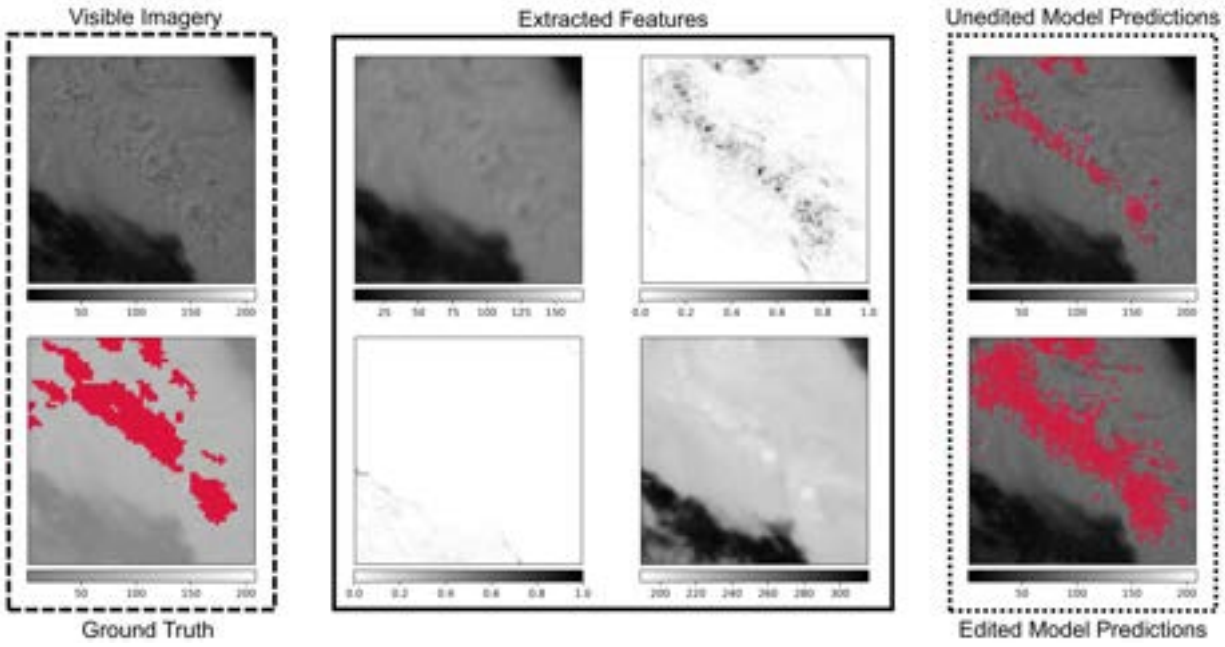
the lower right side of the image—came from regions within the visible imagery that are relatively flat. From this example, it can be clearly seen that the model was able to successfully pick up on the overshooting tops by identifying regions with the most texture. It is also worth noting that the model was able to ignore some of the brightest and most textured regions within the image because of the distinction it learned between regions above and below 250 K.

2) CASE II

The next example features a scene where the ground truth suggested no convection was present and the altered model was able to accurately predict this. Once again, more information regarding this scene and how the model viewed it can be found in the appendix.

Figure 20 gives a brief visual overview of the relevant information pertaining to this case. The information is presented in the same order as it was in Case I.

Here, there is no difference in predictions between the original model and the altered model. Each version was able to accurately predict a lack of convection. Because the model was able to



71 FIG. 21. EBM workflow stages for selected scene from the testing dataset. (a) Visible imagery and Ground
72 Truth (5 May 2021) shown in, (b) Extracted Features, and (c) the performance of the Unedited and Edited models.
73

make a correct prediction on every pixel within the grid, the SR and the POD do not make sense in this context, however We note that the number of correct rejections was 4,096 while all other metrics (the hits, misses, and false alarms) were all 0 for each version of the model.

By examining the Cool Contrast Tiles feature in (b), we note that this case was not necessarily trivial. If all pixels within the grid corresponded to temperatures above 250 K then predicting non-convection would have been the clear choice. Within this scene, however, it can be seen that there exist temperatures below 250 K and that within such regions there exists texture. This, in conjunction with the bright nature of these cold regions, as suggested in the Brightness feature, all indicate the presence of convection. The model, however, was able to accurately predict that no convection was present within this scene.

3) CASE III

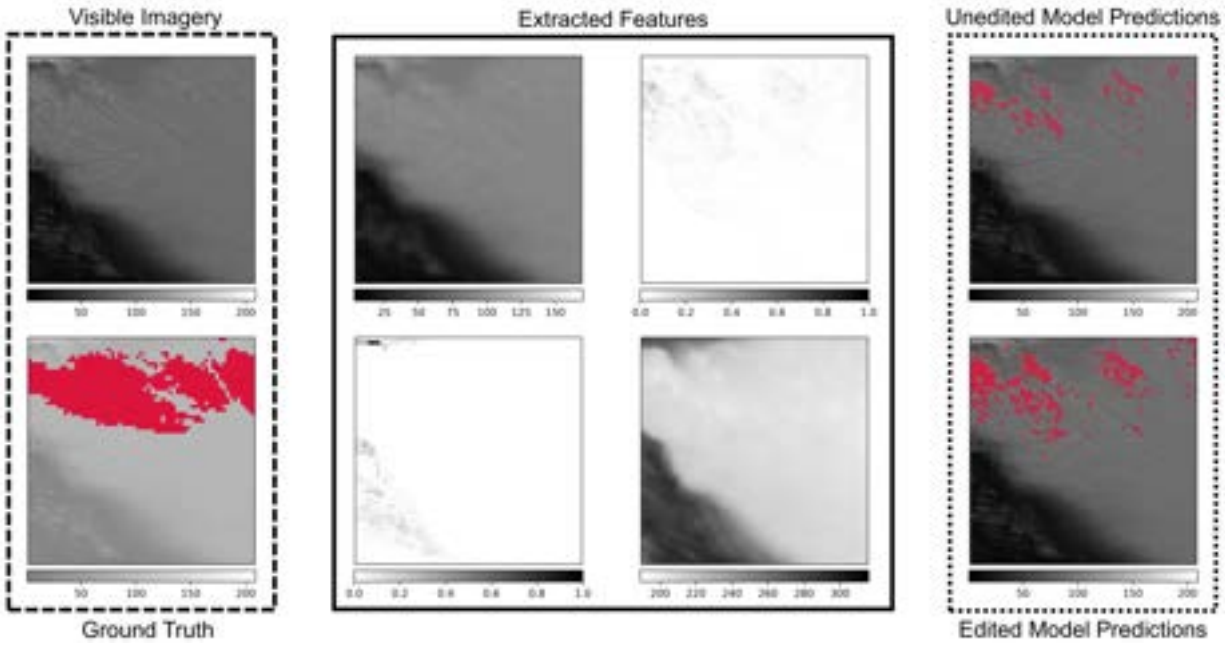
Next, we move into the "bad" examples. First, we depict a scene where the model over-predicted the presence of convection.

Figure 21 gives a brief visual overview of the relevant information pertaining to this case. The information is presented in the same order as it was in both Case I and Case II. Once again, more information about this scene can be found in the appendix.

Because we have predicted convection, we can once again look at the metrics for this specific scene. Before alteration, this scene had an SR of 0.613 and a POD of 0.278. There were 195 hits, 3,271 correct rejections, 123 false alarms, and 507 misses. Interestingly, the model performed well on this scene before it was altered. We note here, however, that it was not very confident in the correct predictions that it made. After the model was altered, there was an SR of 0.436 and a POD of 0.688. There were 483 hits, 2,769 correct rejections, 625 false alarms, and 219 misses. By increasing the number of event-class predictions, both hits and false alarms, we were able to lower the number of misses, but did so, of course, at the cost of more false alarms than hits.

We can examine what went wrong in this scene by observing both the visible imagery in (a) and the Cool Contrast Tiles. First, we note that, by looking at the original scene, a domain expert would easily be able to identify the difference between the overshooting tops jutting out of the cloud top and the surrounding "ripples." But, with the features we derived, the model had a much harder time deciphering the subtle differences. We note this as being another limitation of our model. We have been treating the Cool Contrast Tiles as a way to single out overshooting tops based on their bubbly texture, but the actual mathematical underpinning is not necessarily able to distinguish between bubbly texture and rippled texture as, roughly, it simply measures contrast. Because of this, it has been observed that the model will often predict convection in areas that have an abundance of texture, even if that texture is not, by our standards, the "right" texture.

Further examination of the Cool Contrast Tiles feature, however, especially when examining the opacity of the tiles, allows for a slightly more nuanced approach to this discussion. The ripples present within the scene were observed as being texture, but, as the opacity of these pixels is lower than the opacity of the pixels in the more "bubbly" regions, they have "less" texture according to the Contrast Tiles. The model, however, still took this information, as well as information from the remaining main effects and interactions, and predicted these rippled regions as being convective, leading to over-prediction. Future work in this area may examine further tuning the Cool Contrast Tiles feature function to more precisely differentiate between bubbly texture and rippled texture.



74 FIG. 22. EBM workflow stages for selected scene from the testing dataset. (a) Visible imagery and Ground
75 Truth (5 May 2021) shown in, (b) Extracted Features, and (c) the performance of the Unedited and Edited models.
76

At this time, however, we erred toward over-prediction to ensure we did not miss any convective regions when they could be detected.

4) CASE VI

Finally, we depict a scene where the model under-predicted the presence of convection.

Figure 22 gives a brief visual overview of the relevant information pertaining to this case. The information is presented in the same order as it was in Case I, Case II, and Case III. Once again, more information about this scene can be found in the appendix.

Once again, we can examine the metrics specific to this scene. Before the model was altered, the SR was 0.922 and the POD was 0.172. Additionally, we note that there were 165 hits, 3,125 correct rejections, 14 false alarms, and 792 misses. After alteration, the SR was 0.730 and the POD was 0.257. There were 246 hits, 3,048 correct rejections, 91 false alarms, and 711 misses. Essentially, within this scene, the model was able to correctly identify where convection was located quite reliably when it predicted convection, but did not predict convection nearly enough.

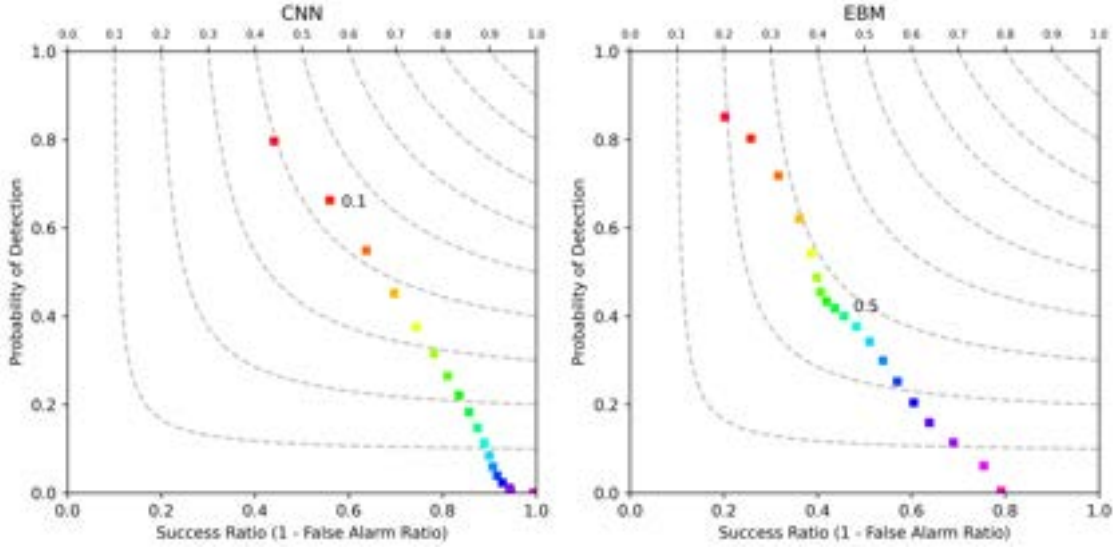
To get a better sense of why the model under-predicted on this scene, we can observe both the Original Scene and the Cool Contrast Tiles. First, examining the Original Scene, it can be seen that there is very little texture in the areas where the MRMS suggested there to be convection. In fact, there is very little measured texture throughout the entire scene. This can be seen by referencing the Cool Contrast Tiles. There, we observe that, even when texture is present, based on the opacity of the tiles, the texture is not very "intense." This is reflected in the final predictions as convection was predicted in the regions with the most texture (i.e. the areas with the darkest Cool Contrast Tiles values) but not in regions with slightly less.

The intended purpose of this feature was to indicate the presence of convection in highly textured areas, so this performance is not surprising given the scene, but it is worth mentioning that, when there is not much texture but there is still evidence of convective regions, the EBM's predictions may suffer.

d. Part Three: Convolutional Neural Network Model Comparison

Finally, to conclude the analysis of our model's performance, we compare the results of our edited EBM and a CNN model that aimed to solve the same problem Bansal et al. (2023). We present in Figure 23 similar SR vs. POD diagrams as were displayed in Figure 17.

We ran the CNN on the same (reduced) testing dataset and obtained measures of these metrics for different thresholds, noting that, as outlined by conversing with Bansal et al. (2023), their optimal threshold was 0.1 while our's was 0.5. An interested reader should consider reading Lee et al. (2021a) in addition to Bansal et al. (2023) for more information on attempting to solve this problem as well as other, related problems. The results of our edited model, as seen previously in Fig. 17(b), are shown in Fig. 23(b). The results of their model, then, are shown in Fig. 23(a). We note that, in general, their model performed worse on this testing dataset than on other datasets (as presented in Lee et al. (2021a)). Upon visual examination of many examples within this dataset, we note the potentially confusing nature of many cases, including cases where the MRMS ground truth did not align visually with where we as human observers would expect to see convection. At their optimal threshold (0.1), the CNN was able to achieve a SR of 0.559 and a POD of 0.663. As discussed, our edited model was able to achieve a SR of 0.456 and a POD of 0.401. In terms of

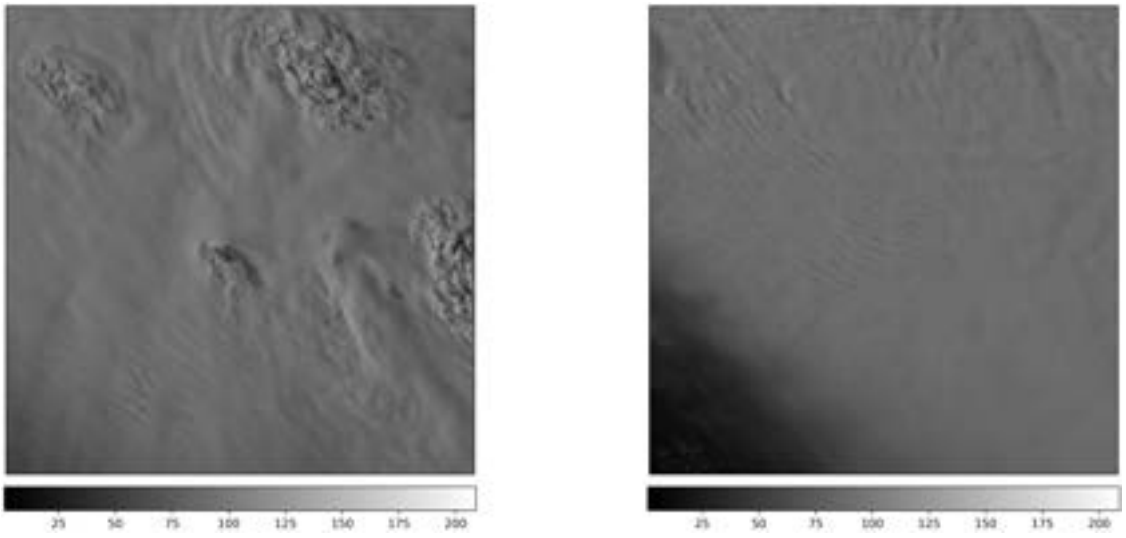


77 FIG. 23. Performance diagrams (1 - FAR vs. POD) for (a) the CNN and (b) the altered EBM for threshold
78 values ranging from 0.05 to 0.95. Presented CNN results were made at the marked threshold in (a), 0.1, and the
79 presented EBM results at the marked threshold in (b), 0.5.

both SR and POD, their model was able to out-perform our model on the chosen testing dataset, though by a smaller margin than we were expecting given their disclosed results on other datasets.

To get a better sense of how these models compares, we present the results of both our model and their model on individual scenes. We will present two scenes not seen in the previous four cases. The predictions will be based off of the visible imagery scenes seen in Figure 24. The first scene, Fig. 24(a), represents a case where we believe the EBM performed reasonably well. The second, Fig. 24(b), represents a case where the EBM over-predicted the presence of convection. In the interest of keeping this discussion brief, we will not mention the individual metrics of these predictions as we did for the previous cases.

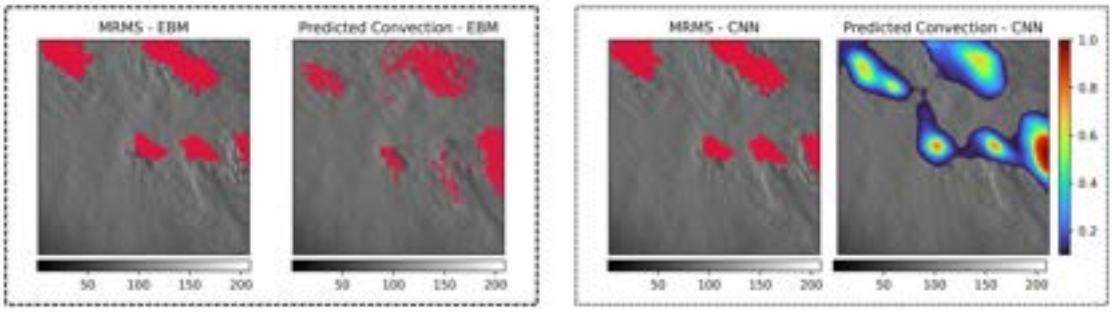
Before we get to the results, we note that their model was trained on full-resolution data where as our data, as mentioned, had a reduced resolution. As such, we will compare our results to the ground truth we used and their results to the ground truth they used. The predictions made corresponding to the visible imagery seen in Fig. 24(a) can be seen in Figure 25 and the predictions made corresponding to the visible imagery seen in Fig. 24(b) can be seen in Figure 26.



(a) (b)

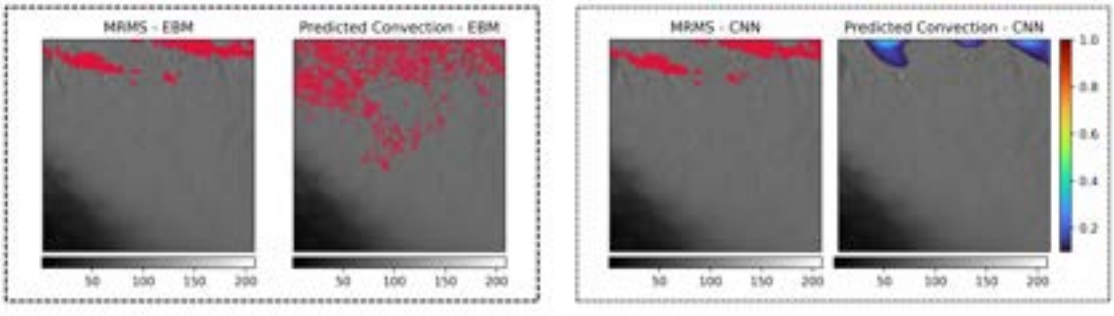
80 FIG. 24. Two scenes of visible imagery (5 May 2021) used to compare model performance between the EBM
81 and CNN models.

Fig. 25(a) shows the MRMS ground truth for our EBM and Fig. 25(b) shows the EBM's predictions. Here, of particular interest are the predictions made in the upper left-hand corner of the scene. Here, by visual inspecion of the scene in Fig. 24(a), we can see an isolated region of "bubbly" texture surrounded by (mostly) smooth clouds. Despite this very clear indication of an overshooting top, the MRMS data suggests the convection is located further up than any human would classify it as being. The same can be said for the overshooting top seen in the upper-right corner. In fact, throughout this scene there appears to be visual displacement errors present within the MRMS ground truth used. Despite this, the EBM was able to accurately pin-point these overshooting tops and predicted convection accordingly. We note, however, that the EBM was not perfect in its predictions as it made a mistake in the lower-right corner of the scene, placing convection in a region with potentially "confusing" texture. Fig. 25(c), then, shows the MRMS ground truth for the CNN and Fig. 25(d) shows the CNN's predictions. Note, again, the difference in resolution between the two models. Here, by comparing the MRMS and the predictions, it can be seen that the CNN was able to more accurately predict the individual locations of convection. Visually, the CNN appears to agree with the MRMS ground truth far more than the EBM did. Additionally, it did not make the same mistake the EBM did when there was potentially confusing texture.



(a) (b) (c) (d)

82 FIG. 25. Performance of our EBM: (a) MRMS and (b) predicted convection vs the performance of the CNN:
83 (a) MRMS and (b) predicted convection.



(a) (b) (c) (d)

84 FIG. 26. Performance of our EBM: (a) MRMS and (b) predicted convection vs the performance of the CNN:
85 (a) MRMS and (b) predicted convection.

Given we cannot know the true accuracy of the MRMS data, we cannot say for sure which model did a better job at capturing the "true" location of convection seen within the scene, only that the CNN was able to more faithfully re-create what was seen in the MRMS data.

Next, we can move on to the predictions made corresponding to the visible imagery seen in Fig. 24(b). Fig. 26(a) shows the MRMS ground truth for our EBM and Fig. 26(b) shows the EBM's predictions. The performance of the EBM seen in this case represents typical performance on scenes that feature confusing texture. The EBM is unable to distinguish between texture causes by overshooting tops and texture via other means and, because of this, predicts the presence of convection in areas that it should not. Interestingly, the EBM did not assign a very high probability to the predictions in made in many of these regions, indicated by the semi-transparent shading of many of the predictions. Fig. 26(c), then, shows the MRMS ground truth for the CNN and

Fig. 26(d) shows the CNN's predictions. Interestingly, the CNN did not necessarily predict the locations of the convective cells any more accurately than the EBM did. Because this particular scene is quite "confusing" in terms of MRMS location, we look past this. We note, then, that the CNN was able to accurately distinguish between convective and non-convective texture, something the EBM, as discussed, was unable to do. In regions where the EBM over-predicted the presence of convection because of the "ripple" texture, the CNN accurately predicted a lack of convection.

In fact, the main reason we share this specific scene is to demonstrate this specific disparity between the EBM and the CNN. Even when striping corrupted the visible imagery, something that wreaked havoc on the EBM's performance so much that we felt the need to remove scenes that featured it to ensure a fair representation of our model's capabilities, the CNN's predictions were unaffected. Speaking to Lee et al. (2021a), they noted that they did not notice the striping issue within the visible imagery as their CNN never had any issues because of it.

5. Limitations and Conclusions

In this work, we make the case for the use of EBMs in combination with clever feature engineering to yield interpretable ML algorithms for certain meteorological applications. This approach has several advantages, including, but not limited to (1) the ability to fully understand the strategies used by the ML algorithm when making predictions, exposing potential failure modes; (2) the opportunity to adjust its strategies to more closely match the strategies expected based on domain knowledge; and (3) the ability to develop a generalizable model from just a few data samples.

We have illustrated how these advantageous aspects of the EBM framework can aid in the approach of detecting convection from satellite imagery. We emphasize, however, that this application of EBMs was only possible due to feature engineering that first simplified the task at hand. Nevertheless, we believe that this method has the potential to be used in a wide variety of meteorological applications.

Lastly, for our considered application, we reinforce the notion that the EBM model was unable to reach the same level of performance as other state-of-the-art CNN approaches. Despite the gap in performance, we were pleased that such a simple and transparent approach, one that makes pixel-based predictions and does not consider temporal relationships, was able to perform as well as it did using very few and very simple features.

In summary, we have only just scratched the surface regarding the exploration of potential uses of EBMs. Much work remains to further explore the use of EBMs in the field of meteorology in terms of both identifying the most suitable applications and developing a larger range of engineered features—endeavors we hope will serve to further improve the performance of these models.

Acknowledgments. This material is based upon work supported by the National Science Foundation under AI institute Grant No. 2019758 and CAIG grant No. 2425923; and by the Machine Learning Strategic Initiative at the Cooperative Institute for Research in the Atmosphere.

References

Ai, Y., J. Li, W. Shi, T. J. Schmit, C. Cao, and W. Li, 2017: Deep convective cloud characterizations from both broadband imager and hyperspectral infrared sounder measurements. *Journal of Geophysical Research: Atmospheres*, **122 (3)**, 1700–1712, doi:https://doi.org/10.1002/2016JD025408, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016JD025408, https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2016JD025408.

Bansal, A. S., Y. Lee, K. Hilburn, and I. Ebert-Uphoff, 2023: Leveraging spatiotemporal information in meteorological image sequences: From feature engineering to neural networks. *Environmental Data Science*, **2**, e31, doi:10.1017/eds.2023.26.

Bedka, K. M., and K. Khlopenkov, 2016: A probabilistic multispectral pattern recognition method for detection of overshooting cloud tops using passive satellite imager observations. *Journal of Applied Meteorology and Climatology*, **55 (9)**, 1983 – 2005, doi:10.1175/JAMC-D-15-0249.1, URL https://journals.ametsoc.org/view/journals/apme/55/9/jamc-d-15-0249.1.xml.

Bedka, Kristopher, and Coauthors, 2019: Analysis and automated detection of ice crystal icing conditions using geostationary satellite datasets and in situ ice water content measurements. *SAE International Journal of Advances and Current Practices in Mobility*, **2 (1)**, 35–57, doi:https://doi.org/10.4271/2019-01-1953, URL https://doi.org/10.4271/2019-01-1953.

Berendes, T. A., J. R. Mecikalski, W. M. MacKenzie Jr., K. M. Bedka, and U. S. Nair, 2008: Convective cloud identification and classification in daytime satellite imagery using standard deviation limited adaptive clustering. *Journal of Geophysical Research: Atmospheres*, **113 (D20)**, doi:https://doi.org/10.1029/2008JD010287,

URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JD010287, https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2008JD010287.

Caruana, R., Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, 2015: Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, 1721–1730, KDD '15, doi:10.1145/2783258.2788613, URL https://doi.org/10.1145/2783258.2788613.

Ebert-Uphoff, I., and K. Hilburn, 2020: Evaluation, tuning, and interpretation of neural networks for working with images in meteorological applications. *Bulletin of the American Meteorological Society*, **101 (12)**, E2149 – E2170, doi:10.1175/BAMS-D-20-0097.1, URL https://journals.ametsoc.org/view/journals/bams/101/12/BAMS-D-20-0097.1.xml.

Gunshor, M. M., T. J. Schmit, D. R. Pogorzala, S. S. Lindstrom, and J. P. Nelson, 2020: GOES-R series ABI Imagery artifacts. *Journal of Applied Remote Sensing*, **14 (3)**, 032 411, doi:10.1117/1.JRS.14.032411, URL https://doi.org/10.1117/1.JRS.14.032411.

Hilburn, K. A., 2023: Understanding spatial context in convolutional neural networks using explainable methods: Application to interpretable gremlin. *Artificial Intelligence for the Earth Systems*, **2 (3)**, 220 093, doi:10.1175/AIES-D-22-0093.1, URL https://journals.ametsoc.org/view/journals/aies/2/3/AIES-D-22-0093.1.xml.

Hilburn, K. A., I. Ebert-Uphoff, and S. D. Miller, 2021: Development and interpretation of a neural-network-based synthetic radar reflectivity estimator using goes-r satellite observations. *Journal of Applied Meteorology and Climatology*, **60 (1)**, 3 – 21, doi:10.1175/JAMC-D-20-0084.1, URL https://journals.ametsoc.org/view/journals/apme/60/1/jamc-d-20-0084.1.xml.

Lee, Y., C. D. Kummerow, and I. Ebert-Uphoff, 2021a: Applying machine learning methods to detect convection using geostationary operational environmental satellite-16 (goes-16) advanced baseline imager (abi) data. *Atmospheric Measurement Techniques*, **14 (4)**, 2699–2716, doi:10.5194/amt-14-2699-2021, URL https://amt.copernicus.org/articles/14/2699/2021/.

Lee, Y., C. D. Kummerow, and M. Zupanski, 2021b: A simplified method for the detection of convection using high-resolution imagery from goes-16. *Atmospheric Measurement Techniques*,

14 (5), 3755–3771, doi:10.5194/amt-14-3755-2021, URL https://amt.copernicus.org/articles/14/3755/2021/.

Lou, Y., R. Caruana, and J. Gehrke, 2012: Intelligible models for classification and regression. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, 150–158, KDD '12, doi:10.1145/2339530.2339556, URL https://doi.org/10.1145/2339530.2339556.

Lou, Y., R. Caruana, J. Gehrke, and G. Hooker, 2013: Accurate intelligible models with pairwise interactions. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, 623–631, KDD '13, doi:10.1145/2487575.2487579, URL https://doi.org/10.1145/2487575.2487579.

Mamalakis, A., E. A. Barnes, and I. Ebert-Uphoff, 2022: Investigating the fidelity of explainable artificial intelligence methods for applications of convolutional neural networks in geoscience. *Artificial Intelligence for the Earth Systems*, **1 (4)**, e220 012.

Mamalakis, A., E. A. Barnes, and I. Ebert-Uphoff, 2023: Carefully choose the baseline: Lessons learned from applying xai attribution methods for regression tasks in geoscience. *Artificial Intelligence for the Earth Systems*, **2 (1)**, e220 058.

Mecikalski, J. R., and K. M. Bedka, 2006: Forecasting convective initiation by monitoring the evolution of moving cumulus in daytime goes imagery. *Monthly Weather Review*, **134 (1)**, 49 – 78, doi:10.1175/MWR3062.1, URL https://journals.ametsoc.org/view/journals/mwre/134/1/mwr3062.1.xml.

Moen, K., 2024: Gray level co-occurrence matrix and its application to weather satellite imagery, master's project, Colorado State University.

NOAA National Centers for Environmental Information (NCEI), 2024: U.s. billion-dollar weather and climate disasters. "NOAA National Centers for Environmental Information (NCEI)", URL https://www.ncei.noaa.gov/access/billions/, doi:10.25921/stkw-7w73.

Nori, H., S. Jenkins, P. Koch, and R. Caruana, 2019: Interpretml: A unified framework for machine learning interpretability. *ArXiv*, **abs/1909.09223**, URL https://api.semanticscholar.org/CorpusID:202712518.

Ribeiro, M. T., S. Singh, and C. Guestrin, 2016: "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, 1135–1144, KDD '16, doi:10.1145/2939672.2939778, URL https://doi.org/10.1145/2939672.2939778.

Rudin, C., 2019: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, **1 (5)**, 206–215.

Schmit, T. J., P. Griffith, M. M. Gunshor, J. M. Daniels, S. J. Goodman, and W. J. Lebair, 2017: A closer look at the abi on the goes-r series. *Bulletin of the American Meteorological Society*, **98 (4)**, 681 – 698, doi:10.1175/BAMS-D-15-00230.1, URL https://journals.ametsoc.org/view/journals/bams/98/4/bams-d-15-00230.1.xml.

Shapley, L. S., 1953: A value for n-person games. *Contribution to the Theory of Games*, **2**.

Smith, A. B., and R. W. Katz, 2013: Us billion-dollar weather and climate disasters: data sources, trends, accuracy and biases. *Nat Hazards*, **67**, 387–410, doi:10.1007/s11069-013-0566-5.

Veillette, M. S., E. P. Hassey, C. J. Mattioli, H. Iskenderian, and P. M. Lamey, 2018: Creating synthetic radar imagery using convolutional neural networks. *Journal of Atmospheric and Oceanic Technology*, **35 (12)**, 2323 – 2338, doi:10.1175/JTECH-D-18-0010.1, URL https://journals.ametsoc.org/view/journals/atot/35/12/jtech-d-18-0010.1.xml.