Enhancing Generalization of Machine Learning Models for Oncology Informatics

By

William Maxwell Schreyer

A DISSERTATION
Presented to

The Department of Biomedical Engineering
School of Medicine
Oregon Health & Science University

In partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
September 2025

Contents

List of Figures

List of Tables and Algorithms

List of Abbreviations

Abbreviation	Definition
AI	Artificial intelligence
AP	Average Precision
AUC	Area Under the Curve
CDRH	Center for Devices and Radiologic Health
CDW	Corporate Data Warehouse
CE	Cross Entropy
CLI	Command Line Interface
CMS	Centers for Medicare and Medicaid Services
CNN	Convolutional Neural Network
CPT	Current Procedural Terminology
DL	Deep Learning
DDI	Diverse Dermatology Images
ECE	Expected Calibration Error
EHR	Electronic Health Record
FDR	False Discovery Rate
FP	False Positive
FN	False Negative
FST	Fitzpatrick Skin Tone
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HAM10000	Humans Against Machine 10000
HCPCS	Healthcare Common Procedure Coding System
HIBA	Hospital Italiano de Buenos Aires
ICD	International Classification of Diseases
ID	In-distribution
kNN	k-Nearest Neighbors
KS	Kolmogorov-Smirnov
MC	Monte Carlo
ML	Machine Learning
MSP	Maximum Softmax Probability
MVP	Million Veterans Program
MSE	Mean Squared Error
NLP	Natural Language Processing
OOD	Out-of-distribution
PCA	Principal Component Analysis
PR	Precision-Recall
RAM	Random Access Memory
RGS	Radiation Gold Standard
ROC	Receiver Operating Characteristic
ROPA	Radiation Oncology Practice Accreditation
RMSE	Root Mean Squared Error
SAE	Supervised Autoencoder
SAGE	Supervised Autoencoder for Generalization Estimates
SGD	Stochastic Gradient Descent
TP	True Positive
UFES	Universidade Federal do Espírito Santo
UQ	Uncertainty Quantification
VA	Veterans Affairs
VAE	Variational Autoencoder
VHA	Veterans Health Administration
VINCI	VA Informatics and Computing Infrastructure

Acknowledgements

I'd first like to thank my mentor, Dr. Reid Thompson, who has supported me without fail since our first conversation over four years ago. Thank you for your encouragement, pulling me out of my rabbit holes and for challenging me to become a better scientist.

Thank you to Dr. Abhi Nellore who was the first teacher to welcome me back into the classroom after years of absence, and to Dr. Anthony Rhodes who taught me the theoretical building blocks that enabled this research.

I'd also like to thank my committee members – Dr. Shannon McWeeney, Dr. Evan Lind, Dr. Olga Nikolova and Dr. Kyle Ellrott – for their feedback in developing this dissertation.

Finally, thank you, Eli, for your love, patience and understanding. You believed in me before I believed in myself. I could not have done this without you.

To my parents, my brother, my son.

Abstract

Generalization – the ability of machine learning (ML) models to perform consistently when evaluated on new or varied data sources – is a key challenge in the development of effective, safe and ethical artificial intelligence (AI) systems. Despite extensive methodological research to improve generalization performance and reduce generalization error, many applications of AI show degraded capabilities when predicting on unseen data. These generalization gaps exist even in areas of intensive ML adoption and risk-sensitive settings such as oncology informatics, a field that seeks to improve the processes and delivery of cancer care. This work attempts to mitigate problems in generalization in three preclinical applications of oncology informatics. First, we demonstrate how universal billing and diagnostic coding systems enable learning across siloed data sources including Center for Medicare and Medicaid Services (CMS) and Veterans Health Administration (VHA) databases. We use this approach to predict historical radiation course dates from administrative data and assemble radiotherapy treatment history for over one million US veterans. Next, we develop a method for robust and interpretable dataset comparison using Supervised Autoencoders for Generalization Estimates (SAGE). We demonstrate how SAGE improves the performance of downstream classification and regression models for benchmark imaging and ecological datasets by removing out-of-distribution examples from evaluation, even from within the training data itself. Finally, we use our ensemble uncertainty estimation method as a comparison tool for dermoscopic imaging datasets for the purpose of identifying skin cancer malignancies. We automatically find and remove images with artifacts like measuring device occlusions and non-skin background and show how we can improve generalization of a separate malignancy predictor to a mixed dermoscopy dataset. Future research will

work to standardize implementations of SAGE within medical AI pipelines as a real-time measure of expected generalization error.

1 Introduction

Recent advances in computing hardware and algorithmic design have engendered the now flourishing field of oncology informatics which endeavors to build a software infrastructure for cancer care. The confluence of human expertise and powerful artificial intelligence (AI) systems is now recognized as a pathway to augmenting basic research, clinically translating findings and, ultimately, improving patient outcomes in cancer. This chapter addresses foundational topics in machine learning and applications to oncology that provide context to the remainder of the document.

Section 1.1. provides a brief overview of the history and design of common machine learning models, including those used in subsequent chapters, and introduces the problem of machine learning generalization along with methods to remedy gaps in performance that are found in the computer science literature. Section 1.2 gives a summary of the field of oncology informatics including an overview of how machine learning methods have been integrated to improve the efficiency and efficacy of various aspects of cancer care pipelines. This section also describes current challenges of machine learning generalization in oncology informatics and persistent problems in data drift detection. Section 1.4 summarizes the contributions of this dissertation regarding the novel research presented and perspectives advanced by our work.

1.1 Machine Learning and the Problem of Generalization

1.1.1 Machine Learning History and Overview

Machine learning (ML) is a methodology developed in the mid-20th century that uses algorithms to capture statistical patterns in data without explicit programming. The learned function represented by a given ML model is then able to automatically perform a task such as sorting examples based on preexisting groups. One of the earliest implementations of ML in practice was undertaken by the psychologist F. Rosenblatt who sought to model the perceptive functions of human neurons, coining the term "perceptron".[1] The perceptron is a simple classification algorithm that takes a real-valued vector as input and maps it to an output by taking the dot product of the input values and an equal-sized vector of tunable parameters, termed weights. (Figure 1.1) In Rosenblatt[1] a final activation function is applied to the product of the inputs and weights to produce a binary output. This forms the simplest building block of a feed-forward network, where an input is modulated by a series of nodes to produce a useful output.

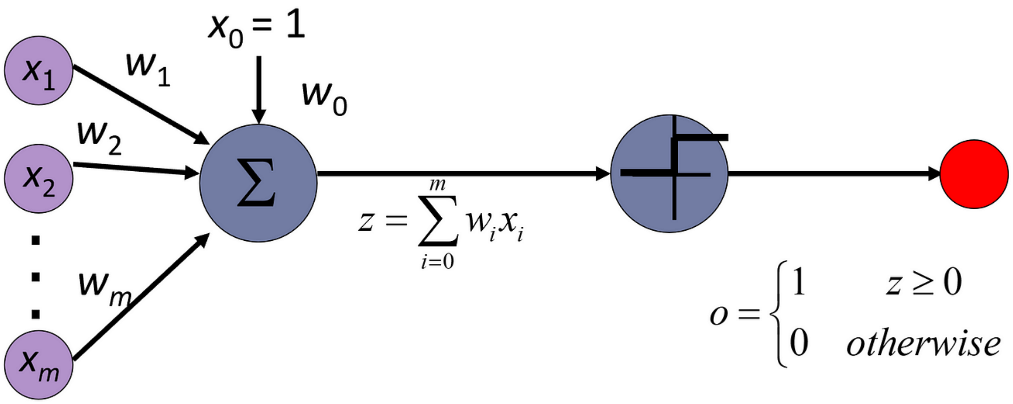

$x_0 = 1$
w_0
x_1
w_1
x_2
w_2
x_m
w_m
Σ
$z = \sum_{i=0}^{m} w_i x_i$
$o = \begin{cases} 1 & z \geq 0 \\ 0 & otherwise \end{cases}$

Figure 1.1 – Schematic of perceptron architecture. Image reproduced from [2].

In the case of the perceptron, when the binary output does not match the true label of the input vector a weight update procedure is implemented to change the decision boundary of the model and "learn" a new linear function. For each input, x_i, weights are updated as follows:

$$w_i \leftarrow w_i + \eta(t - y)x_i \quad \text{for } i = 0, \ldots, n$$

Where η is the learning parameter that scales the rate at which the weight values are updated. Note, this process only changes the values of the perceptron weights if the output value y does not match the target (ground-truth) value t in the term $(t - y)$. Successive rounds of weight updates are described colloquially as "training" a model to approximate a function for a series of inputs that better aligns the predicted outputs with the targets.

The discriminant capabilities of the perceptron were later expanded through the introduction of multilayer perceptrons, with additional "hidden" layers imbuing these neural networks with the ability to approximate any function instead of only linear ones.[3] The weight update process was also necessarily modified to allow for calculation of each node's contribution to the output values using a process called backpropagation.[4] Unlike perceptrons, neural networks have a defined loss function L with weight updates occurring during each round of model training. The gradient of L is efficiently calculated and utilized to update weight w connecting node j in the previous layer $(l-1)$ to node i in the current layer l:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial L}{\partial w_{ij}^{(l)}}$$

$$\text{for} \quad l = 1, \dots, m, \quad i = 1, \dots, n^{(l)}, \quad j = 1, \dots, n^{(l-1)}$$

In this way, the gradient of the objective function is distributed backwards to all weights connecting nodes across layers and enables learning across larger, fully connected networks.

In tandem with the spread of neural network architectures in the early 21st century, two other trends simultaneously developed to lay the groundwork for rapid advancements in ML development. First, increasing rates of data generation paired with lower costs of data storage vastly expanded the available sources of information for use in model training and testing. Fradkov [5] posits the adoption of statistical learning methods during this period was in part out of necessity, as there existed a fundamental inability to utilize big data sources without the automation capabilities afforded by ML. At the same time, the joint effects of increasingly available random-access memory (RAM) and the emergence of technologies for parallel computation like graphics processing units (GPU) both increased efficiency of the serial matrix operations required to train deeper neural network models and afforded larger quantities of data to remain in active operation. The confluence of these advances led to the development of the first truly deep neural networks with many hidden layers and up to millions of parameters capable of automatically learning complex and generalizable representations of high-dimensional data. For the first time, general-purpose models enabled by deep learning (DL) could be trained without exhaustive feature engineering or domain expertise.[6] (Figure 1.2) Further improvements to DL models for tasks like image recognition were achieved with convolution, where image features are abstracted from groups of pixel values in context, building the first convolutional neural networks (CNN).[7,8]

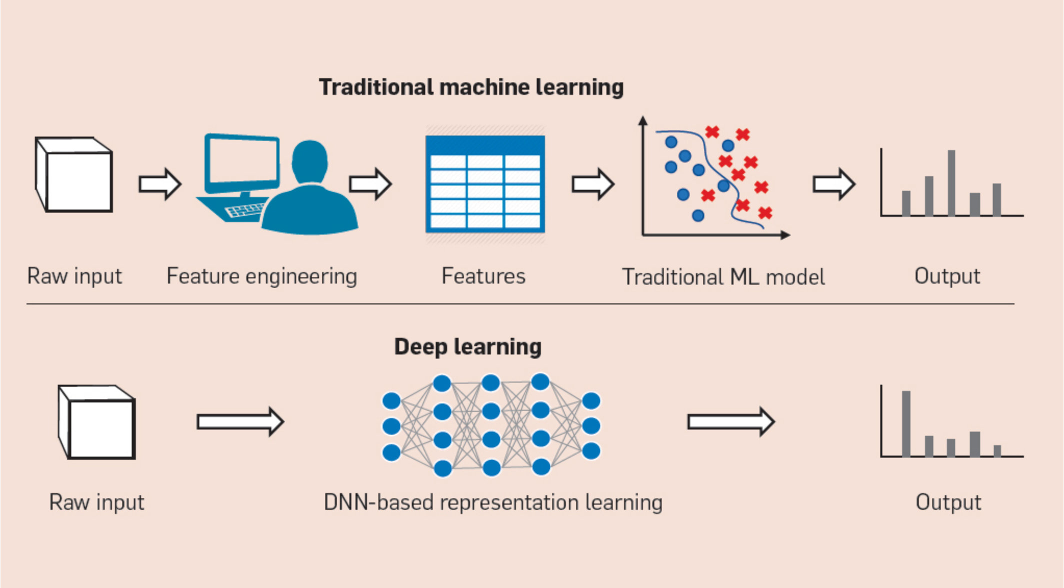


Figure 1.2 – Feature engineering required in traditional machine learning is bypassed by DL. Image reproduced from [9]. DDN – Deep Neural Network

Like the evolution of the perceptron, a similar process of experimentation and improvement was undertaken for decision tree classifiers and regressors. These simple models tended to overfit training examples, so constraints were applied to the process of creating decision trees such as restrictions on depth, the predictors used to construct leaves and the subsamples of data used during training. By bundling groups of weak trees together to form ensembles, random forests were able to deliver stronger performance by using the average of their uncorrelated predictions.[10] Whereas the bootstrap aggregating, or "bagging", process used in random forest creates an independent ensemble, "boosting" grows a tree ensemble sequentially by incorporating the residual error in a recursive fashion.[11] During this process, the m^{th} tree is added to the previous ensemble f_{m-1} for input x with a shrinkage term η that scales $h_m(x)$, the output of the new tree:

$$f_m(x) = f_{m-1}(x) + \mathrm{\eta} \cdot h_m(x)$$

In applications like AdaBoost (portmanteau of "Adaptive Boosting"), the boosted tree ensembles are quickly honed by weighting new inputs based on previous misclassifications. For sample i, the weighted input of the next tree $(m+1)$ is given as the product of the weight of the current ensemble and the performance of the m^{th} tree:

$$w_i^{(m+1)} = w_i^{(m)} \cdot e^{\mathrm{\alpha}_m \cdot I(h_m(x_i) \neq y_i)}$$

The variable α_m is determined by the inverse of classification error where better performance yields a higher value, giving those trees more influence on how the final ensemble outputs are calculated. Ensemble methods with variable weighting such as AdaBoost are intrinsically capable of handling data issues such as class imbalance and are therefore considered robust options for learning from noisy or incomplete datasets.

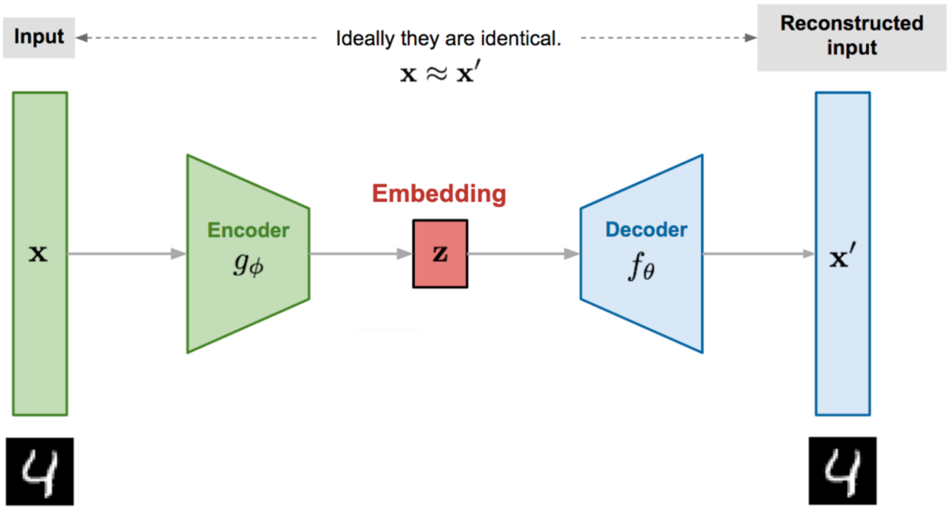


Figure 1.3 – Overview of the basic components of an autoencoder model using an image example from the MNIST dataset. Figure adapted from [12].

The examples of ML discussed thus far are from a discipline known as supervised learning, where sample labels are known in advance and required for training and evaluation. In situations where labeling is difficult or impractical, unsupervised learning is employed to model the intrinsic variation within datasets, often learning useful separations or representations of the data. A common example of unsupervised learning that incorporates neural network components is the autoencoder.[13,14] (Figure 1.3) Autoencoders contain an encoding network $g_\phi(\cdot)$ and a decoding network $f_\theta(\cdot)$ joined by z, a compressed embedding vector. By simultaneously training the encoder and decoder parameters (ϕ, θ) to take an input x and attempt to reconstruct it from z, the model learns a smaller representation of the data that can be used for additional tasks like clustering. In practice, the encoder and decoder functions are separate neural networks which learn weights that minimize the differences between the reconstructed input $x' = f_\theta(g_\phi(x))$ and the true input x so

$x' \approx x$. A common loss function used to train autoencoder models is the mean squared error (MSE):

$$L_{AE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - x'_i)^2$$

Like the process of neural network training, fitting an autoencoder involves feeding inputs through the network and weight updates via backpropagation.

Other variations of the autoencoder were introduced to accomplish tasks related to input reconstruction, such as remedying data corruption with a denoising autoencoder.[15] Iterations based on variational Bayesian methods like the variational autoencoder (VAE) use a probabilistic latent space instead of a fixed latent vector (z above).[16] By fitting the parameters of the latent space *distributions* to the training data, one can sample from this space to generate synthetic outputs that are similar to the training dataset. In this sense, VAEs can be considered an early example of deep generative models that now see widespread use in applications such as text-to-image creation, image enhancement, dataset augmentation, and more.

Finally, the flexible nature of the autoencoder architecture allows for applications of semi-supervised learning which combines the label-free training of a traditional autoencoder with a supervised task such as classification. These models, termed supervised autoencoders (SAE), typically contain a classifier function $h_\omega(\cdot)$ taking the compressed embedding vector as input and can be trained simultaneously with the decoder function.[17] A simple weighted loss equation used to optimize the encoder, decoder and classifier components during training can thus be constructed as follows:

$$L_{SAE} = \underbrace{\alpha \cdot \mathrm{MSE}\left(x, f_\theta\left(g_\phi(x)\right)\right)}_{\text{Reconstruction Loss}} + \underbrace{\beta \cdot \mathrm{CE}\left(y, h_\omega\left(g_\phi(x)\right)\right)}_{\text{Classification Loss}}$$

Here, the cross entropy (CE) loss uses the distribution of the classifier prediction for the training example x compared to the example's one-hot encoded label y. The loss terms are summed after applying weighting variables (α, β) which determine the strength of the contribution of each SAE component during training. In Le et al.[17], inclusion of a reconstruction error term with a small weight (α= 0.01) during SAE training was shown to yield higher test accuracy than a strict neural network of a similar size trained on the same data, thus demonstrating how adding an unsupervised regularization term such as MSE loss can enhance performance on a supervised task like classification.

1.1.2 Defining Generalization

After a ML model is trained it can be externally validated on a new dataset, sampled from some unknown distribution, to determine how the model performs on unseen data. The problem of machine learning model *generalization* is thus defined as the ability of a model, trained on data sampled from a source domain $\mathcal{D}_S$, to perform similarly on data sampled from a target domain $\mathcal{D}_T$. Ben-David et al.[18] establish the conceptual bounds of ML model generalization where they define a hypothesis function $h(\cdot)$ tasked with learning the true labeling function $f(\cdot)$ over the source domain with some error (or "risk") $\epsilon_S(h)$. The generalization error for the target domain is thus:

$$\epsilon_T(h) \leq \epsilon_S(h) + d(\mathcal{D}_S, \mathcal{D}_T) + \lambda$$

where $d(\mathcal{D}_S, \mathcal{D}_T)$ is the divergence between the source and target distributions, and λ is the minimum error for the true labeling functions. The source domain error is effectively minimized by a training algorithm while the difference between the source and target labeling functions is expected to be small so $E_{\mathcal{D}_S}[|f_S - f_T|] \approx E_{\mathcal{D}_T}[|f_S - f_T|]$. The primary impediment to successful generalization of machine learning models is therefore determined by the divergence between source and target distributions.

Differences in the target distribution that cause divergence are commonly referred to as *distribution shifts* and can arise from several phenomena including shifts in the covariates, labels or concepts underlying a given model.[19] Covariate shift is the most commonly studied form of data drift and occurs when the marginal distribution of input features, $P(x)$ differs between the training and test data, which does not necessarily alter the feasibility of a task like classification but can cause performance to suffer because the learned relationship between the labels and features $P(y|x)$ has changed. Sampling biases, data corruption and changes to the methods of data collection such as the collection device can all affect input feature distributions and cause covariate shift. Conversely, a changing relationship between ground-truth classes $P(y)$ with mostly invariant input features $P(x|y)$ is termed label shift and can similarly be caused by sampling biases or human labeling errors. For instance, a diagnostic algorithm may lack sufficient samples of a disease class in its training dataset because of some difficulty in sourcing the data. If the disease is more prevalent in the test patient population, the model experiences a label shift scenario causing a rift between predicted and actual ground truth labels. The third form of drift, concept shift, occurs when the relationships between the feature distributions and labels change and may cause the model to decay or become invalid over time. Concept shift is observed when assumptions about the data are violated such that the relationship between the label given the underlying input feature distributions $P(y|x)$ is no longer true. For example, a hypothetical recession that affects website activity on e-commerce sites can invalidate a purchase prediction model based on pre-recession online behaviors. Other changes to the data related to concept shift can affect model generalization such as the introduction of new ground-truth data classes during evaluation. This can occur when evaluating on populations beyond the scope of the model's expertise, such as evaluation of a taxonomic classification model on a new species that remained undiscovered at the time of model training and can lead to confident yet incorrect predictions.

The presence of the above phenomena and their effects on divergence lead to the formation of generalization gaps. Methods for the identification and amelioration of drift scenarios are discussed in the next section.

1.1.3 Strategies for Improving Generalization

Techniques for improving machine learning generalization have been developed in an attempt to reduce performance gaps when models are evaluated on new datasets and are enacted at different stages of model development, from data-centric interventions like augmentation and regularization during training to model-centric approaches like the recognition of data drift and post-deployment monitoring. (Figure 1.4)

The first data-centric strategy for improvement of generalization is geared towards making the training data itself more robust through augmentation, or the modulation of the input features via the addition of noise and other perturbations. By applying a composite of different image augmentations to the training dataset, Hendrycks et al. [20] show that DL methods can reduce their generalization error when tested on corrupted benchmarking datasets. Pre-training on adversarial examples, where imperceptible amounts of random noise are added to image vectors, also results in the reduced generalization error and improved model robustness compared with normal or adversarial training approaches alone.[21]

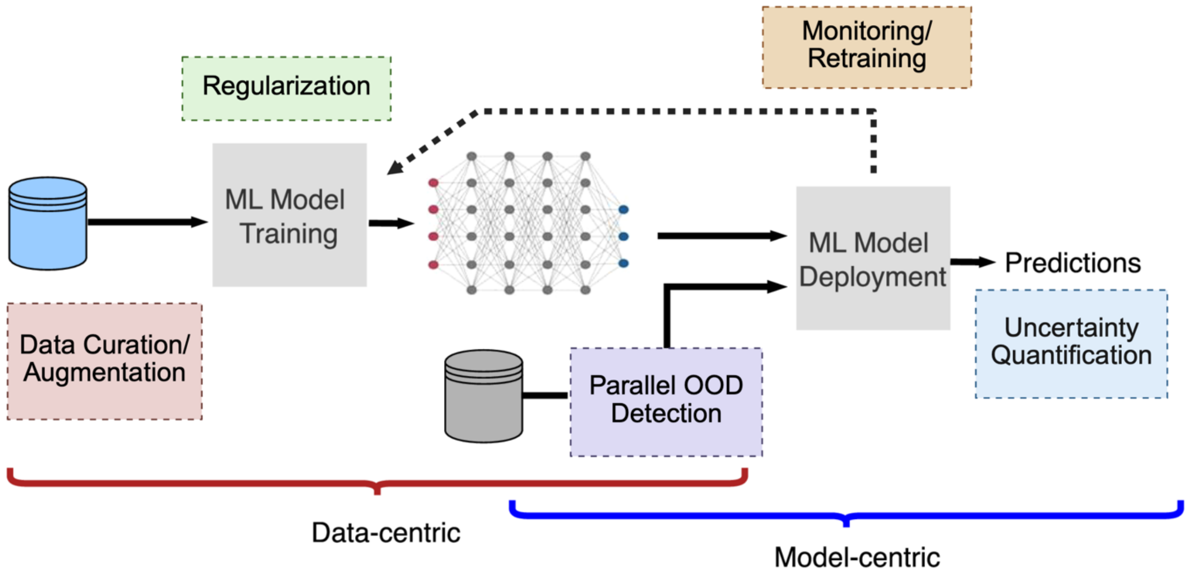


Figure 1.4 – Interventions to improve generalization of machine learning models during various stages of development. Figure adapted from [22].

Next, training time interventions such as regularization reduce the likelihood of overfitting, where a model is highly tuned to spurious patterns in the training data at the cost of test performance. This can occur when a mismatch exists between model complexity and the target function of the training data, such as a deep neural network containing many more parameters than the number of training examples. A simple method to explicitly limit complexity of large neural networks and force them to learn simpler solutions is to penalize the growth of weight sizes during training. [23] Adding the L_1 or L_2 norm of the weight vector to the model's error function, also called "weight decay", is a simple and common regularization technique applicable to a variety of ML models, from linear regression (LASSO[24], Ridge [25]) to adaptive optimization algorithms for deep neural networks[26].

Another explicit regularization method is ensembling, where several (ideally uncorrelated) models are averaged together to yield a more robust output than any of its members alone. A related idea is that of "dropout" in neural networks, where connections between nodes on the forward pass are randomly silenced during training to "thin" the network.[27] The trained model then approximates the ensemble of the thinned networks, which are prevented from co-adapting groups of neurons, and improves generalization performance.

Other forms of regularization are implemented implicitly in the model training schema, such as tracking the loss of a separate validation sample and "early stopping" the process once the validation loss begins to increase.[28] Interestingly, even the optimization algorithms commonly used in DL training have been demonstrated to have implicit regularization effects such as Stochastic Gradient Descent (SGD)[29] which uses an approximation of the gradient of the loss function from a random subsample of the data to reduce the computation burden of training. The choice to randomly sample the training data and optimize the approximation of the gradient causes models trained with SGD to generalize well even without other explicit regularization techniques.[30]

Turning to model-centric interventions, a key consideration for improving generalization performance is the identification and removal of test examples that are beyond the scope of the training data, or out-of-distribution (OOD), with respect to the input feature distributions. Intrinsic methods for OOD detection at the time of evaluation exist and attempt to quantify how "confident" a model is in its prediction. A popular approach for intrinsic uncertainty quantification (UQ) is to apply a softmax function to the raw logit output values of a DL classifier as a measure of confidence.[31] Maximum softmax probability (MSP) also stacks well with model calibration methods that reduce overconfidence in class estimates, such as temperature scaling, by making the MSP value match the percentage of the time that value is correct (e.g. MSP=0.8 means 80% of predictions are accurate).[32,33] More recently, the use of the negative maximum unnormalized logit output, MaxLogit, has garnered more usage because of its simplicity and inherent ability to avoid overconfident predictions when classes are semantically related.[34] Hybrid perturbation and calibration methods have also been proposed such as ODIN [35] which increases the confidence disparities between in-distribution (ID) and OOD samples, improving identification of the latter.

Other methods for UQ combine the advantages of well-developed mathematical frameworks of Bayesian inference with the strengths of deep networks, such as Monte Carlo (MC) dropout proposed by Gal and Ghahramani.[36] Interpretations of MC dropout as a form of model ensembling also inspired the creation of deep ensembles for UQ, where the prediction confidence is averaged across M simultaneously-trained networks.[37] Additionally, some methods use DL as a form of dimensionality reduction enabling comparison of statistical distances of embedded test datapoints to groups of embeddings from the training data. Examples include calculation of Euclidean distance to the k-Nearest Neighbors (kNN)[38] in the latent embedding space or the calculation of Mahalanobis distance of a test datapoint embedding to multivariate Gaussian representations of the training data[39,40].

These UQ methods for drift detection allow for labeling individual test samples as OOD, either with or without augmentation, model calibration and the use of model ensembles. Simple methods are also implemented at the sample level or dataset-wide by using robust statistical measures for drift detection. Rabanser et al. [41] rigorously explore the use of two-sample testing with various forms of dimensionality reduction to detect distribution shift, finding that both multiple univariate and multivariate Kolmogorov-Smirnov (KS) tests worked well to discriminate groups of OOD images when they reached between 100 – 1000 examples.

Two-sample tests for OOD detection are frequently employed as a method for monitoring input data changes over time and evaluating ML models in production. If test samples are determined to have statistically significant drift which yields lower performance, the test data is flagged and the original model may be re-trained with newer data that is a better representation of the current deployment environment.

1.2 Oncology Informatics

1.2.1 Overview of Machine Learning Integration

Oncology informatics is a branch of information technology research that applies computational tools to cancer research, prevention, diagnosis and treatment.[42] Over the last several decades this interdisciplinary field has grown in response to the meteoric increase in cancer datasets generated using new molecular and imaging technologies as well as the widespread adoption of electronic health records (EHR). The evolution of database infrastructure as well as recognition of the value of retaining key clinical and administrative data were the primary drivers behind the surge in big data in medical research in general, and in oncology in particular.[43] Furthermore, collaborative consortiums have built online databases that facilitate sharing of cancer datasets such as tumor DNA, RNA and epigenomic sequencing datasets[44], diverse imaging data including computed tomography, digital radiography, and magnetic resonance scans[45], and combinations of genomic and lifestyle data[46], greatly increasing the resources available to researchers across the globe. With expanded access to vast troves of cancer data, practitioners of oncology informatics quickly turned to ML algorithms; the insights once distilled in rules-based systems of explicitly coded logic[47] were now overwhelmed by the flood of new information which only automated statistical learning processes could integrate.

Sensibly, incorporation of machine learning models into aspects of clinical and pre-clinical oncology has been undertaken as limited, task-specific AI "touchpoints" rather than general purpose decision-making tools, each linked to its own use case in the continuum of cancer care.[48] Table 1.1 provides a non-exhaustive list of examples from the literature where ML tools have been introduced to address a key problem in oncology.

Table 1.1 – Examples of research studies involving machine learning methods in the oncology informatics literature. Studies are organized by the oncology task they address.

Oncology Task	Tissue	Data Source	Modality	Model Type	Output	Ref.
Risk Determination	Breast	BCAC Consortium	Genotyping Array	LASSO Regression	Polygenic Risk Score	[49]
Risk Determination	Prostate	PRACTICAL Consortium	Genotyping Array	Cox Regression	Polygenic Hazard Score	[50]
Risk Determination	Multiple	UK Biobank	Whole-body MRI	CNN	Future Cancer, binary	[51]
Screening	Breast	Digital Database for Screening Mammography	Breast mammography	CNN	Tumor Presence, binary	[52]
Screening	Colon	UC Irvine	Colonoscopy video	CNN	Polyp Presence, binary	[53]
Diagnosis	Skin	ISIC Archive, Dermofit, Stanford Univ.	Dermoscopic imaging	CNN	Taxonomic Disease Group	[54]
Diagnosis	Brain	National Center for Tumour Diseases	Long read DNA methylation	Neural Network	CNS Tumor Diagnosis	[55]
Tumor Subtyping	Breast	Various (n=5 academic centers)	Gene expression microarray	Nearest Shrunken Centroids	Tumor Subtype Label	[56]
Tumor Subtyping	Various (n=26)	The Cancer Genome Atlas	Multi-omics	Various (n=737)	Tumor Subtype Label	[57]
Treatment Response Prediction	Breast	Univ. Cambridge	Clinical, Pathology, Genomic, Transcriptomic	Model Ensemble (n=3)	Complete Response, binary	[58]
Adverse Event Prediction	Various (n=8)	Stanford Univ.	EHR-derived features	Logistic Regression	Acute Care Use, binary	[59]
Adverse Event Prediction	Lung, Skin, Kidney	CancerLinQ	EHR-derived features	XGBoost	Cardiac Adverse Event HR	[60]
Clinical Data Extraction	Breast	Peking Union Medical College Hospital	Free text clinical notes	BERT	Named Entry Recognition	[61]
Clinical Data Extraction	Breast	Partners HealthCare	Free text clinical notes	Conditional Random Field	Sentiment Analysis	[62]

For example, the ability to identify which people are at an elevated risk of developing cancer due to genetic and lifestyle factors holds the potential to improve screening protocols and allow for faster enactment of preventative measures. As such, multivariate regression models are constructed using genotype data with the goal of enabling interventions that could prevent cancer from arising in high-risk individuals.[49,50] Further areas of oncology undergoing targeted improvements with ML assistance include cancer screening[52,53], diagnosis[54,55], molecular subtyping[56,57] therapy response prediction[58], and data extraction from clinical notes[61,62]. Despite the many thousands of academic oncology studies utilizing ML methods, relatively few technologies have received clearance by the Federal Drug Administration (FDA) in the United States, with 736 unique device authorizations as of December 31, 2024.[63] The majority of approvals (84.4%) have been for devices that utilize imaging data as their core input with nearly 90% of those having radiology review panels. Thus, while the early promise of AI-assisted oncology tools is widely confirmed by the research, the tools receiving regulatory approval are ones which utilize highly standardized input data sources with objective interpretations like radiology scans and few others. The challenge in translating ML models from oncology research into clinical settings therefore largely stems from lack of consistent and generalizable performance with heterogenous real-world datasets, an obstacle seldom encountered in controlled academic settings.

1.2.2 Current Challenges in Generalization

The assumption that datasets are comprised of independent and identically distributed samples is often violated when ML approaches utilize medical datasets to address problems in oncology like those surveyed above. In such potentially risk-sensitive settings, a clear delineation of how and when these models fail is needed.

An initial illustration of this point is the current widespread use of benchmark datasets for developing oncology ML applications like tools for image-based diagnosis and natural language processing.[64–67] As Doerrich et al.[68] describe, academic ML and DL research is still heavily focused on incremental performance improvements on few influential benchmarking datasets, forgoing considerations of computational complexity or translational relevance that may limit future clinical adoption. Although important for establishing standardized evaluations and advancing theoretical performance gains, the use of benchmarking datasets does little to address the main difficulties in producing robust and generalizable algorithms and may worsen existing biases.[69]

A key reason why models trained on benchmarked data fail in practice is due to their brittleness when evaluating new samples that are poorly represented or completely absent from the training data. Important work has studied the effects of real-world distribution shifts on pretrained ML models, revealing that gains in accuracy on ID datasets are decoupled from robustness on OOD samples.[70] Such findings have been confirmed by studies like Schömig-Markiefka et al. [71] who discovered that a DL model trained on high-quality digital pathology images for detecting prostate cancer experienced a loss in performance when encountering commonly-occurring artifacts that were omitted from the training data, demonstrating the disconnect between training in controlled environments and evaluation in more heterogenous settings. Similarly, Petrie et al. [72] found that slightly degraded quality in curated dermoscopic skin lesion images through pixel-level perturbations like color shift or blur caused rapid drops in performance when assessing DL diagnostic tools.

Data augmentation techniques can be used to synthetically supplement imaging data and preempt sensitivity to distribution shifts by including geometric transforms like flips and rotations, random crops, masking and noise injection.[73] Others have gone further and introduced wholesale synthetic training examples through the use of generative unsupervised learning methods like VAEs and Generative Adversarial Networks (GAN), which are particularly promising for correcting dataset imbalance.[74] However, the possible augmentation space is vast with little standardization outside of imaging data. The introduction of complex generative methods sufficient to create high-quality synthetic data also significantly increases the computational cost of training.

Even with the use of augmentation techniques, the most theoretically reliable approach to avoiding performance degradation from distribution shift is for a model to have the ability to accurately report its uncertainty: that is, to tell users what it knows confidently and to avoid trusting predictions from what it does not. The reality of current best practices for quantifying predictive uncertainty is ironically quite ambiguous.

As discussed in Section 1.1.3, the most common measure of predictive uncertainty in DL methods is softmax confidence.[31,32] Although only 3 of the 14 models displayed in Table 1.1 provide an analysis of predictive uncertainty, all use a version of calibrated MSP likely due to its simple calculation at time of inference.[53,55,57] Nevertheless, a 2019 study by Ovadia et al. demonstrated that most popular UQ methods at the time including vanilla softmax confidence and various calibration techniques like temperature scaling increasingly failed to produce accurate measures of predictive uncertainty and experienced higher expected calibration error (ECE) as images became more corrupted.[75] They found the most resilient measures were derived from models that considered epistemic uncertainty, like stochastic Bayesian methods or deep ensembles, although there was a tradeoff with computational cost (deep ensembles used $M = 10$ neural networks). More recent studies in image diagnosis in oncology settings have separately converged on the finding that robust

UQ is best achieved by ensemble methods, further demonstrating that the use of single point estimates of uncertainty like MSP or MaxLogit should be eschewed in risk-critical settings.[76,77] As it currently stands there is no consensus on which UQ methods to adopt for OOD detection, leaving practitioners of oncology informatics without a reliable framework for interpretable estimates of ML generalization failures.

Despite the lack of a consistent approach for UQ, the need for infrastructure governing AI model deployment and monitoring has been clearly identified. As of June 2024, the FDA was actively developing guidelines for radiology AI system monitoring as part of its postmarket surveillance program within the Center for Devices and Radiological Health (CDRH).[78] As Figure 1.5 depicts, the proposed monitoring steps would require unspecified data and model-level analyses of drift detection with subsequent alerts tied to model updates including retraining.

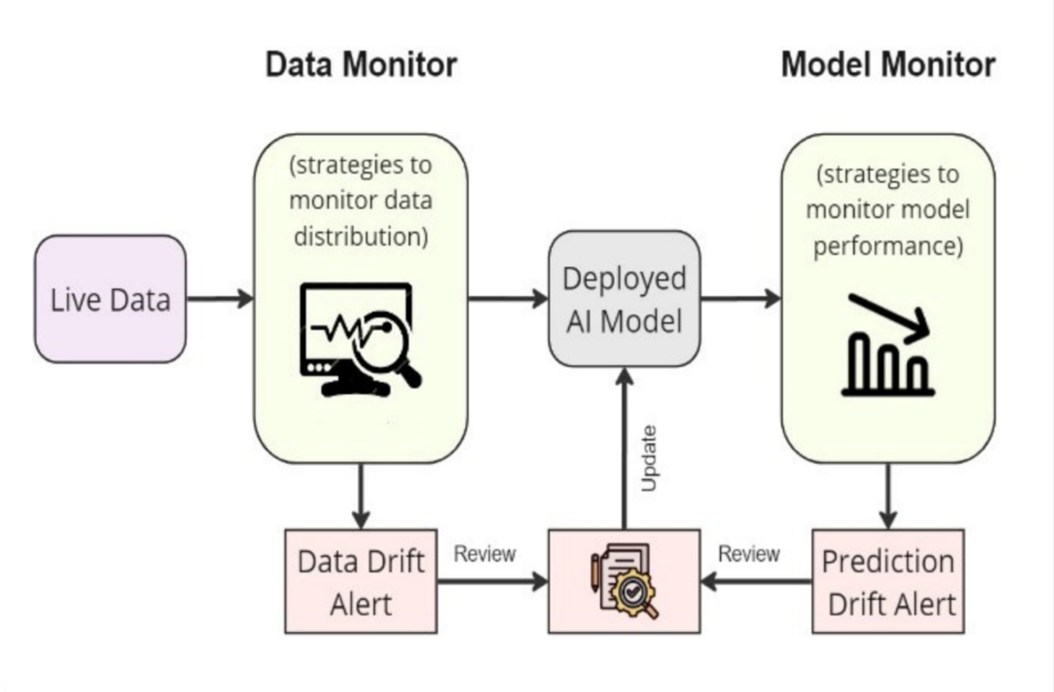


Figure 1.5 – FDA schematic of proposed radiology AI system monitoring including data and model-level analyses of drift. Image adapted from [78].

Although the FDA has acknowledged the importance of such monitoring frameworks, as of August 2024 less than half of approved devices have taken intermediate steps to assess generalizability such as publishing studies on post-deployment clinical performance.[79]

In the meantime, health care organizations have begun developing their own governance systems for AI monitoring with significant variations in design choices such as monitoring criteria and the statistical methods employed to detect data drift.[80] While analyzing changes to input features and model prediction distributions are common, they typically utilize sample-level testing (Wasserstein distance, KS two-sample test, chi-squared, etc.) or methods from control charts, a methodology originally developed for quality control in industrial manufacturing processes and of questionable suitability for complex medical datasets.[81] Furthermore, these analyses of data drift are implemented either after deployment or without the ability to discern which examples are causing drift that triggers alerts, demonstrating a lack of interpretability at the level of individual test examples.

1.3 Contributions of this Dissertation

This dissertation directly addresses the challenges of ML generalization in oncology informatics settings and is organized into three chapters based on works of original research. A final chapter proposes future directions and offers concluding thoughts on the ethical implications of this work. Chapter 2 details a data mining and feature extraction approach to retrospectively predict radiotherapy courses using universal billing and diagnostic coding systems as a basis for ML training and testing. Chapter 3 proposes a standardized method for using supervised autoencoders for generalization estimates (SAGE) and shows how an ensemble metric of predictive uncertainty is robust to varying degrees of data perturbation and corruption. Chapter 4 extends SAGE to the challenge of adapting the task of skin cancer malignancy prediction to new and heterogenous lesion images. We successfully identify image artifacts and demonstrate how filtering images with high uncertainty scores leads to improved predictive performance of a pretrained convolutional neural network model.

Chapter 2 is adapted from the following manuscript under review at *JCO Cancer Clinical Informatics*:

W. Max Schreyer*, Ryan Melson*, Christopher Anderson, Cecilia Madison, Evangelia Katsoulakis, Reid F. Thompson, "Automated identification of radiotherapy courses from US Department of Veterans Affairs administrative data", manuscript under review at *JCO Cancer Clinical Informatics* (2025), * equal contributors.

Chapter 3 is based on a manuscript under review at *IEEE Transactions on Artificial Intelligence*:

W. Max Schreyer, Christopher Anderson, Reid F. Thompson, "Generalization is not a universal guarantee: Estimating similarity to training data with an ensemble out-of-distribution metric", manuscript under review at *IEEE Transactions on Artificial Intelligence* (2025).

Chapter 4 is based on the following manuscript in preparation:

W. Max Schreyer, Ravi Samatham, Elizabeth Berry and Reid F. Thompson, "Ensemble uncertainty estimation improves skin cancer malignancy prediction", manuscript in preparation (2025).

2 Predicting Radiation Treatment Courses from VA Administrative Data

2.1 Abstract

Radiotherapy is a critically important cancer treatment; however, its details are often not well represented in electronic health record datasets. We present a supervised machine learning model that utilizes billing and diagnostic codes from Veterans Health Administration (VHA) and Center for Medicare and Medicaid Services (CMS) databases to predict radiation course dates with compelling accuracy (micro-average of 0.974 across classes). The retrospective application of our model to 1,331,342 patients coupled with a heuristic algorithm for assembling radiation courses identified 1,526,660 predicted courses of radiotherapy. The identified courses were collected into a shared resource to facilitate future VHA-based studies, and our predictive model is available for application to a wider range of non-VHA datasets, particularly those leveraging CMS data.

2.2 Introduction

Radiotherapy is a well-studied and robustly established pillar of cancer treatment, with over 60% of cancer patients receiving radiotherapy at some point over the course of their disease.[82] Radiotherapy can provide a targeted adjuvant and alternative to systemic therapies or surgical interventions to control tumor growth and metastasis, and in many cases serves as the primary curative treatment.[83–89] Furthermore, combination therapies that prescribe the coupling of radiation with chemotherapy and immunotherapy treatments have shown additive beneficial effects for cancer patients.[90–93] Beyond treatment of cancer, radiotherapy is indicated for many benign diseases such as trigeminal neuralgia and acoustic neuroma.[94] Overall, modern radiotherapy methods depend upon extensive planning and imaging procedures, and encompass a wide range of treatment options, each with its own potential variations in dose, frequency and other parameters tailored to the patient and their disease.

The Veterans Health Administration (VHA) reports nearly 50,000 new cancer diagnoses per year, the majority of which will undergo radiotherapy treatments.[95] Registered veterans are eligible to receive care at any of the 172 VHA hospitals or 1,138 outpatient treatment centers, which together comprises the single largest integrated healthcare system in the United States.[96] Nationwide, there are currently 41 facilities that deliver radiotherapy for cancer within the VHA, while approximately 60% of veterans receive radiotherapy at non-VHA centers, with their care paid for in whole or in part through the VHA. Records of radiation courses, the period over which a radiotherapy treatment is delivered, are maintained in siloed radiation oncology databases that do not integrate with a patient's health record. Furthermore, records for treatments delivered in a non-VHA setting are absent from these databases, even though they are the largest source of radiation courses for the veteran population. We therefore sought to develop a generalizable system for identifying radiation courses using a combination of Centers for Medicare and Medicaid Services (CMS) and VA Corporate Data Warehouse (CDW) administrative data, which co-locate with other health record information and allow for analyzing the full scope of past radiotherapy treatments.

Our initial attempts to identify prior radiation courses were manually coded and used rules-based heuristics from billing and diagnostic codes, showing promise in identifying clusters of days with radiation data but lacking key course date labels (e.g. “start” or “end”) that would be necessary for richer analyses. To improve the identification of specific radiation course dates, we sought to augment our efforts with the introduction of supervised machine learning (ML) models. The past two decades have seen an exponential rise in the number of oncology studies using ML methods and electronic health records (EHR) to extract or supplement key information from patient files.[97] Several approaches have successfully used ML models to classify patient phenotypes from a combination of diagnostic and billing codes, both of which are available within the VHA’s CDW and CMS databases.[98,99] We therefore aimed to train a multiclassifier model on retrospective administrative data and assemble past course dates into full radiotherapy treatments. Our goal was to formulate a tool that could clearly define radiation courses across the available history of VHA health records and generalize to other radiotherapy centers using the same billing and diagnostic code systems.

2.3 Results

2.3.1 Patient Cohort

We selected a cohort of 1,982 patients for training and testing our machine learning models (Figure 2.1A) — 419 from manual chart review alone (RGS) and 1,563 from a prior study of VA-wide radiation practices (ROPA). For each radiation course we extracted a subset of all course days, up to eight per course, with one of five labels assigned to each selected course date. (Figure 2.1B) A set of 304 features was calculated for each course date using a combination of administrative procedure and diagnostic code usage patterns over specified time windows. (Figure 2.1C)

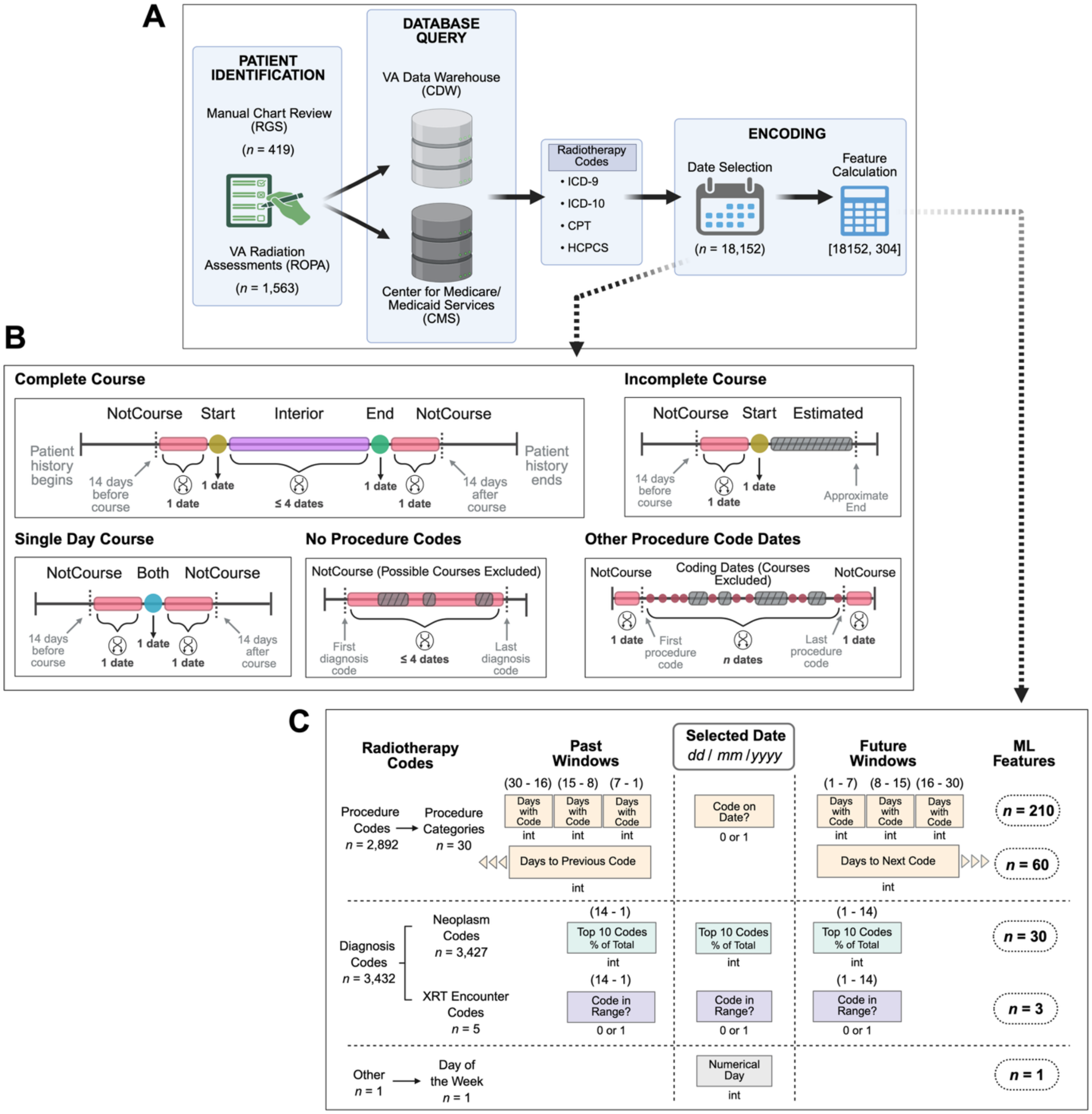


Figure 2.1 – Visual guide to the development of data sets used by the radiation course date prediction models. A) Overview of ML cohort selection and data extraction. B) Visual description of the sampling process used to select course dates from complete, incomplete and single-day radiation courses. C) Expanded depiction of the feature encoding process.

To broadly compare the review sources used for our ML cohort, we calculated summary course and patient demographic statistics for both RGS and ROPA. (Table 2.1) Approximately 97% of patients were male across both data sources and the average age was 67.25 years, in line with demographic statistics for US veterans overall. ROPA patient diagnoses consisted of lung and prostate cancers receiving either external beam therapy or brachytherapy, while RGS patients were specifically chosen to cover a wider spectrum of diseases and radiation treatments. To this end, we observed differences in measures of average complete course length (33.44 vs. 49.96 days), the range of courses per patient ([1 - 23] vs. [1 - 3]) as well as the range of complete course lengths between the RGS and ROPA datasets. In total we identified 2,147 radiation courses, 476 from RGS patients and 1,671 from ROPA.

Table 2.1 – Cohort statistics for patients utilized for machine learning training and testing.

Cohort	**RGS**	**ROPA**
Patients	419	1,563
Average Age	66.52	67.46
% Male	92.39%	98.46%
Courses	476	1,671
Average Complete Course Length (days)	33.44	49.96
Range Complete Course Length (days)	[2 – 126]	[5 – 103]
Median Courses/Patient	1	1
Average Courses/Patient	1.13	1.07
Range Courses/Patient	[1 – 23]	[1 – 3]

We further compared the distribution of features in our final RGS and ROPA datasets after dimensionality reduction, observing that RGS not only encompassed but significantly expanded the ROPA feature space when visualizing the combined data. (Figure 2.2) This confirmed the increased diversity in cases with the introduction of RGS — plotting ROPA data alone showed distinguishable clusters of radiation course date labels whereas RGS clustering patterns were diffuse. The complete ML dataset with course date labels also showed distinct clusters with clear separation between dates within a radiation course and those outside of a course, qualitatively indicating the feasibility of our proposed classification task.

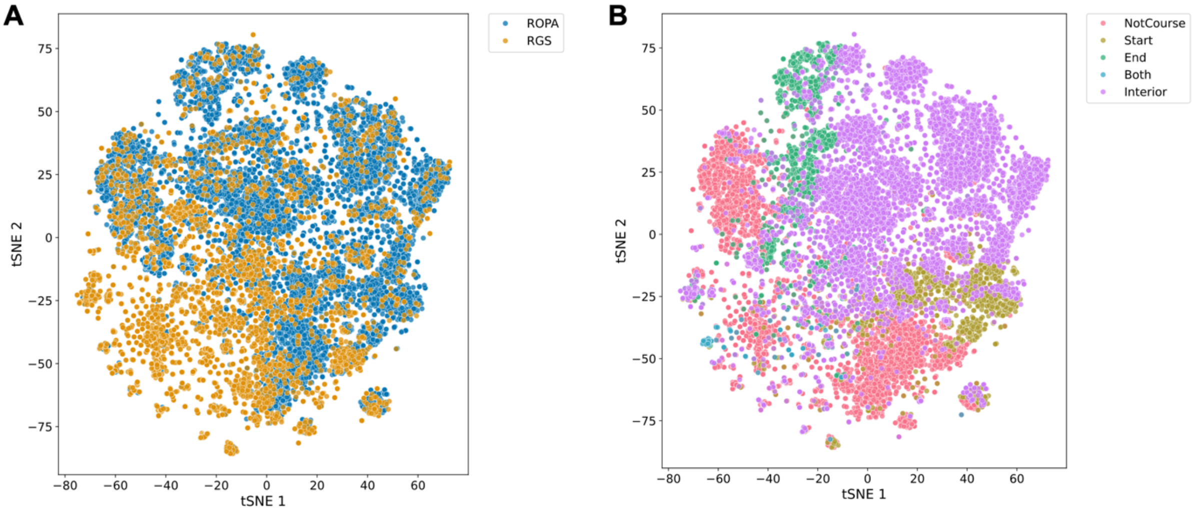


Figure 2.2 – Summary visualizations of radiation dates after feature encoding process projected into t-SNE space. A) Points labeled by cohort show inter-dataset differences of encoded radiation dates. B) Points labeled by radiation date class.

2.3.2 Trained Models Predict Radiation Course Dates

We split our combined feature set into training and test groups by patient and trained out-of-the-box random forest, AdaBoost, and neural network ML models (Methods). Each of the models achieved high overall accuracy when applied to the held-out dates (minimum 96.3%), and results were averaged across predicted classes. The majority of correct predictions were attributed to dates outside of a radiation course ('NotCourse') or dates within a course ('Interior'). AdaBoost and random forest models shared the highest prediction accuracy for these classes with 99% and 98%

respectively and shared the highest overall accuracies at >97%. (Figure 2.3A, D) Our neural network was effective at predicting radiotherapy course start and end dates (94% and 95% accuracy) but achieved slightly lower performance overall. (Figure 2.3H) All models struggled to accurately predict dates corresponding to single-day radiotherapy courses ('Both') with a maximum 77.8% accuracy for AdaBoost.

In addition to measures of accuracy, we calculated Receiver Operating Characteristic (ROC) curves. These showed extremely high areas-under-the-curve (AUC) both overall and on a per class basis but were heavily skewed by the consideration of true negative rates. (Figure 2.3B, E, I) We therefore generated Precision-Recall (PR) curves to refocus performance evaluation on positive predictive ability. PR curves showed high average precision (AP) values for all models (average = 0.99), with our random forest model giving the highest AP measures for each class individually and overall. (Figure 2.3C) The random forest's AP values for minor classes including radiation start and end dates (AP = 0.99) and single-day courses (AP = 0.82) demonstrated the model's ability to maintain performance when tasked with differentiating less common radiation course events. The AdaBoost model performed similarly to random forest when measuring positive predictivity of majority classes but yielded slightly lower precision when predicting start dates (AP = 0.97) and single-day courses (AP = 0.77). (Figure 2.3F) Our neural network model, although the weakest predictor of single-day courses (AP = 0.53), could detect dates within the bounds of standard radiation courses ('Interior' AP = 0.99). (Figure 2.3J) Overall, our random forest model demonstrated the best overall performance among the three architectures tested and was therefore utilized in all subsequent analyses.

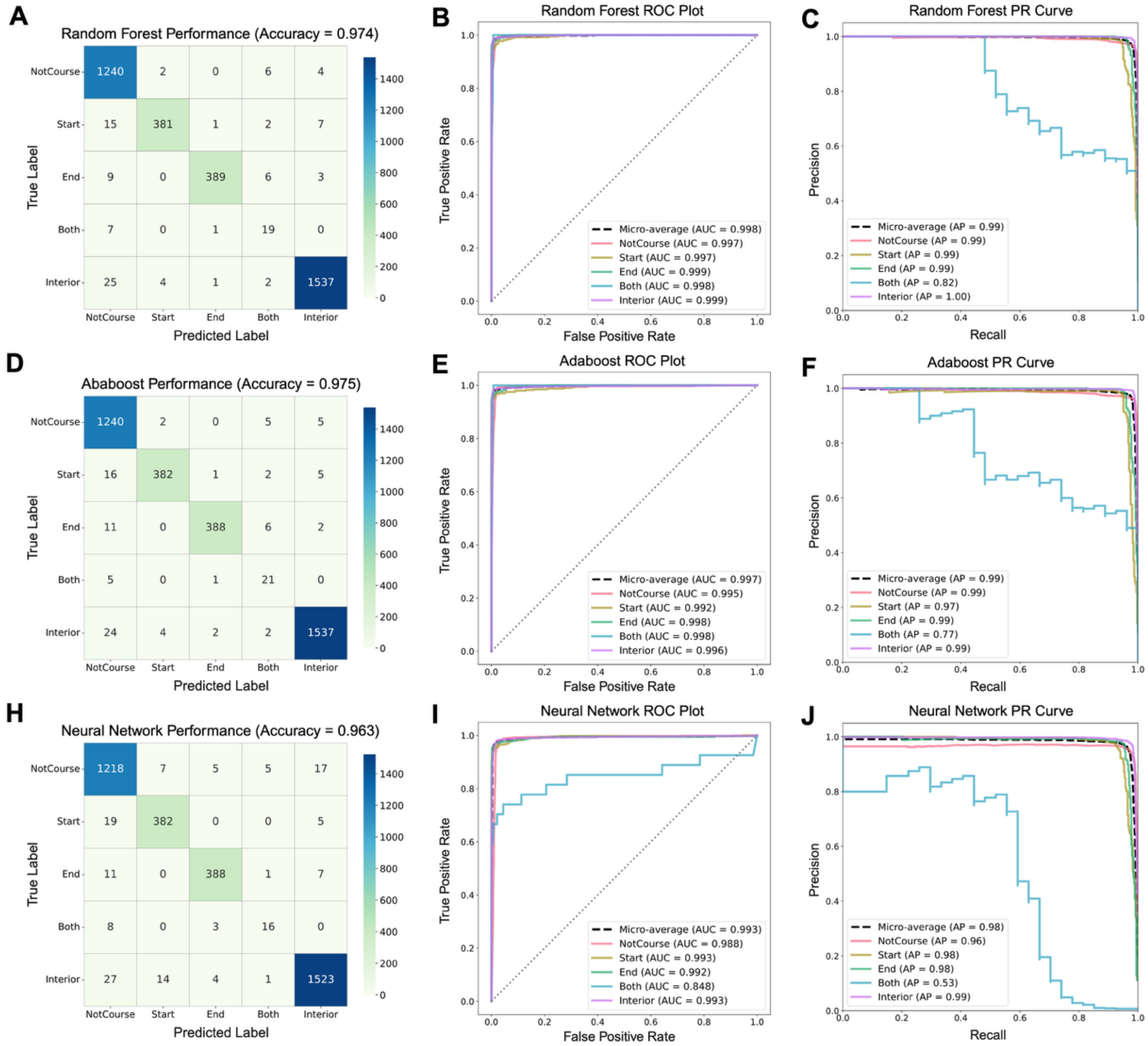


Figure 2.3 – Performance of trained random forest, AdaBoost and neural network models when evaluated on holdout radiation course dates. Confusion matrices (A, D, H) demonstrate classification accuracy. ROC curves (B, E, I) demonstrate label-wise classification performance across prediction thresholds. Precision recall curves (C, F, J) showcase positive predictive performance across thresholds.

2.3.3 Longitudinal Predictions Enable Radiation Course Assembly

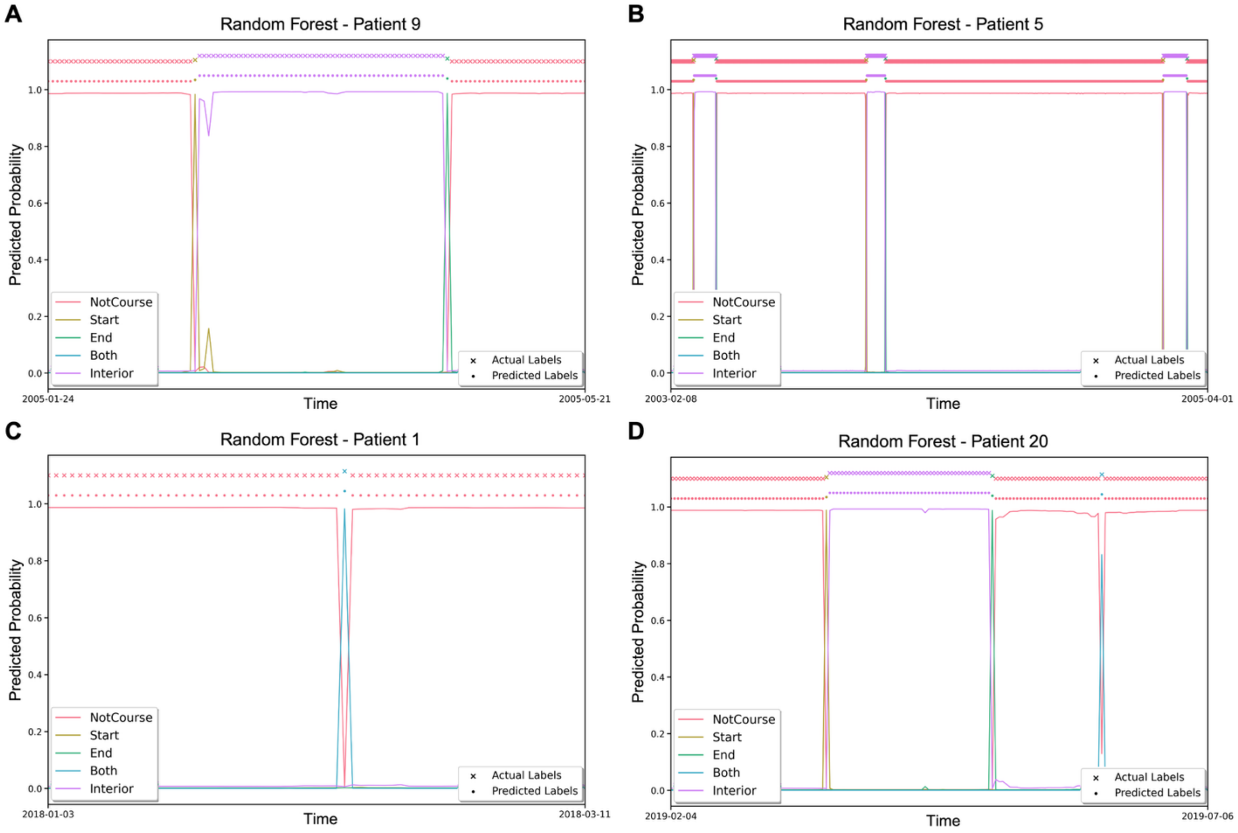
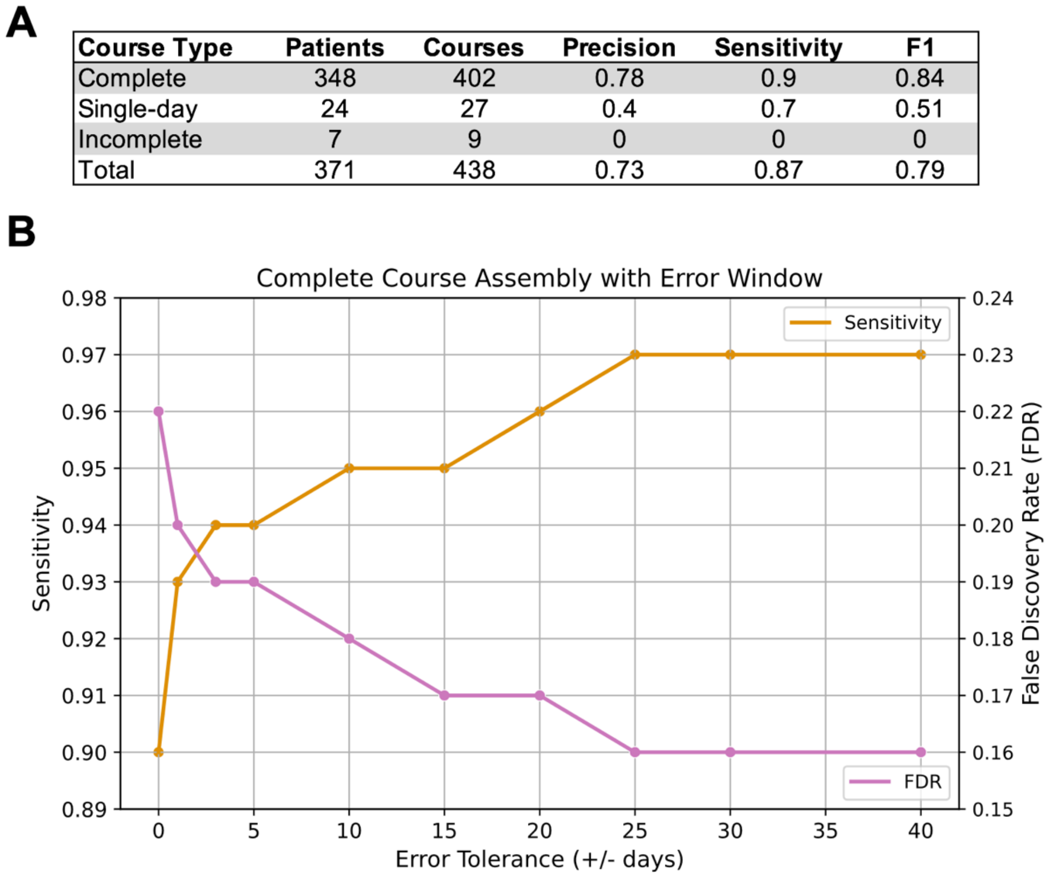


Figure 2.4 – Prediction probabilities for contiguous blocks of unseen dates. A) Complete radiation course, B) Multiple complete courses, C) Single day radiation course, D) Combination of complete and single day courses.

To ascertain whether our ML approach could be used to identify complete courses of radiation *de novo*, we applied it to a group of previously unseen, manually reviewed VA patients. We found that output probabilities for different course labels were well-behaved with clear agreement between predicted and actual classes over time. Our model was able to correctly identify course intervals from standard multi-day courses, single-day courses, multiple courses in succession and courses gapped by a many month hiatus with high confidence. (Figure 2.4)



A

Course Type	Patients	Courses	Precision	Sensitivity	F1
Complete	348	402	0.78	0.9	0.84
Single-day	24	27	0.4	0.7	0.51
Incomplete	7	9	0	0	0
Total	371	438	0.73	0.87	0.79

Figure 2.5 – Results of course assembly with heuristic algorithm after procedure date predictions for unseen patients. A) Summary table of performance metrics. B) Trend lines in Sensitivity and FDR metrics of course assembly with increasing error window for 'start' and 'end' date matching.

To test our ability to assemble radiation courses, we extracted only days with procedure codes for ML holdout patients and encoded features for each of the selected dates. We used our pretrained random forest model to predict radiation course dates for this dataset and assembled courses from predictions by matching start and end dates within a specified 105-day window (Methods). Our assembly algorithm showed a sensitivity of 0.9 and an F1 score of 0.84 when results were compared to the exact ground-truth start and end dates for holdout courses. (Figure

2.5A) We then expanded criteria for complete course matches by using error windows on either side of the ground-truth course dates (range +/- [1 - 60] days), further increasing sensitivity to a value of 0.97 and decreasing our false discovery rate (FDR) to a value of 0.16. (Figure 2.5B)

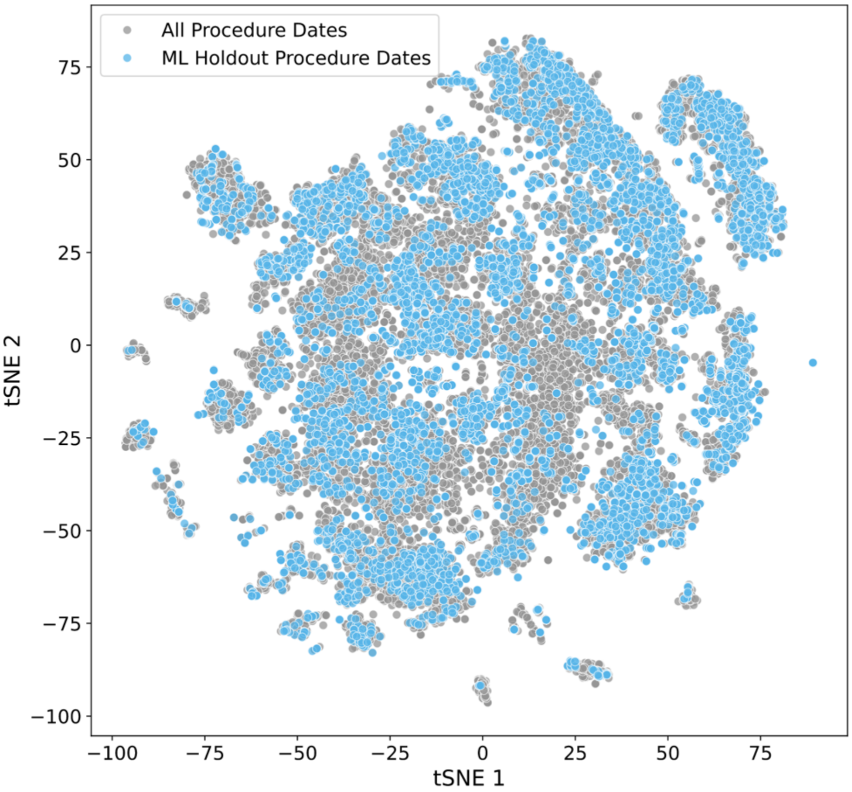


Figure 2.6 – Comparison of clustering patterns for VA-wide and holdout test datasets after dimensionality reduction with t-SNE. 20,000 sampled data points from all VA radiation procedure dates were joined with 10,000 sampled procedure dates from holdout test patients to create the combined dataset.

To assess the potential generalizability of our approach to a broader sample of patients, we randomly extracted 20,000 dates with radiation procedure codes from the VHA and combined this with 10,000 randomly sampled ML holdout procedure dates.

We calculated features for these dates and repeated our dimensionality reduction protocol, observing a high degree of overlap between clustering patterns for the two datasets (Figure 2.6). This led us to believe that application of our methodology to the full range of VHA administrative data could capture the vast majority of prior radiotherapy treatments.

2.3.4 Identification of Radiotherapy Courses Across VA Databases

We next applied our random forest model and course assembly algorithm to all known dates with radiation procedure codes available from within the CDW and CMS administrative databases. In total we extracted 32,406,809 procedure code dates for 1,331,342 patients which were processed using the previously described steps. Assembly of predicted course dates resulted in 1,526,660 radiotherapy courses from 1,152,310 unique patients, representing 92.8% of the possible pool of veterans with radiation procedure code data. The majority of predictions constituted complete courses (*n* = 1,191,110) followed by single-day courses (*n* = 292,462), with only 1.78% of start or end dates being classified as an incomplete course. (Figure 2.7A) Yearly counts of predicted complete courses showed a substantial uptick starting in the late 1990's and peaking during 2015-2018. We note that CMS data post-2018 was not available at the time of analysis resulting in a decrease in the number of predicted courses between the years 2019-2022. (Figure 2.7B) We next calculated the average length of complete courses per year, finding an increase between the years 2000-2010 before a notable decline lasting until the end of the study time frame. (Figure 2.7C) Additionally, we calculated the densities of complete course lengths finding the data conformed to a bimodal distribution: one sharp peak occurred at 15 days and another flatter peak emerged for 45-60 day courses. Finally, we determined the day of the week for all start and end dates, finding that only 0.15% of predicted start dates and 0.13% of end dates occurred on a Sunday.

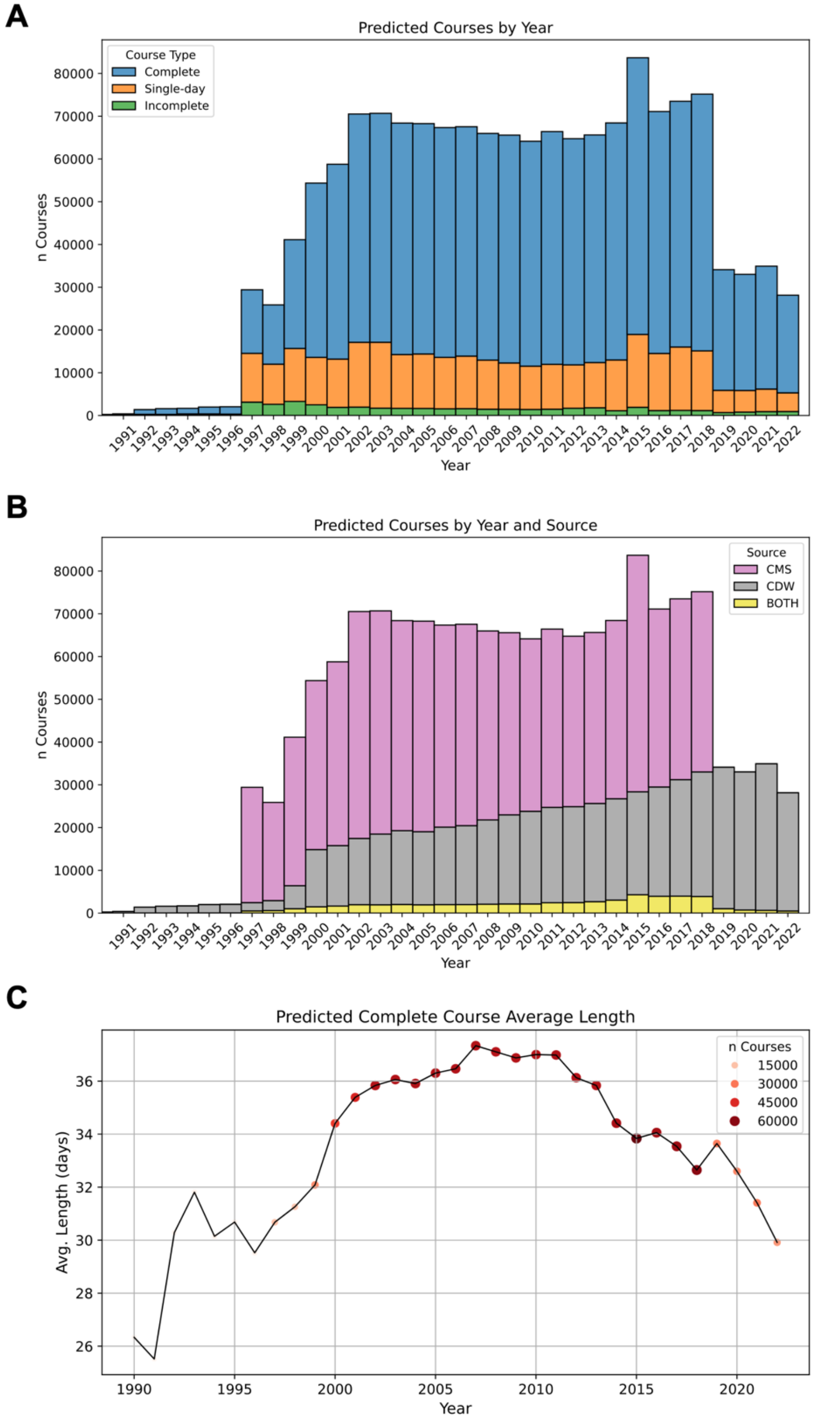


Figure 2.7 – Radiation courses predicted from all patient procedure dates (n = 32,480,062) across the available history of VA administrative data. A) Bar plots of radiation course predictions between the years 1991 to 2022 by course type. B) Bar plots during same period labeled by database source. C) Line plot of average complete course length per year. Dot size and color corresponds to the number of predicted complete courses for a given year.

2.4 Discussion

Our machine learning models, trained using administrative procedure and diagnosis data from 1,982 US Veterans, demonstrated excellent performance at identifying distinct radiotherapy courses from retrospective data. To the best of our knowledge, this is the first study to summarize radiotherapy treatment across the entire VHA system, as well as the first to provide a patient-specific resource of radiation course dates for the VHA. Moreover, using our random forest model and course assembly algorithms, we identified the largest cohort of radiotherapy treatments published to-date (*n* = 1,526,660), providing clear potential for future research. We believe our approach may successfully extend to other healthcare systems as our training data was sampled from all 41 radiotherapy-providing VA facilities and other non-VA sources, all of which use universal coding systems. Extension of these methods would simply require the extraction of the listed radiation codes per given date (Supplemental Table 2) and the generation of ML features as specified above (Supplemental Table 3) for input into our pretrained models.

In addition to predicting retrospective radiation courses, our chronological analysis of radiotherapy treatments allowed us to uncover trends in the administration of radiotherapy which are mirrored in other peer-reviewed studies. For example, efforts to limit radiotherapy adverse events and over-irradiation of patients lead to the adoption of hypofractionation and shorter average courses overall in the years post-2010.[100,101] The same patterns were observed in our data for predicted complete courses where the average course length dropped from a peak of >37 days in 2010 to about 30 days in 2022. We assessed the potential viability of these predicted courses by checking the proportion of course dates falling on a Sunday, a technique used previously as an indication of false positives for procedure data in the clinical radiation oncology setting, finding that only a tiny percentage of predictions matched this condition (average 0.14% for predicted start and end dates).[102] Further refinement of radiation course data could uncover more granular treatment trends filtered down to individual healthcare centers or patient demographic groups.

There are, however, several limitations to this study. Both the methodologies and patient cohorts we reported here are specific to the VHA, and we did not further extend our approach to non-veteran data. While VHA cohorts represent substantial diversity in some respects, we note that women are highly underrepresented, and pediatric cohorts are entirely absent. Furthermore, the date range of our predictions was limited by the availability of EHR and radiation courses delivered before the widespread adoption of computer systems in VHA hospitals (pre-1986) are missing, which also impacted the quality of administrative data during the transitional years of EHR adoption. Finally, while our method for classifying course dates relies on open-source ML packages, we used custom heuristics for broadly determining complete radiation courses, capturing the majority of common course lengths but potentially missing rare instances of treatment. This extends to the identification of single-day courses which comprise a wide range of treatment options (e.g. brachytherapy, stereotactic radiosurgery, palliative dose) and may not be faithfully captured by this multi-classification approach.

Radiation treatment dates and assembled courses from our retrospective analysis are available for use within the VA network – these predictions can be utilized to better understand delivery of radiotherapy at the VA nationwide, and enable data-focused policy guidance for future improvements to oncology care. Integrating our VA radiotherapy resources with patient health records would allow for the creation of patient cohorts that should rival or exceed the largest groups from retrospective meta-analyses of dose fractionation as well as intraoperative and post-surgical radiotherapy applications in different disease contexts.[103–105] Additionally, our radiotherapy course determinations could enhance predictions of radiotherapy outcomes including survival, tumor control and normal tissue toxicity by providing additional training data.[106–108] Given the unique access to paired health records and genomic data provided through the VA's Million Veterans Program (MVP), our work could also facilitate the future study of genetic links to radiotherapy outcomes like secondary malignancy at the site of radiation.[109]

2.5 Methods

2.5.1 Cohort Construction

To identify patients having received radiotherapy within (or paid for by) the VHA, we leveraged 307 radiotherapy procedure billing codes from the Current Procedural Terminology (CPT) and Healthcare Common Procedure Coding System (HCPCS) as well as 2,580 procedure codes from the International Classification of Diseases (ICD-9, ICD-10) systems. We queried the CDW (November 22, 1985 - October 31, 2022) using custom SQL scripts (Microsoft SQL Server Management Studio v18.11.1) with additional queries performed in SAS (Enterprise Guide 9.4_M6) on CMS data housed within the VHA (January 1, 1997 - December 31, 2018). We identified 1,333,286 patients with at least one radiotherapy-related procedure code.

From this set, we randomly-selected 217 patients for manual chart review to confirm radiation treatment along with course start and end dates, where relevant. We selected another 220 patients as representative edge cases for chart review to provide a richer and more diverse dataset. Edge cases included patients whose radiation courses utilized rare-procedure code categories or those lacking procedure codes entirely. After dropping patients with unconfirmable radiation histories or mismatching identifiers (*n* = 18), the randomly selected and edge case supplemented groups were combined to form our Radiation Gold Standard (RGS) cohort of 419 patients. We then incorporated an independent dataset of 1,563 individuals from a prior study of nationwide practice variation and quality assessment ("ROPA"), having removed one patient for overlap with our RGS cohort and another for only having approximate radiation treatment dates.[110] We conducted additional chart review for 785 ROPA cohort individuals to confirm the accuracy of radiation course dates. Chart reviews were conducted by R.M. and C.M. under the supervision of R.F.T., with radiotherapy start and end dates recorded as first and last days of treatment delivery for a course, respectively. Single-day courses were encoded with start and end dates occurring on the same date. If additional treatment course start dates were initiated prior to the end of another, they were considered part of a single compound course whose start date was the first occurring start date and whose end date was the last occurring end date.

2.5.2 Date Selection

We selected a subset of dates from radiotherapy course timelines to be used for the feature encoding process and the subsequent training and testing of our ML models. For all patients with procedure codes, a random date was selected before the first and following the last procedure code. Patients with a radiotherapy start and end date confirmed by chart review had up to eight dates chosen per course. These complete course timepoints consisted of 1) a random date 1-14 days preceding the start date, 2) the start date, 3-6) up to four randomly selected interim dates between the start and end dates (two with procedure data and two without), 7) the end date, and 8) a random date 1-14 days following the end date. Courses bearing a single confirmed start or end date were included as: 1) the start or end date and 2) a random date 1-14 days preceding or following the confirmed start or end date determined by manual chart review. For instances of single-day radiotherapy courses, time points consisted of 1) a random date 1-14 days preceding the single-day course, 2) the single-day course date, and 3) a random date 1-14 days following the single-day course. Where the gap between radiation courses was less than 28 days, the interval for selecting random dates preceding or following a course was calculated as half the distance to the closest course. Courses where both the start and end dates were ambiguous or unconfirmed were dropped from further consideration. For patients without procedure code data or with unknown/unconfirmed radiotherapy administration, four random dates were selected from the interval between their first and last neoplasm diagnosis code. Additional non-course dates with procedure code data were selected as follows: 4 random procedure dates between the first and last procedure code for patients confirmed to have not received radiotherapy, and $2 + (2 \times number\ of\ complete\ courses)$ for confirmed RGS radiotherapy recipients. Every time point was assigned one of five class labels: "Not Course" (*n* = 6099), "Start" (*n* = 2011), "Interior" (*n* = 7907), "End" (*n* = 2020), or "Both" (*n* = 115) for instances of single-day radiotherapy courses. In total we extracted input feature sets and corresponding class labels for 18,152 patient-dates used in model training and testing.

2.5.3 Feature Encoding

To develop a dataset appropriate for subsequent modeling, we constructed a set of date-specific quantitative features using 2,887 radiotherapy procedure codes, 3,432 neoplasm diagnosis codes and 5 radiotherapy encounter ICD diagnostic codes. We grouped procedure codes into 30 categories according to distinct aspects of the radiotherapy workflow (e.g. "Simulation", "Treatment - External Beam"). We created nine features for each procedure code category, reflecting the presence and density of code usage over past, present and future time windows, as follows: 1) a binary flag recording the occurrence or absence of at least one code on the given date, 2) the number of days between the specified date and the most recent past occurrence of a code in that category, 3) the number of days between the specified date and the next occurrence of a code in that category, 4) the number of days with at least one code occurring in the preceding 1-7 days from the selected date, 5) in the preceding 8-15 days, 6) and in the preceding 16-30 days, as well as 7) in the following 1-7 days from the selected date, 8) in the following 8-15 days and 9) in the following 16-30 days. This resulted in a total of 270 procedure code features, 9 for each of the 30 categories. For neoplasm diagnosis codes, we identified the 10 most used codes over a 29-day period with the selected date in the center of the time window. We then tallied the number of times each of the 10 codes was used and divided this number by the total number of neoplasm diagnosis codes to get a percentage value for three time windows: 1) the selected date, 2) 1-14 days before and 3) 1-14 days after the selected date. This resulted in 30 neoplasm diagnostic features, one for each of the top 10 codes over specified windows. For radiotherapy encounter diagnostic codes, we created a binary flag recording the presence or absence of a code over the same time windows as neoplasm diagnosis codes. This resulted in three encounter code features, one per window. For the final feature, we recorded the day of the week for the selected date numerically. In total, we generated 304 features for each selected date and used this feature set as input into our machine learning models.

2.5.4 Machine Learning Model Training

Final feature sets were generated on VA Informatics and Computing Infrastructure (VINCI) secure servers and loaded using Python (v. 3.10.11), NumPy (v. 1.21.5) and pandas (v. 1.4.2).[111,112] We randomly partitioned dates from 80% of patients into a training dataset using the scikit-learn (v. 1.2.2) package.[113] The remaining 20% of patients' data were held out as a final test set, while ensuring the approximate ratio of dates corresponding to each class was preserved between partitions. We normalized features using the 'StandardScaler' class fit to the training dataset. After scaling, we upsampled minor classes corresponding to radiation start days, end days and single-day courses by 100%, 100% and 300% respectively to reduce class imbalances. We implemented AdaBoost and random forest models from scikit-learn's ensemble methods and constructed a neural network model using TensorFlow (v. 2.11.0).[114] Using a grid-search with five-fold cross-validation from scikit-learn's 'StratifiedGroupKFold' function, we selected the following model hyperparameters yielding highest average validation accuracy: AdaBoost [tree depth = 10, number estimators = 200, learning rate = 0.1], random forest [tree depth = 15, number estimators = 300, maximum number features = 75], neural network [layers = [150, 75], dropout = [0.2, 0.3], learning rate = 0.001]. We then re-initialized and trained all models on the entire training dataset with the selected parameters. We simultaneously trained and calibrated our AdaBoost and random forest models using the 'CalibratedClassifierCV' function with default options.

2.5.5 Radiation Course Assembly Algorithm

To assemble radiation courses from ML model predictions, we devised a heuristic algorithm using a patient's predicted course dates organized by label. (Algorithm 2.1) In words, we sorted all predicted start and end dates chronologically for each patient and paired the first available start date with the last available end date – if this period was less than 105 days the pairing was considered a match. If the pairing was unsuccessful, we then tested the given start date with the patient's remaining end dates, assessing the end dates in reverse chronological order. This process was repeated until a match had been attempted for all predicted end dates. Successful matches were designated as predicted complete radiation courses, while any additional start and end dates that fell between matched course dates were removed from further consideration. We then tallied the number of consecutive predicted interior dates occurring within each complete course. This process was repeated until no start dates-remained, with any surviving start or end dates recorded as incomplete courses. All dates with a predicted 'Both' label were assigned as single-day radiation courses.

```
Require: predicted_dates (per patient), each labeled as Start, End, or Both
Ensure: complete_courses, incomplete_courses, single_day_courses
Initialize complete_courses ← $\emptyset$
Initialize incomplete_courses ← $\emptyset$
Initialize single_day_courses ← $\emptyset$
for each patient do
    Separate predicted_dates into start_dates, end_dates, both_dates
    Sort start_dates in ascending order
    Sort end_dates in ascending order
    Add all both_dates to single_day_courses
    while start_dates is not empty do
        $s$ ← first element in start_dates
        for all $e \in$ end_dates in reverse chronological order do
            if $e \geq s$ and $(e - s) \leq 105$ days then
                Add $(s, e)$ to complete_courses
                Remove $s$ from start_dates
                Remove $e$ from end_dates
                Remove any start or end dates strictly between $s$ and $e$
                break
        if no match was found for $s$ then
            Remove $s$ from start_dates
            Add $s$ to incomplete_courses
    for all $e \in$ remaining end_dates do
        Add $e$ to incomplete_courses
return complete_courses, incomplete_courses, single_day_courses
```

Algorithm 2.1 – Pseudocode of heuristic algorithm for assembling treatment courses from predicted radiation course date labels.

2.5.6 Evaluation of Date Predictions and Course Assembly

We evaluated our re-trained models on the holdout test dataset and built confusion matrices to show overall model performance and accuracy by label with scikit-learn's 'confusion_matrix' class. We calculated precision, recall and specificity using the 'metrics' module. Sensitivity and specificity values were used to generate Receiver Operating Characteristic curves, and Precision-Recall curves were created using corresponding measures using the default number of threshold values ($\# \, target \; scores \, + \, 1$). We calculated Area-Under-the-Curve and Average Precision values for each model and class. Additionally, we calculated micro-averaged curves for each plot, giving equal weight to predictions across class labels for each of the three models. All plots used in model evaluation were generated using the corresponding built-in scikit-learn class.

To evaluate our radiation course assembly algorithm, we manually labeled ML holdout patient course dates and used them as the ground-truth for testing predicted complete, incomplete and single-day courses. We compared predicted courses to ground-truth by exactly matching course start and end dates, calculating true positive (TP), false positive (FP) and false negative (FN) predictions. From these metrics we determined precision, sensitivity and F1 scores as well as the false discovery rate (FDR) for each course type. Additionally, we re-calculated complete course metrics by matching dates to a specified error window both before and after the true start and end dates, with windows of 1, 3, 5, 10, 15, 20, 25, 30, 35, 40 and 60 days. We then re-assessed precision, sensitivity, F1 score and FDR for each error window and recorded results.

2.5.7 Predicting Radiation Courses for Cohorts of Unseen Patients

An additional cohort of 26 manually validated radiotherapy patients that did not feature in our ML cohort was created to test the potential for our random forest model to identify whole courses of radiotherapy *de novo* from individual date classifications. These patients represented an array of radiation course types including conventional courses, single-day courses, and multiple successive courses with varying interval lengths in addition to numerous modalities of radiotherapy delivery (e.g. external beam, stereotactic body, brachytherapy). We extracted dates for all occurrences of radiation procedure codes, as well as 15-day flanking windows before and after each code instance, collapsing overlapping windows to form a single contiguous time block. We encoded features for all time block dates as previously described and generated class probabilities using each of our pre-trained ML models.

Line plots corresponding to the predicted probability of each multiclassifier output were created in Matplotlib.

For our broader analysis of radiotherapy courses across CDW and CMS databases, we extracted all dates between November 22, 1985, and October 31, 2022, with at least one radiotherapy procedure code for all patients, excluding those from the ML cohort (*n* = 1,331,342). We encoded features for the extracted dates (*n* = 32,406,809) and applied our trained random forest model and course assembly algorithm, tallying predicted courses by type. All subsequent plots were created with seaborn (v. 0.13.2).[115]

2.5.8 Dimensionality Reduction and Visualization

We performed principal component analysis (PCA) using the scikit-learn 'decomposition' module for 100 components in the ROPA and RGS feature sets and 125 components in the combined (RGS and ROPA) dataset, accounting for >80% of variance. We then randomly-sampled 20,000 radiation procedure code dates from the VA-wide dataset and 10,000 procedure code dates from our holdout patients, and repeated PCA using 50 components. We used the resulting matrices to perform t-SNE for each data source using the manifold module with a random initialization state and a perplexity of 40. Course and data source labels were subsequently applied to data points before plotting with Matplotlib (v. 3.5.1).[116]

3 Ensemble Uncertainty Estimation with SAGE

3.1 Abstract

Failure of machine learning models to generalize to new data is a core problem limiting the reliability of AI systems, partly due to the lack of simple and robust methods for comparing new data to the original training dataset. We propose a standardized approach for assessing data similarity in a task-aware, model-agnostic manner by constructing a Supervised Autoencoder for Generalization Estimates (SAGE). We compare points in a low-dimensional embedded latent space, defining empirical probability measures for *k*-Nearest Neighbors (kNN) distance, reconstruction of inputs and task-based performance. As proof of concept for classification tasks, we use MNIST and CIFAR-10 to demonstrate how an ensemble output probability score can separate deformed images from a mixture of typical test examples, and how this SAGE score is robust to a battery of transformations of increasing severity. As further proof of concept, we extend this approach to a regression task using non-imaging data (UCI Abalone). In all cases, we show that out-of-the-box model performance increases after SAGE score filtering, even when applied to data from the model's own training and test datasets. Our out-of-distribution scoring method can be introduced during several steps of model construction and assessment, leading to future improvements in responsible deep learning implementation.

3.2 Introduction

The presence of generalization gaps, where machine learning performance degrades when a trained model encounters previously-unseen data, represents a critical ongoing challenge in the implementation of AI systems.[117,118] Model performance may suffer when the underlying distributions of input features for new data shift away from those learned during the training process. A baseline method for monitoring predictive uncertainty in neural networks without retraining is the maximum softmax prediction probability[31], where the highest output node value may decrease for out-of-distribution data points. While this technique has been improved with calibration of the softmax probabilities via temperature scaling[35,119], the approach has proved unreliable with increasingly-deformed input features and can be erroneously overconfident when predicting on unrecognizable images.[75,120] Ensemble methods have been proposed to improve the reliability of uncertainty measures, but this requires the simultaneous training of *m* networks instead of a single model, increasing computational overhead.[37]

Whereas neural network prediction confidence is a black box measure of data similarity, there exist simple-to-understand visualization methods such as UMAP[121] and t-SNE[122], allowing users to examine similarity of points in high-dimensional space by localization patterns in two or three dimensions. While the resulting plots are appealing and easy to digest, the local and global structure of the data can become distorted by these methods of compression, limiting the effective use of distances as an "all-in-one" measure of data similarity.[123] Furthermore, these dimensionality-reduction techniques are not reproducible without degrading algorithmic performance (i.e. no multi-threading) or perpetuating the random initialization state.[124] Rabanser et al. introduce a method for quantifying dataset differences via dimensionality-reduction by embedding both a reference and novel dataset before statistically comparing the resulting distributions.[41] Other recent approaches use deep embeddings to calculate a latent distance metric for identifying out-of-distribution data, a key advantage of which is the ability to discriminate individual samples instead of reporting whole dataset statistical differences.[38,125]

The tradeoffs between explainability, quantifiability and robustness have thus far been barriers to a consensus approach to determining which individual samples are appropriate to use for a given machine learning model. We therefore propose the use of Supervised Autoencoders for Generalization Estimates (SAGE) as a standardized approach to uncertainty estimation that draws from the strengths of previously-described methods.[17] SAGE scoring is introduced as a dataset companion which allows for the uncoupling of uncertainty estimation from downstream prediction tasks with separate, more complex models. We calculate a combined out-of-distribution score using three model-intrinsic measures of uncertainty and show examples of outlier detection for classification tasks using MNIST and CIFAR-10 and a regression task using the UCI Abalone dataset. Finally, we show how filtering outliers using the combined out-of-distribution score improves generalization to separate, stronger classification and regression models, even with perturbed and corrupted data.

3.3 Results

3.3.1 SAGE Embedding Space for Original and Corrupted Images

We demonstrate a supervised autoencoder (SAE) framework for faithfully encoding MNIST training data images in two dimensions (Figure 3.1A), with low error in image reconstruction across held-out test images (Δ mean squared error = 0.005, *n* = 8,000). (Figure 3.1B) Designed in part to capture digit identity in the latent space through multitask learning, the model also demonstrates excellent classification performance on held-out test images (f1 = 97.9). (Figure 3.1C) Moreover, the latent space distribution of MNIST training data is closely approximated by encoded MNIST test data, which is drawn from the same original image distribution. (Figure 3.1D) Note that we observed local differences in data density across the embedded latent space, along with local differences in average reconstruction error and calibrated classification confidence (Figure 3.2).

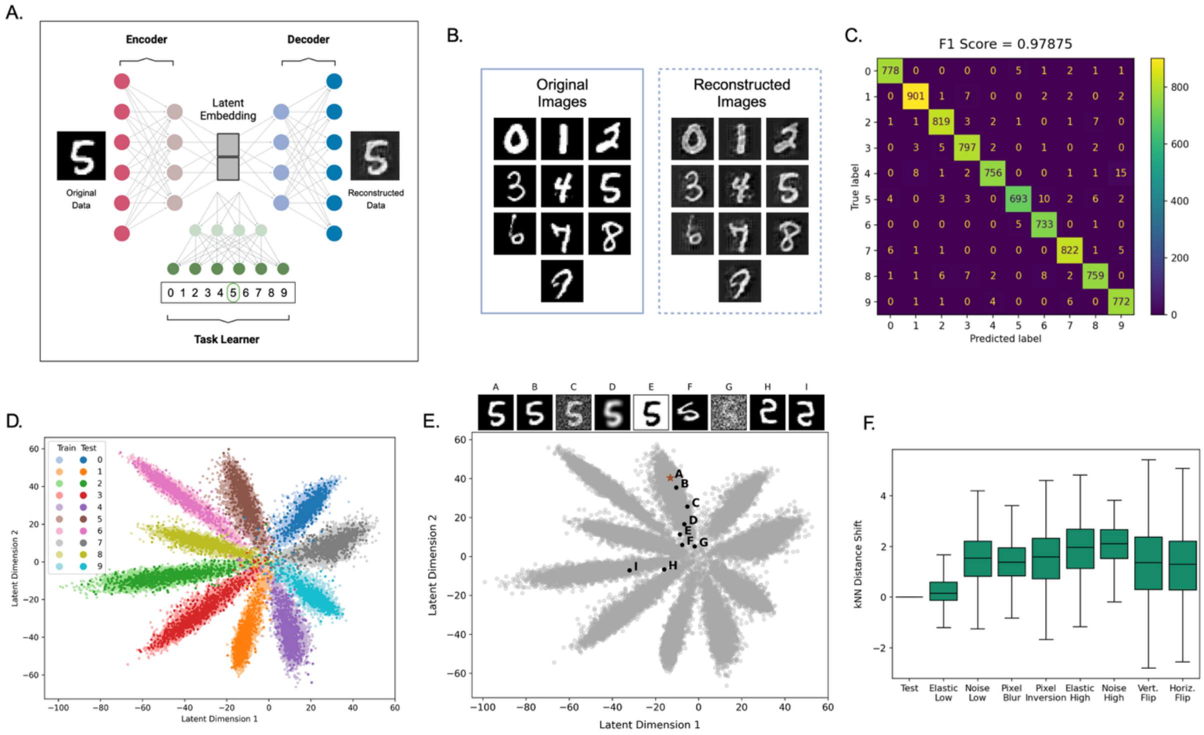


Figure 3.1 – SAGE model overview. A) Schematic of SAGE architecture with example MNIST input. B) Decoder reconstructions of MNIST images. C) Confusion matrix of SAGE classifier accuracy on MNIST test images. D) Overlayed scatter plots of 2D latent embedding space for MNIST train and test images, colored by image class. E) Examples of transforms applied to MNIST test images and subsequent embedding locations overlayed with train data latent space. Brown star indicates original image. F) Box plots showing MNIST test kNN distance changes after application of image transforms.

Importantly, modified test images (i.e. transformed data that intentionally deviate from native MNIST examples) mapped to different areas of the latent space, with increasing severity of transform mapping to lower density, lower confidence regions of the latent space. (Figure 3.1E) Indeed, each transformation of test images results in measurable deviations from the original latent space encoding, with increased distance to the k-nearest training points (Figure 3.1F), increased image reconstruction error, and decreased calibrated classifier confidence. We found that minimally transformed images (e.g. “low” elastic deformation) tend to map closely to the original image set, whereas larger deformations (e.g. “high” elastic deformation) are significantly more distinct in their latent space embedding. (Figure 3.1E, F). We identified a small minority of so-called “imposter” transformations, where vertical or horizontal geometric transforms resulted in effective misclassification (e.g. vertically-flipping a ‘5’ will be read as ‘2’) (Figure 3.1E) and removed such imposters from subsequent analysis.

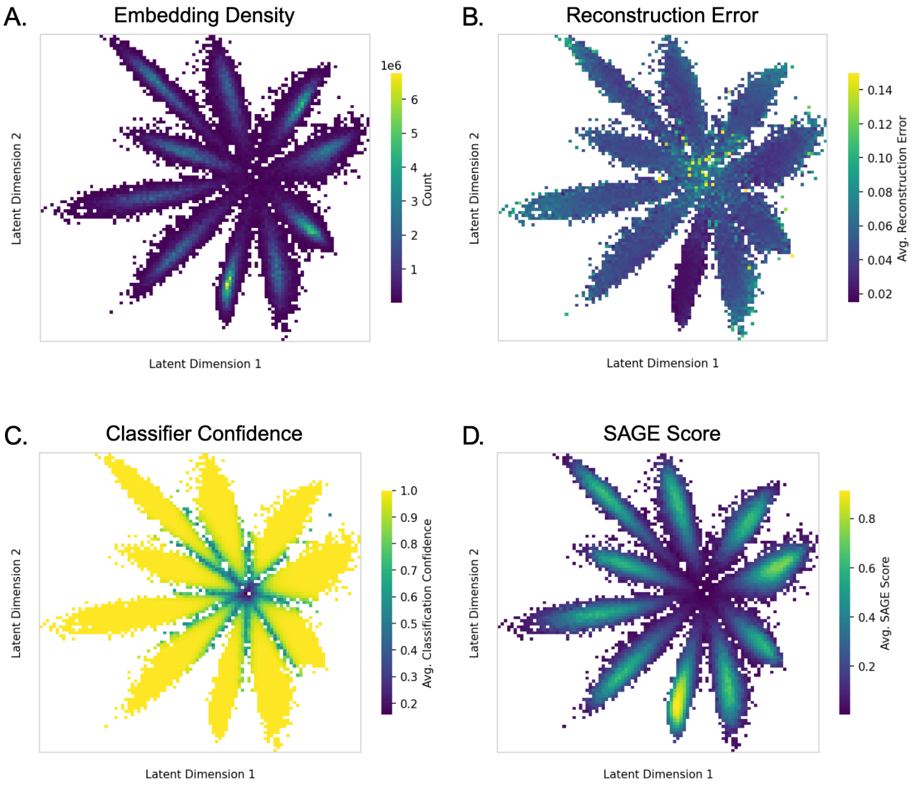


Figure 3.2 – Binned output measures of a SAGE model trained on MNIST images. Latent dimensions were split into 100 x 100 blocks with bin color intensity depicting key SAGE output metrics.

3.3.2 SAGE Scoring Separates Transformed Images

Recognizing that latent space density (as assessed by kNN distance), reconstruction error, and classifier confidence are all distinct and largely independent phenomena, we created an ensemble score using the combined exceedance probabilities of new test data with respect to the training data distributions of SAGE output measures. (Figure 3.3A, Methods)

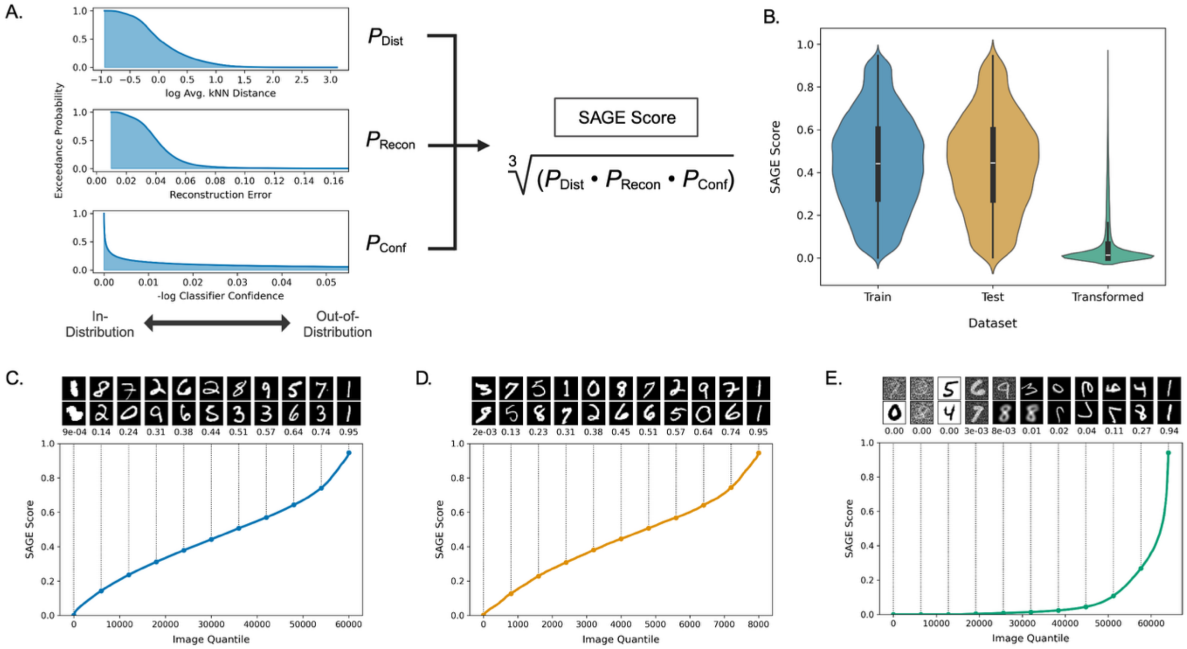


Figure 3.3 – SAGE scoring process and calculated values for train, test and transformed test MNIST images. A) Example of train image SAGE output distributions converted into exceedance curves. For any image, its score is calculated as the geometric mean of the three SAGE output exceedance probabilities w.r.t training distributions. B) Violin plots of SAGE scores for MNIST train, test and transformed test images. C-E) Line plots show SAGE score values across dataset quantiles sorted from low to high. Image examples and score values are displayed above each decile.

This ensemble approach clearly separated original MNIST train and test sets from transformed data (mean SAGE score train = 0.444, test = 0.441 and transformed = 0.075). (Figure 3.2B) We found that the lowest SAGE scores identify outlier images among MNIST training and testing data (Figure 3.2C-D), and are particularly discriminative for transformed images, where the majority of scores were zero or near-zero. Milder image transforms of MNIST, such as "low" elastic deformation, are prominent only towards the 90th percentile of SAGE scores for the transformed test dataset, where scores begin to increase appreciably. (Figure 3.2E) The lowest probability scores were associated with high degrees of pixel intensity changes including pixel inversions and heavy Gaussian noising.

3.3.3 Removal of Low-Score Data Improves Separate Classifier

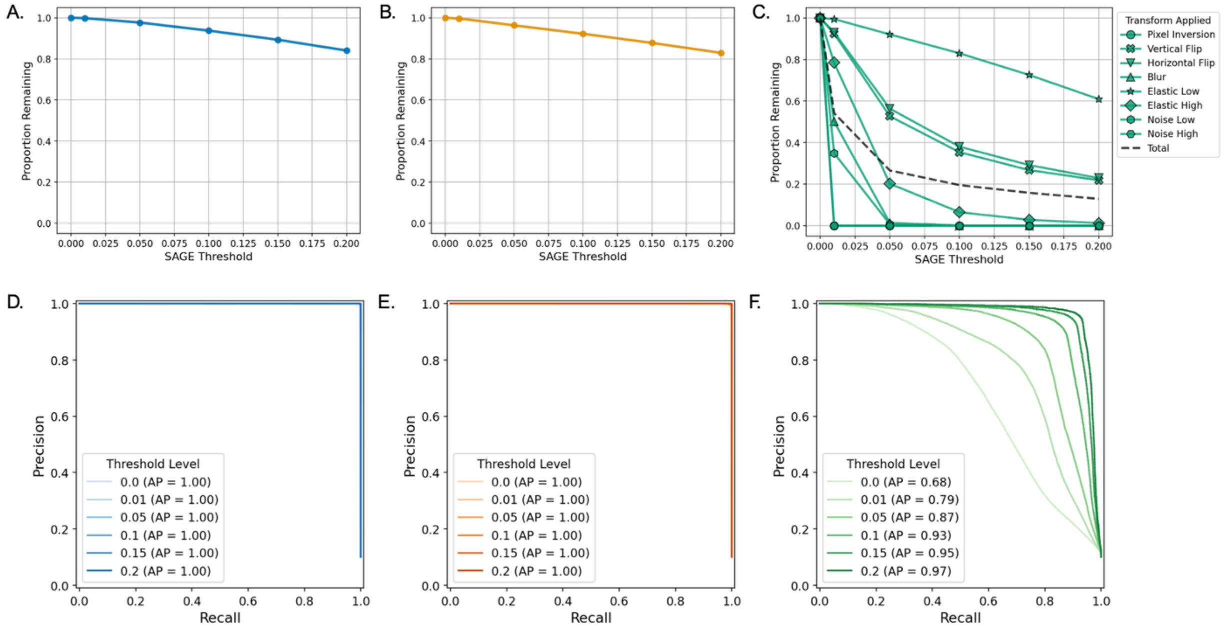


Figure 3.4 – Effects of SAGE score filtering on performance of a separate ResNet18 MNIST classifier. A-C) Line plots of proportion of MNIST images remaining after filtering examples below 6 SAGE score threshold values. D-E) Precision-recall curves plotted for train, test and transformed test datasets after filtering images at 6 SAGE score thresholds.

To improve out-of-the-box performance of an independent ResNet18 model trained on MNIST, we sought to leverage SAGE score as a data filter to ensure similarity of input data to the model's own training data. We first demonstrate that our combined score succeeds as a tunable filter selectively distinguishing outliers (transformed data) while preserving original training and testing data (Figure 3.4A-C), with filter threshold values corresponding inversely to anticipated original dataset retention (e.g. threshold of 0.1 retains 93.7% of the train dataset).

We next note that the independent ResNet18 model performs exceedingly well on MNIST held-out test data (f1 = 0.99) but shows degraded performance on transformed data as expected (f1 = 0.76); whereas, transformed dataset performance improves significantly with even mild SAGE score filtering (e.g. f1 = 0.90 for threshold of 0.05). (Figure 3.4D-F) We also note that out-of-the-box model performance can be improved even for data used during the training process, with increasing filter stringency improving observed accuracy. (Figure 3.5).

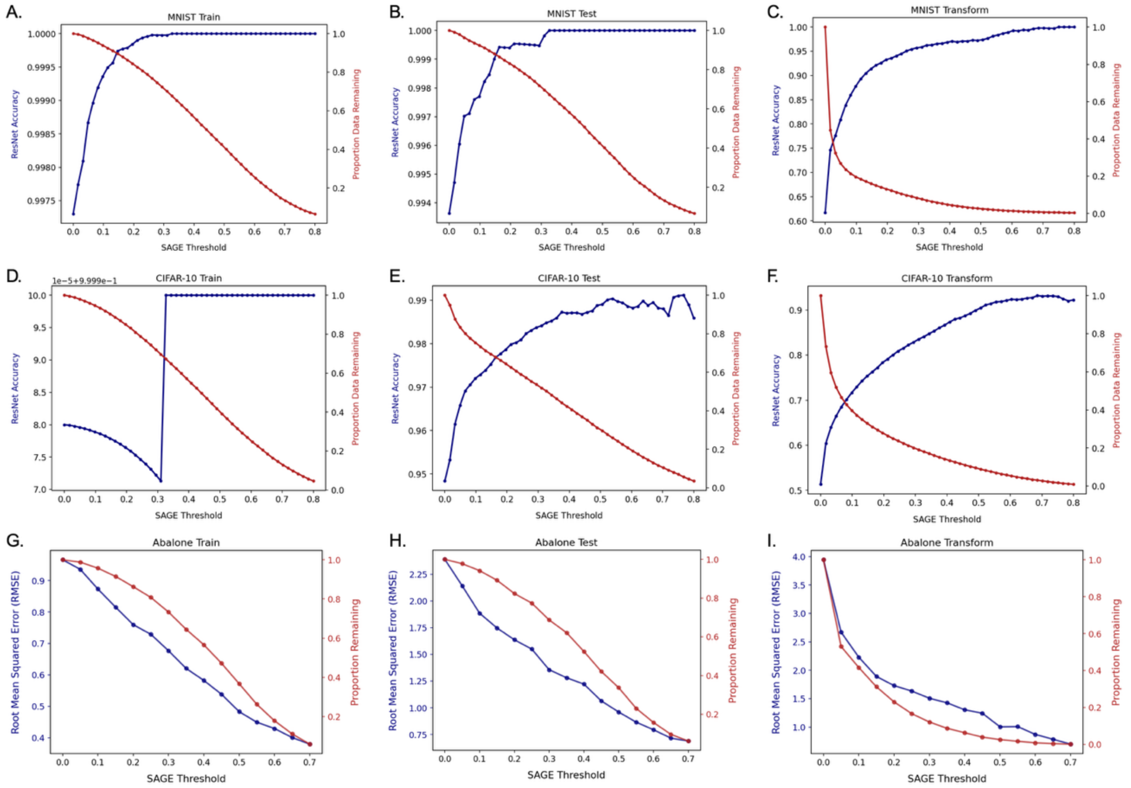


Figure 3.5 – Prediction accuracy improvement as train, test and transformed samples are removed below a series of SAGE score thresholds. A-C) MNIST, D-E) CIFAR-10, G-I) UCI Abalone.

3.3.4 Applying SAGE to RGB Images

Given the relative simplicity of MNIST as a use case, we next sought to apply our approach to a more complex image classification task using the CIFAR-10 dataset, which contains three color-channel data (3,072 RGB pixel values per image). We applied a panel of image perturbations of different intensities to the original CIFAR-10 test dataset (Figure 3.4A) and demonstrate that a 2-dimensional latent space embedding, in this case using a deeper architecture and introducing a contrastive loss term, can faithfully encode distinct image clusters in this dataset. (Figure 3.6B) Reasoning that increasing dimensionality of the latent space could further improve performance, we demonstrated a reduction in overall training loss up to 16 dimensions, after which performance plateaued. Applying the trained 16-dimensional SAGE model to the train, test and transformed test CIFAR-10 image sets, we reproduce our findings from MNIST, where increasing SAGE score identifies increasing severity of image transformation, while minimal transformations (e.g. horizontal flip) behave similarly to untransformed test data. (Figure 3.6C-E) Importantly, SAGE score-based filtering improved performance of a separate out-of-the-box ResNet34 model pre-trained on CIFAR-10, particularly when applied to transformed images (average precision (AP) of 0.86 with SAGE and 0.44 without SAGE, 0.2 threshold value. (Figure 3.6F-H)

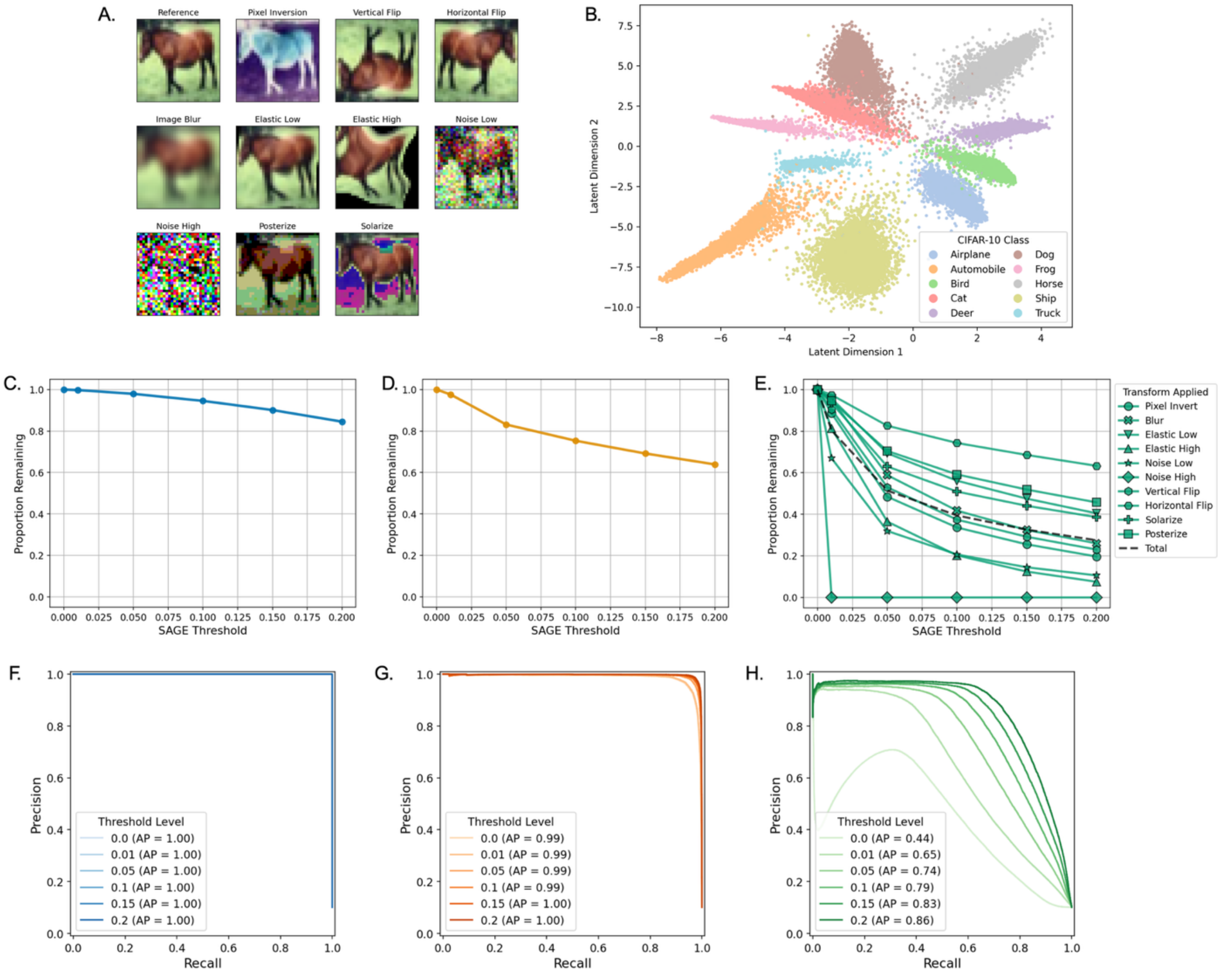


Figure 3.6 – Training and evaluating SAGE on CIFAR-10 images in RGB color channels. A) Image examples CIFAR-10 image of class 'Horse' with test transforms applied. B) Scatter plot of 2D latent embedding space of CIFAR-10. C-E) Line plots of proportion of CIFAR-10 images after filtering images below 6 SAGE score threshold values. SAGE scores calculated from 16D latent embeddings. F-H) Precision-recall curves corresponding to train, test and transformed test CIFAR-10 images after thresholding at six score values.

3.3.5 SAGE Improves Performance on Regression Task

Finally, we sought to explore the potential of this approach for regression tasks. For proof-of-principle, we fit an SAE model to the UCI Abalone dataset, compressing the input feature space down to a single latent dimension. (Figure 3.7A) As before, SAGE scores for most transformed data points revealed significantly lower values compared with the training data set distribution (Figure 3.7B), and filtering based on score demonstrated favorable exclusion of transformed data with relative retention of training and testing data. (Figure 3.7C-E) Finally, we applied score-based filtering of input data to a separate random forest regression model trained on the original training dataset, demonstrating improved root mean squared error (RMSE) of predictions with increasing threshold values, including for samples within the original train and test sets. (Figure 3.7F-H)

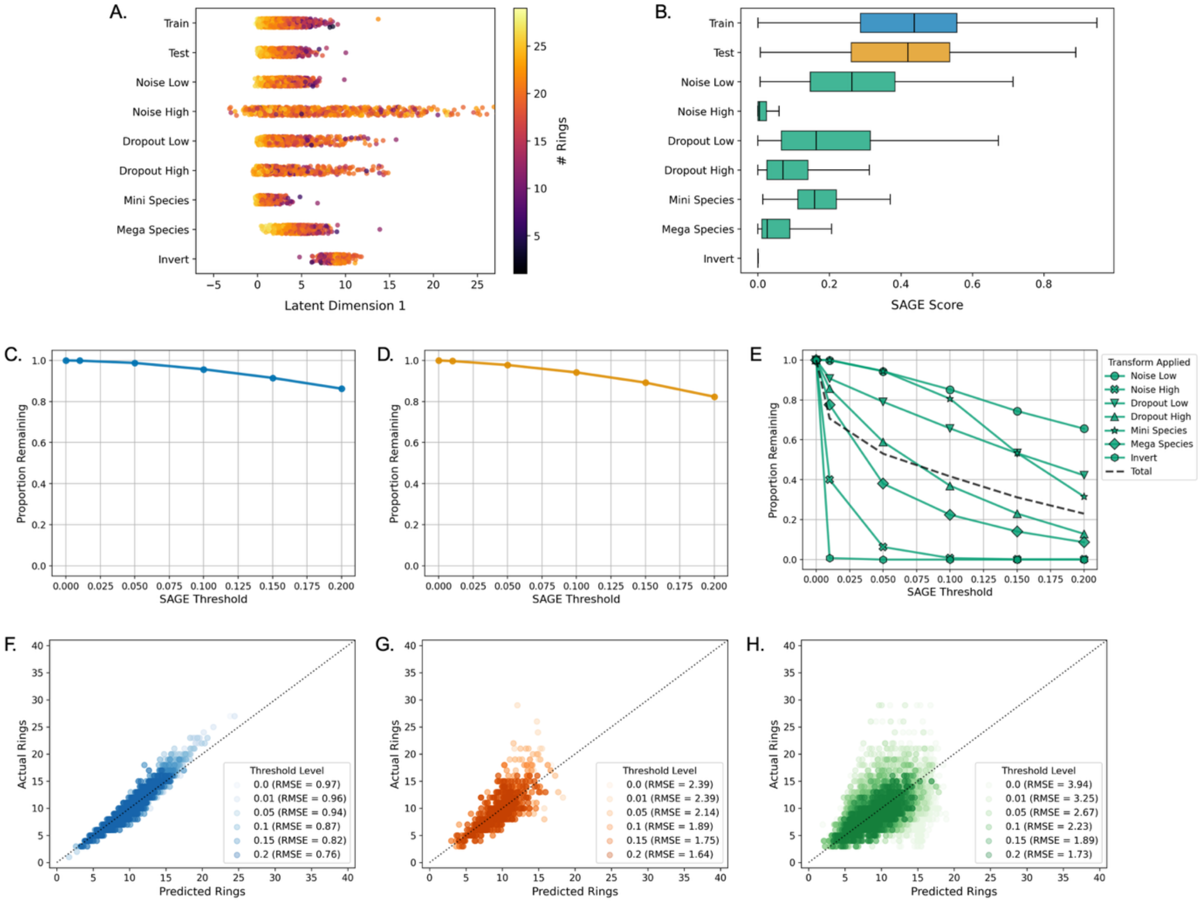


Figure 3.7 – Analysis of SAGE with a regressor module for predicting the inner-shell rings of Haliotis rubra. A) Scatterplots of 1D latent embeddings for train, test and transformed test phenotype data. Color gradient indicates ground-truth number of inner-shell rings. B) Box plots of SAGE scores for train, test and transformed test samples. C-E) Line plots show proportion of samples remaining after application of 6 SAGE score thresholds. F-H) Scatter plots of predicted rings versus actual values using separate regression model. Root mean-squared error (RMSE) of regression shown beside each SAGE score threshold.

3.4 Discussion

We present here a flexible, model-agnostic, dataset-focused approach for prospective detection of out-of-distribution data points. We demonstrate the SAGE score's potential use as a selective filter of input data prior to model application, with several advantages over existing techniques. Moreover, SAGE is applicable to different datasets and tasks, including both classification and regression, and has potential implications for model development (e.g. via outlier identification and model refinement), refining or adapting existing models to new data, and supporting regulatory review and post-market surveillance. To our knowledge, this is the first approach that standardizes generalizability estimation across modalities and tasks while prioritizing interpretability and remaining sensitive to covariate shifts in the underlying data.

We chose a supervised autoencoder as the backbone of the SAGE approach because of its ability to yield an interpretable, class-separable latent space and due to its relatively small compute requirements which should improve scalability compared to bulkier approaches like the variational autoencoder (VAE). The SAGE score is an ensemble metric that combines three independent measures of out-of-distribution estimation, compensating for the relative weaknesses of each component. For example, the model encoder outputs latent embeddings which allow for visualization in a low-dimensional space but does not exclusively rely on compressed distances as a measure of similarity, an attribute that has been shown to be problematic for popular methods like t-SNE and UMAP. Data reconstructions, assessed by mean squared error, can also yield misleading results as is the case when CIFAR-10 images are subjected to a Gaussian blur transform. While blurring results in a lower overall reconstruction error than that of original train images, higher kNN distance to training points in the embedding space and lower classification confidence allow these images to be recognized as out-of-distribution. Furthermore, unrecognizable images such as unnormalized Gaussian noise in RGB color channels exhibit perfect classifier confidence but are easily detected and removed by a combination of reconstruction error and kNN distance.

Despite these strengths, we note several limitations to our work. First, we do not perform exhaustive benchmarking of SAGE against datasets featuring realistic or naturally-occurring distribution shifts. We furthermore do not compare our approach against state-of-the-art practices for uncertainty quantification from the machine learning literature or employ SAGE as a binary out-of-distribution vs. in-distribution classifier that could be useful in automated decision-making pipelines. Our study focuses on classification and regression with benchmark imaging and biological datasets as our primary machine learning tasks, neglecting any number of other common problems and data modalities (e.g. image segmentation or time-series forecasting). We also concede the potential to further improve SAGE performance through increasing model size, complexity, and encoder pretraining, as well as alternative or additional architectures. For instance, the inclusion of Bayesian dropout for neural network classifiers could improve variational inference without the need for retraining pre-existing models.[36,126] Other approaches for Bayesian inference have been suggested for neural network regression, and could be similarly applied.[127]

Furthermore, we do not perform data augmentation before training and our method can therefore be considered a form of normative modeling.[128] Prior work by Hendrycks et al.[129] has shown that inclusion of few augmented examples during training can improve the robustness of subsequent classifier confidence measures to outliers, a simple method that negates the use of more expensive generative models to create synthetic data.[130] Upsampling training data that has a low similarity score to itself could further augment the training process and improve generalization in a complementary manner, however, these iterations are considered out-of-scope in this proof-of-principle study.

We further note that SAGE scoring is unable to distinguish "imposter" data examples (e.g. where a vertical flip of a '5' in MNIST may be mistakenly recognized as a '2'). We did not observe any such instances within our transformed test sets of CIFAR-10 and therefore expect this phenomenon to be rare in real-world applications as images increase in complexity. Importantly, we also note that SAGE may expose sensitive, private, and/or proprietary details about a model's training dataset through the retention of both encoder and decoder elements in addition to the full latent space embedding. We envision the possibility of privacy-preserving implementations of this work but note that these are out-of-scope in the current study. Future work will focus on the extension of out-of-distribution estimation to a wider range of tasks and modalities, including more complex biomedical imaging datasets, and the inclusion of improved measures of intrinsic uncertainty.

3.5 Methods

3.5.1 Datasets

The MNIST[131] dataset was downloaded using the torchvision package (version 0.17.2). MNIST consists of 28 x 28-pixel grayscale images of handwritten digits (0 - 9) and comes pre-split into training (*n* = 60,000) and testing (*n* = 10,000) sets, with 6,000 and 1,000 images per class respectively. We randomly divided the test set into class-balanced, held-out test (*n* = 8,000) and validation (*n* = 2,000) sets in order to set aside images for classifier calibration.

The CIFAR-10[132] dataset was downloaded using torchvision and consists of 32 x 32 pixel RGB color images of ten vehicle and animal classes. Like MNIST, CIFAR-10 is pre-split into a training (*n* = 50,000) and testing (*n* = 10,000) set which we randomly subdivided further into held-out test (*n* = 8,000) and validation (*n* = 2,000) sets, ensuring class balance. The image classes consist of: 'Airplane', 'Automobile', 'Bird', 'Cat', 'Deer', 'Dog', 'Frog', 'Horse', 'Ship', and 'Truck'. Importantly, 'Automobile' and 'Truck' vehicle classes consist of only cars and tractor-trailers, respectively, to reduce label overlap, whereas 'Airplane' and 'Ship' consist of different grades of planes (e.g. commercial passenger jets, military jets) and watercraft (e.g. leisure boats, commercial shipping vessels). All animal classes include multiple species or breeds. CIFAR-10 images also exhibit a variety of naturally occurring viewer perspectives and subject color patterns, lending to the increased complexity of this dataset.

The UCI Abalone dataset was downloaded from the UC Irvine Machine Learning Repository website (https://archive.ics.uci.edu/dataset/1/abalone) and is included in our project repository as a CSV file. The dataset was adapted from a 1994 technical report[133] and consists of 4,177 examples of 8 animal phenotypes and body measurements including Sex, Length, Diameter, Height, Whole Weight, Shucked Weight, Viscera Weight, and Shell Weight, with the number of inner-shell rings representing the ground-truth labels. We split examples into training (80%), held-out test (16%) and validation (4%) datasets.

3.5.2 Data Transformations

We built and applied a panel of image transformations to MNIST and CIFAR-10 held-out test images using the v2 transform module of torchvision's library. The panel included a 100% horizontal flip, 100% vertical flip, 100% pixel value inversion, Gaussian blur (kernel size = 5, sigma = 2), Gaussian noise ("low", sigma = 0.2; "high", sigma = 0.8) and elastic stretching ("low", alpha = 50; "high", alpha = 200). For CIFAR-10 we included two additional photometric transformations: a 100% solarize filter (threshold = 0.75) and 100% posterize filter (bits = 2). All MNIST and CIFAR-10 images were converted to torch float32 data types, scaled and normalized using the following values before use in training and analysis: MNIST (mean = [0.1307], std dev = [0.3081]), CIFAR-10 (mean = [0.4914, 0.4822, 0.4465], std dev = [0.247, 0.243, 0.261]).

For the UCI Abalone dataset, we introduced custom transformations of the testing data including the random addition of Gaussian noise (Low $\sigma = 0.05$, High $\sigma = 0.5$), inverting features (1 - feature value), randomly dropping feature columns (Low $n = 1$, High $n = 3$) and multiplying the features by a factor of 2.0 or 0.5 to simulate abalone species with larger (factor of 2) or smaller (factor of 0.5) body proportions while keeping the number of rings constant. All features were standardized by removing the train set mean and scaling to unit variance before training and testing our model.

3.5.3 SAGE Model Architecture

All supervised autoencoder models consisted of a neural network encoder, a neural network decoder and third task-focused neural network module. All models were built using PyTorch (version 2.2.2) and python (version 3.10.14).

For MNIST, we constructed an encoder module with two convolutional layers (kernel size = 3, stride = 1, padding = 1) followed by 2D batch normalization and max pooling (kernel size = 2). The last two layers of our encoder were fully connected from the flattened output of max pooling. The classifier module consisted of a two-layer fully connected network using the encoder's latent embedding as its input, with 20 and 10 layers respectively. The decoder architecture for MNIST mirrored the encoder, with two fully connected layers followed by unflattening and max un-pooling (kernel size = 2), after which two de-convolutional layers (kernel size = 3, stride = 1, padding = 1) return the original image size ([batch, 1, 28, 28]). All layers are followed by a Leaky RELU activation function, and we use dropout (p = 0.2) between convolutional/de-convolutional layers.

For CIFAR-10, we instantiated a ResNet18 model from PyTorch with default ImageNet pre-trained weights as the encoder module. We re-initialized the last fully connected encoder layer before training. The classifier module consisted of two fully connected layers using dropout (p = 0.2), with 20 and 10 layers respectively. The decoder contained a fully connected layer with 1,024 nodes followed by unflattening and three de-convolutional layers (kernel size = 4, padding = 1, stride = 2). Like MNIST, we use Leaky RELU activation for all three modules.

The UCI Abalone model features a four layer, fully connected encoder and decoder module each followed by Leaky ReLU activation and dropout (p = 0.2) with 64, 32, 16 and 1 nodes respectively. The regressor module consists of three fully connected layers with a single output node and no activation function using 32, 16, 8 and 1 node layers. All non-final layers of the encoder, decoder and regressor use Leaky RELU activation.

3.5.4 Model Training

Training was performed on a laptop with a 6-core CPU and 32GB of RAM. For MNIST, we trained our supervised autoencoder model over 20 epochs with early stopping. We used an Adam optimizer with a learning rate of $3x10^{-4}$ and batch size of 64. Decoder loss was measured using mean squared error (MSE) loss and classification loss was measured using cross-entropy loss. The total loss was calculated as the unweighted sum of the decoder and classifier loss terms. We utilized the pre-split MNIST training set (*n* = 60,000) to fit the model without inclusion of any image transformations.

For CIFAR-10, we implemented a two-stage training process, each occurring over 10 epochs (20 epochs total) with a batch size of 64. The first stage only involved training the encoder and classifier weights with an Adam optimizer with a learning rate of $3x10^{-4}$. We used cross entropy loss to quantify classification error and included a center loss term down-weighted by a coefficient ($\alpha = 0.1$). We randomly-initialized a cluster center coordinate for each of the 10 classes. The first training phase maximized the distance between cluster centers, yielding improved latent separation of the image classes. For the second stage, we trained the encoder, decoder and classifier using a second Adam optimizer and learning rates of 1x10^{-4}, 3x10^{-4} and 1x10^{-5} respectively. We used different learning rates within the stage two optimizer to allow for the simultaneous training of the decoder and preservation of the latent embedding structure established during the first stage. The decoder and classifier loss terms were quantified using the MSE loss and cross entropy loss respectively. The total loss for stage two was calculated as the unweighted sum of decoder and classifier error.

Our UCI Abalone model was trained over 100 epochs with an Adam optimizer and a learning rate of 3x10^{-4}. We used MSE for both the decoder and regressor loss functions, and the total loss was the unweighted sum of these terms.

3.5.5 Model Calibration

After the training process for MNIST and CIFAR-10, we calibrated the autoencoder classifier modules with temperature scaling. For each dataset, we classified all validation set images using the trained models and divided the raw logits by a tunable parameter, *T*, in order to align model predictions with the true likelihood of correct predictions. We used cross entropy loss and a L-BFGS optimizer (learning rate = 0.01, batch size = 64) to tune *T* over one epoch for each model.

3.5.6 k-Nearest Neighbors Distance

The training split for each dataset was designated as the 'reference' embedding for both classification and regression analyses. The reference data was compressed using the trained model encoder and a Balltree[134] was fit to the resulting latent space. Each test example underwent the same encoding process, and the tree was queried using the latent coordinates to determine the average distance to the point's k-Nearest Neighbors (kNN). For MNIST and CIFAR-10 datasets, $k = 100$ whereas for the UCI Abalone dataset $k = 20$.

3.5.7 SAGE Scoring

Let $x = (x_1, x_2, x_3)$ represent the observed output values for a given image, corresponding to the three SAGE model measures:

1. x_1: L1 latent embedding distance to the k nearest training neighbors
2. x_2: Softmax classifier confidence (argmax)
3. x_3: Reconstruction error

Let X_i be the random variable denoting the distribution of the model's output for measure i across the training data and let $F_i(x) = P(X_i \leq x_i)$ denote the cumulative distribution function (CDF) of X_i.

We define the **exceedance probability** for measure i as:

$$E_i(x_i) = P(X_i > x_i) = 1 - F_i(x_i)$$

The **SAGE score** for an image is then computed as the **geometric mean** of the three exceedance probabilities:

$$SAGE(x) = \left(\prod_{i=1}^{3} E_i(x_i) \right)^{1/3}$$

3.5.8 Evaluation with Pre-trained ResNet Models

Pre-trained ResNet models for MNIST and CIFAR-10 were initialized using the timm[135] (version 1.0.12) library and incorporated into our workflow for assessing the effects of filtering data points based on SAGE score thresholds. We did not make any modifications to these models which were used as out-of-the-box classifiers on the train, test and transformed datasets.

3.5.9 Random Forest Regression

The UCI Abalone training set was used to train a separate random forest regressor model from scikit-learn[113] (version 1.4.2). We performed grid search cross-validation to determine the best model parameters, testing a variable number of estimators ([25, 50, 75, 100]), tree depths ([5, 10, 15, 20, 40]) and maximum features ([2, 4, 6, 8]). The best model had 50 estimators, a tree depth of 15 and used a maximum of 2 features. Regression error was assessed as the root mean square error (RMSE) between the number of inner-shell rings and predicted values for the train, test and transformed test sets.

3.5.10 Score Thresholding and Performance Visualization

SAGE scores were calculated for all examples in the MNIST, CIFAR-10 and UCI Abalone datasets as described above. For each set, data was filtered at six SAGE score values ([0.0, 0.01, 0.05, 0.1, 0.15, 0.2]) where samples greater than or equal to the threshold were retained and all others were discarded. Retained samples were passed to the separate, ResNet or random forest regression models and predictions were recorded. For MNIST and CIFAR-10 we used the scikit-learn 'LabelBinarizer' to one-hot encode labels and 'PrecisionRecallDisplay' to create micro-averaged precision-recall curves from ResNet predictions. We repeated this process for the training, test and transformed test data separately, calculating average precision at each score threshold. Abalone predictions were assessed using sci-kit learn's 'root_mean_square_error' function and visualized as matplotlib scatterplots.

4 Extending SAGE to Skin Cancer Malignancy Detection

4.1 Abstract

Widespread access to imaging technologies and stronger machine learning (ML) architectures for dermatology tasks such as malignancy prediction have spurred a race to develop models to assist in the automated diagnosis of skin cancer. However, high diagnostic performance on benchmarking datasets quickly deteriorates when models are challenged with data from disparate clinical sources. Generalization gaps stem from the high variability in skin lesion images due to lighting, capture angle, imaging technology and patient phenotype among other factors, impeding the safe application of diagnostic ML models in practice. In this study, we apply a novel ensemble uncertainty-estimation approach to detect out-of-distribution skin lesion images from four publicly available datasets across five countries. Using our method, Supervised Autoencoders for Generalization Estimates (SAGE), we quantify likeness of images from patients in Argentina, Brazil and the United States to the popular HAM10000 benchmarking dataset and identify problematic image artifacts that affect the reliability of predictions in a teledermatology setting. We show how filtering images based on SAGE score thresholds can improve the performance of a separate malignancy prediction model and how our approach is robust to variations in image modality and the introduction of new diagnostic classes, providing users with a powerful tool for interrogating key differences between their data and the training distribution of an ML model before clinical implementation.

4.2 Introduction

Skin cancers are the most commonly-diagnosed malignancy worldwide, with deadly melanomas accounting for over 330,000 new cases per year with steadily rising incidence rates.[136,137] However, malignancies represent only a fraction (22%) of all index skin lesions referred to dermatologists for biopsy and review, resulting in a clinically significant tension between diagnostic sensitivity and specificity.[138] Moreover, there are surprising gaps in diagnostic ability between primary care physicians and dermatology specialists, and between dermatologists of varying skill levels or experience, amplified by the use of dermoscopy.[139] Further barriers to skin cancer detection such as low socioeconomic status and rural location combine to yield worse outcomes and exacerbate pre-existing health disparities.[137,140] Fortunately, the combination of widely available imaging technologies and automated detection through machine learning (ML) have the potential to revolutionize skin cancer diagnosis and improve global health equity as a result.[141,142]

Many ML models have been trained to identify cutaneous malignancies within skin imaging datasets.[143–148] However, there are notable differences between highly-curated benchmarking datasets used to train and evaluate ML models and the images encountered in real world clinical settings, often leading to reduced model effectiveness in practice.[139] Even state-of-the-art models suffer from these performance drops; the top 25 entries from the ISIC 2019 Grand Challenge misclassified nearly 50% of previously-unseen skin lesion images despite stellar performance on benchmarking datasets.[149] Ambitious ML-assisted dermatology smartphone phone apps have also been shown to have low accuracy and can therefore pose a harm to users attempting to self-diagnose potentially life-threatening diseases without expert oversight.[150–152] One major source of difficulty in automating skin cancer diagnosis is the variability in imaging technologies like dermoscopes which cannot properly standardize inputs to predictive algorithms – outputs can vary based on whether or not the dermatoscope is polarized and the type of light source.[153,154] Indeed, even the presence of modest artefacts such as blur or blue/red shifted-pixel intensities showed marked decreases to ML performance for both diagnostic and disease management tasks when compared to control images.[72] Beyond device or light-induced inconsistencies, changes in patient phenotypes such as skin phototype have been shown to affect ML model accuracy and represent an ethical dilemma in the deployment of dermatological algorithms to underrepresented populations when trained on benchmarked datasets.[155] It is therefore critically important to assess the similarity of new dermatological images with a model's training data on a case-by-case basis, in order to reduce disparities and identify potentially problematic samples before a model is used in any treatment pipeline.

Some diagnostic dermatology models have attempted to remove corrupted or low-quality samples through out-of-distribution (OOD) detection or by building inherent measures of uncertainty into the prediction task.[156–158] This typically combines uncertainty estimation with another primary task, lacking flexibility in adapting to different tasks or pairing with stronger models as they become available. Other single-pronged approaches to measuring uncertainty such as calibration of classifier softmax outputs can be intrinsically measured at time of prediction, but fail silently under scenarios of data drift.[75] Our previous work has introduced an ensemble approach (Supervised Autoencoder for Generalization Estimates [SAGE]) for image uncertainty estimation that is robust to corruptions and perturbations missed by singular OOD metrics, and can be paired with any downstream model's training data.[159] In this study we implement SAGE to assess the generalization potential of skin cancer detection algorithms across dermatology images from five countries and multiple imaging modalities. We use our SAGE scoring system to improve performance of a pre-trained malignancy predictor, enriching for images that are appropriate versus problematic for ML-assisted diagnosis.[155]

4.3 Results

We obtained four publicly-available skin lesion imaging datasets (HAM10000[160], HIBA[161], UFES[162] and DDI[155]) each with distinct characteristics and metadata (Table 1; Methods 4.5.1). HAM10000, the most widely used dataset for benchmarking, contains only one image size ([600 x 450] pixels) whereas images from other datasets vary in resolution and dimensions (range: [147 x 147] - [4,128 x 3,176] pixels). (Figure 4.1A) HAM10000 features lesions with seven diagnostic classes, five considered benign (actinic keratosis, benign keratosis, dermatofibroma, melanocytic nevi, and vascular skin lesion) and two considered malignant (basal cell carcinoma, melanoma). (Figure 4.1B) Notably absent from the HAM10000 dataset are examples of squamous cell carcinoma, a common form of keratinocytic malignancy that comprises a significant minority of the images present in the HIBA (9.78%) and UFES (8.12%) datasets. Even more substantially, nearly half of all lesions in the DDI dataset (45.12%) represented 36 other diagnoses not present in the HAM10000 dataset. Both dermatoscope and clinical smartphone imaging modalities are included in the HIBA and UFES datasets where the ratio of dermoscopic to smartphone images is 3.67:1 for HIBA and not provided for UFES. (Figure 4.1C)

Using HAM10000 as a primary reference dataset, we trained a SAGE model to simultaneously encode an image into a 32-dimensional compressed latent space vector, reconstruct the original image and predict its diagnostic class from the compressed embedding. (Figure 4.2) We found that the SAGE classifier module performed well on HAM10000 holdout test images (weighted F1 score = 0.842) while the decoder was able to coarsely reconstruct lesion size, shape and pigmentation color.

Additionally, the SAGE encoder latent space could distinguish variation within and between different image sets (e.g. HAM10000, training and testing distributions are highly similar whereas HIBA, UFES, and DDI occupy areas of training data paucity). To quantify conformity between the training and test images, we calculated each image's SAGE score – an ensemble metric for assessing similarity based on the reference distributions for the model's latent embedding distance, classifier confidence and reconstruction error.

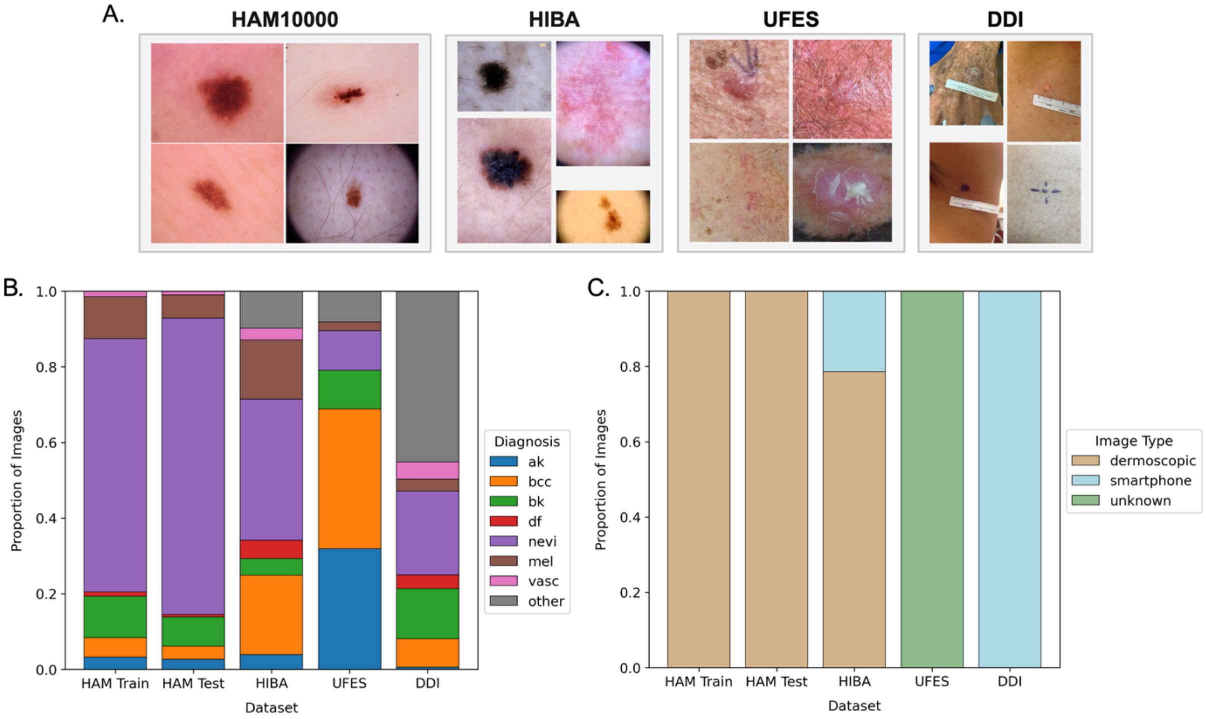


Figure 4.1 – Overview of skin lesion imaging datasets. A) Four randomly-selected image examples from the four main datasets including original photo perspectives. B) Bar charts showing the ratios of diagnostic classes for each dataset. C) Bar charts depicting image modality composition for datasets. (ak –actinic keratosis, bcc – basal cell carcinoma, df – dermatofibroma, nevi – melanocytic nevi, mel – melanoma, vasc – vascular skin lesion)

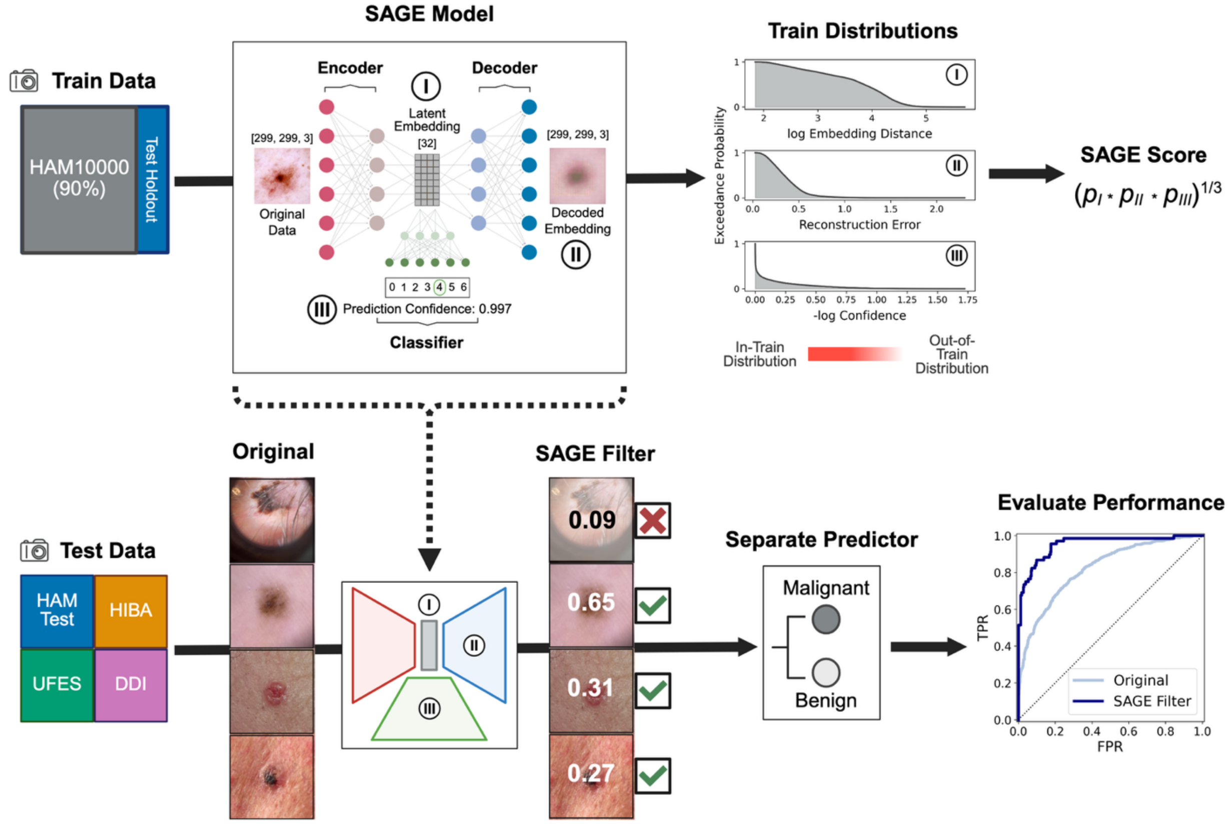


Figure 4.2 – Overview of SAGE model training and scoring process. Images from the HAM10000 training split are used to fit the encoder, decoder and classifier modules. SAGE score is calculated from the train dataset distributions of latent kNN distance, classifier confidence and reconstruction error. The trained model is subsequently used to generate similarity scores for test images, and a filter is applied before predicting with a separate malignancy recognition model.

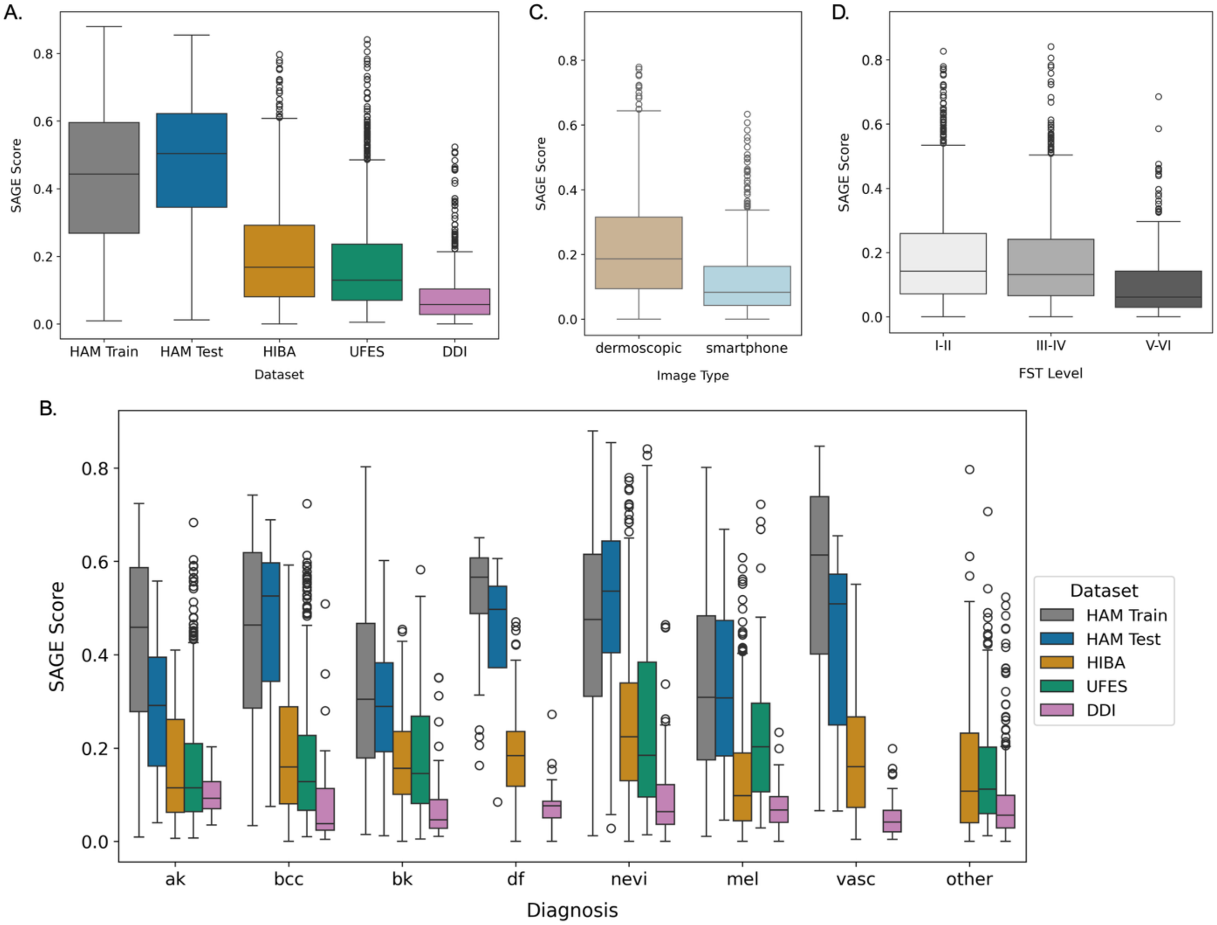


Figure 4.3 – SAGE scores reveal differences between and within dataset categories. A) Distributions of SAGE scores for imaging datasets calculated using the trained model are shown as boxplots. B) Boxplots displaying SAGE score by diagnostic category split by dataset. C) Boxplots of test images from HAM10000, HIBA and DDI showing SAGE score by imaging modality. D) Images from HIBA, UFES and DDI are split according to three FST groupings (I-II, III-IV and V-VI), with SAGE score distributions shown as boxplots. (ak –actinic keratosis, bcc – basal cell carcinoma, df – dermatofibroma, nevi – melanocytic nevi, mel – melanoma, vasc – vascular skin lesion)

The distributions of SAGE scores for HAM10000 testing and non-HAM10000 datasets (i.e. HIBA, UFES, DDI) were significantly lower than the train score distribution, with the HAM10000 test images showing the highest median score (median = 0.50) and DDI the lowest (median = 0.06). (Figure 4.3A) These dataset-level differences were largely consistent across diagnoses. (Figure 4.3B) However, we observed that images associated with diagnoses missing from the HAM10000 dataset had a lower average SAGE score than those corresponding to diagnoses present in the training data (mean = 0.12 vs. mean = 0.22). When we removed images of lesions with diagnoses absent in the HAM10000 dataset from HIBA, UFES and DDI, we expected that differences in SAGE score might be clearly delineated by metadata such as imaging modality and skin phototype. However, we found a surprising degree of overlap in these distributions (Dermoscopic, IQR: [0.09 - 0.32]; Smartphone, IQR: [0.04 - 0.16]; FST I-II, IQR: [0.07 - 0.26]; FST III-IV, IQR: [0.07 - 0.24]; FST V-VI, IQR: [0.03 - 0.14]) pointing to the presence of other distinguishing image attributes that were unidentifiable from the metadata or lesion type alone. (Figure 4.3C-D)

We note that HIBA, UFES and especially the DDI dataset are laden with low SAGE scoring images with many out-of-distribution features not present in the HAM10000 data, such as measuring devices (e.g. rulers), dense hair, skin coverings and other markings. (Figure 4.4C-E) Even within the HAM10000 training and test data we identify uncommon edge cases with low SAGE scores such as lesions cropped by the image frame or bubbles resulting from a lapsed contact between the dermoscope lens and skin. (Figure 4.4A, B) To establish how the presence and severity of out-of-distribution features affect SAGE score, we manually-annotated 12 independent features for all HIBA, UFES and DDI images (*n* = 4,527). (Methods 4.5.5, Table 4.1) Some features had clear negative correlations with SAGE score such as the presence of a ruler or use of camera flash which showed 51.4% and 10.7% reductions compared to baseline, respectively. Similarly, the presence of non-skin background in

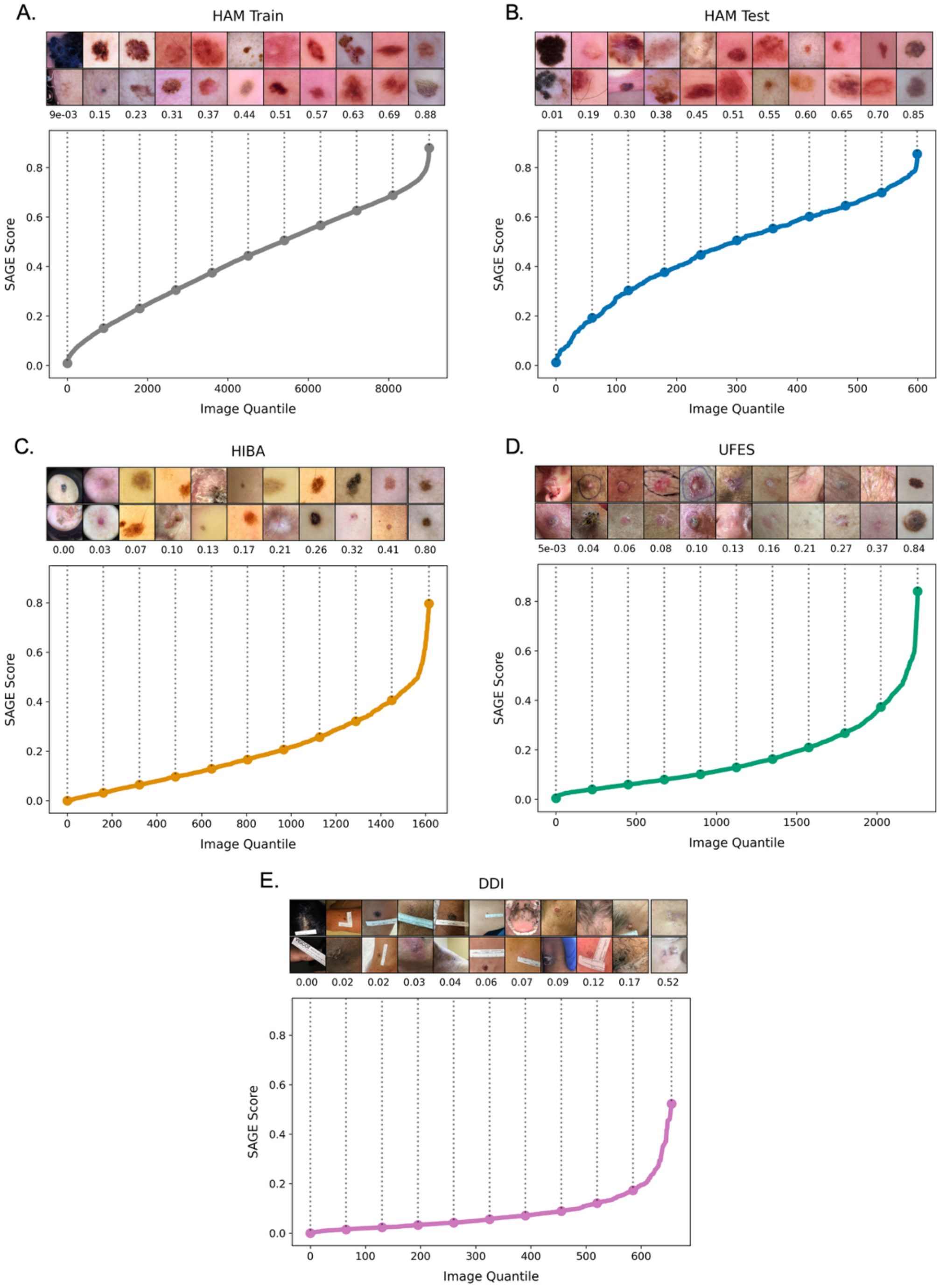


Figure 4.4 – Quantile plots show imaging artifacts associated with low SAGE scores. Images are sorted by SAGE score (y-axis) and plotted by image quantile (x-axis) for the A) HAM10000 Train (n = 9,013), B) HAM10000 Test (n = 600), C) HIBA (n = 1,616), D) UFES (n = 2,255) and E) DDI (n = 656) datasets. For each quantile plot a scatter point marks decile intervals and a vertical dotted line leads to the two nearest SAGE score values and the corresponding lesion images after resizing and center crop transforms have been applied.

an image had a severe negative effect on SAGE score even at low (-44.4%) and medium (-74.5%) intensities. Linking multiple features together with a SAGE score overlay gives a concise description of what constitutes a typical "high-quality" image, such as having no or low amount of hair, high contrast between the lesion and skin and little to no non-skin background. (Figure 4.5A) To that end, we frequently observed compounding negative effects when detrimental image features were stacked such as the 94.7% reduction to SAGE score when both dense hair and non-skin background are co-present (mean = 0.01) vs. a 63.2% decrease for dense hair only (mean = 0.07) and a 68.4% decrease for majority non-skin background alone (mean = 0.06). (Figure 4.5B) We also noted that the presence of some image features did not affect patients with different skin phenotypes equally. For instance, inclusion of a ruler had a 38.9% reduction to average SAGE score in patients with light skin (FST I-II) whereas this decreased by 75.0% for patients with dark skin (FST V-VI), possibly resulting from the higher contrast between the illuminated measuring devices and darker skin. (Figure 4.5C) Intriguingly, score differences between FST levels for test images were largely ablated after low-quality images were removed, illustrating that some extraneous image features are possibly a larger detriment to image quality than changes to patient skin phototypes. (Figure 4.6)

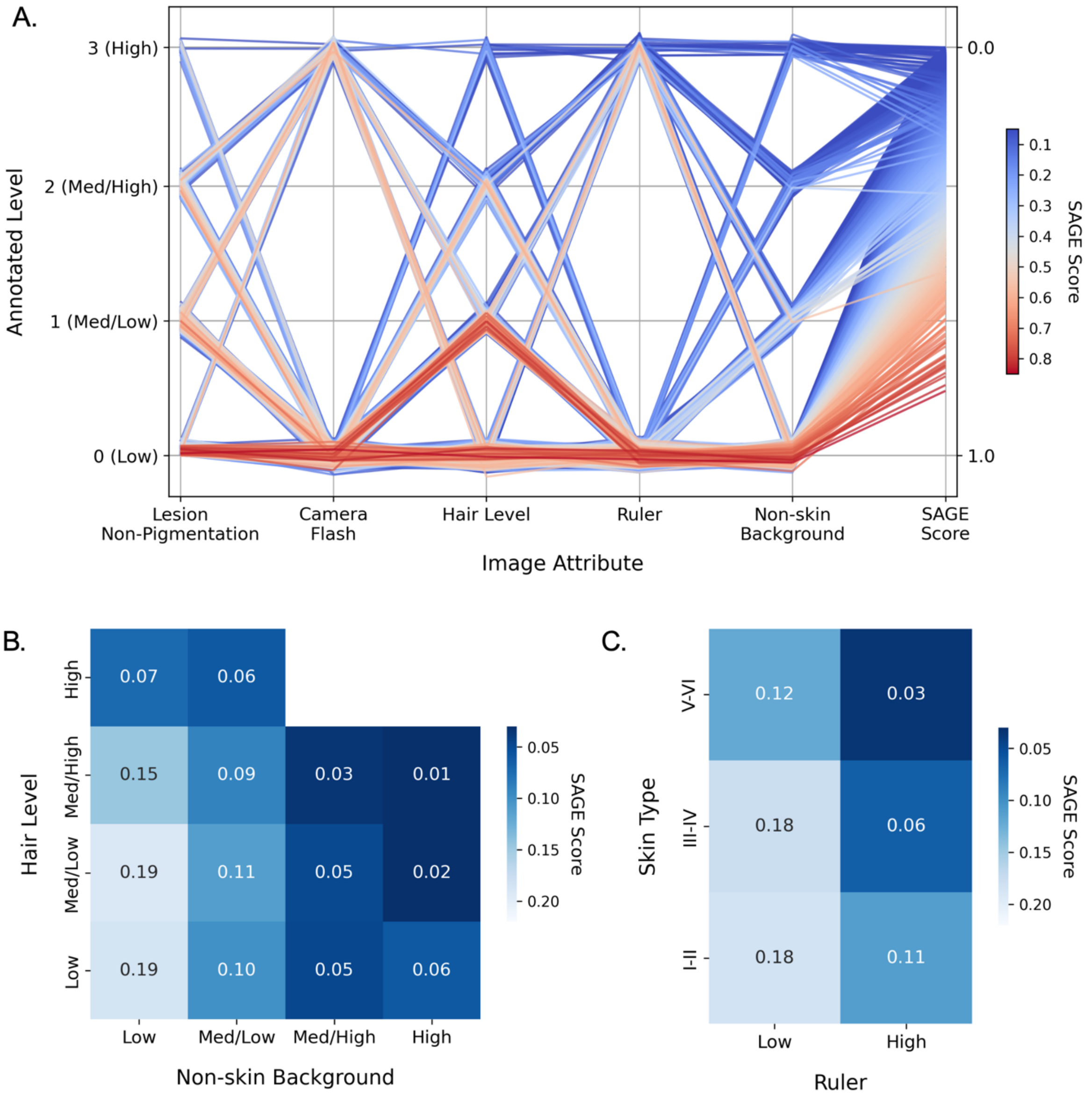


Figure 4.5 – Manually-annotated image feature associations with SAGE score. A) Parallel coordinates plot for test images (n=3,889) with select image features shown along the x-axis with normalized feature level plotted on the left y-axis. Right y-axis shows SAGE score from low (top) to high (bottom). B) Heatmap of SAGE score by hair density (y-axis) and non-skin background (x-axis) feature severity. C) Heatmap of SAGE score by FST level and presence of a measuring device (ruler).

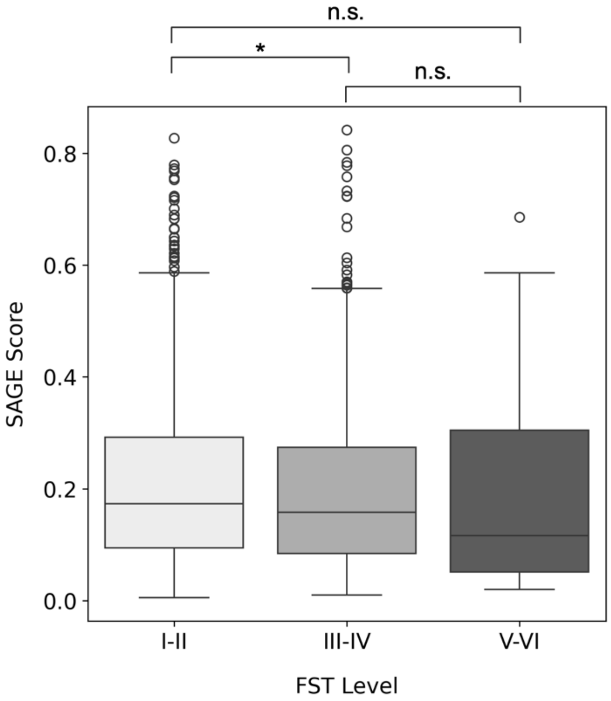


Figure 4.6 – Comparison of SAGE score distributions for HIBA, UFES and DDI datasets after controlling for image quality. Brackets show statistical significance between distributions calculated using the Wilcoxon rank-sum test (α = 0.05). SAGE score differences are not significant (n.s.) between FST levels I-II and V-VI (p = 0.06) and levels III-IV and V-VI (p = 0.19) and significant between levels I-II and III-IV (p= 0.03).

We next evaluated the ability of SAGE score to improve model performance when deployed as a prospective filter. Using a mild SAGE score threshold (0.2), we observed that HAM10000 images were retained at high rates (train = 83.9%, test = 87.2%), whereas the majority of images from other datasets were identified as outliers (e.g. DDI retained only 8.4% of its data). (Figure 4.7A-B) This selective filtering demonstrated significant improvements to overall performance of a pre-trained open source predictor[155] as assessed by area-under the ROC curve (AUC) in both the training (original AUC = 0.96, filtered AUC = 0.98) and mixed test datasets (original AUC = 0.82, filtered AUC = 0.95), with progressively larger performance improvements as SAGE threshold increased to 0.4 (Figure 4.7C-D). Positive predictivity of the mixed test set also improved to equal the average precision (AP) performance of the training data at the highest threshold level tested (AP = 0.87, threshold = 0.4) (Figure 4.7E-F). Importantly, we found that SAGE score filtering improved malignancy prediction performance for both smartphone and dermoscopic images and all skin types, with patients at the highest FST levels (V-VI) seeing the largest improvements to model performance (Figure 4.8).

Using SAGE, we find that images of previously unseen lesion types such as squamous cell carcinoma, cutaneous T-cell lymphomas, kaposi sarcoma and metastatic carcinoma have substantially lower overall score distributions than e.g. melanoma and basal cell carcinoma, demonstrating how new disease classes can be isolated by their dissimilarity to the training data. (Figure 4.9A) Additionally, we discovered that new disease images had poor zero-shot performance when evaluated on the pre-trained malignancy prediction model, and that performance for these classes was poor across all SAGE score thresholds – accuracy for squamous cell carcinoma*,* cutaneous T-cell lymphomas*,* kaposi sarcoma and metastatic carcinoma declined by 3.6% whereas melanoma and basal cell carcinoma accuracy increased by 19.6% after thresholding at a SAGE score of 0.2. (Figure 4.9B, C) For each disease class, we also identified false negative (FN) instances where malignant lesions fell below the predictor's decision boundary (0.733) and analyzed the spread of points above and below a conservative SAGE score cutoff (0.151) where 90% of the training data is retained. This revealed an enrichment of FN images falling below the score cutoff for all malignant classes, with unseen malignancies showing a high proportional capture of FN examples (mean = 0.88). (Figure 4.9D)

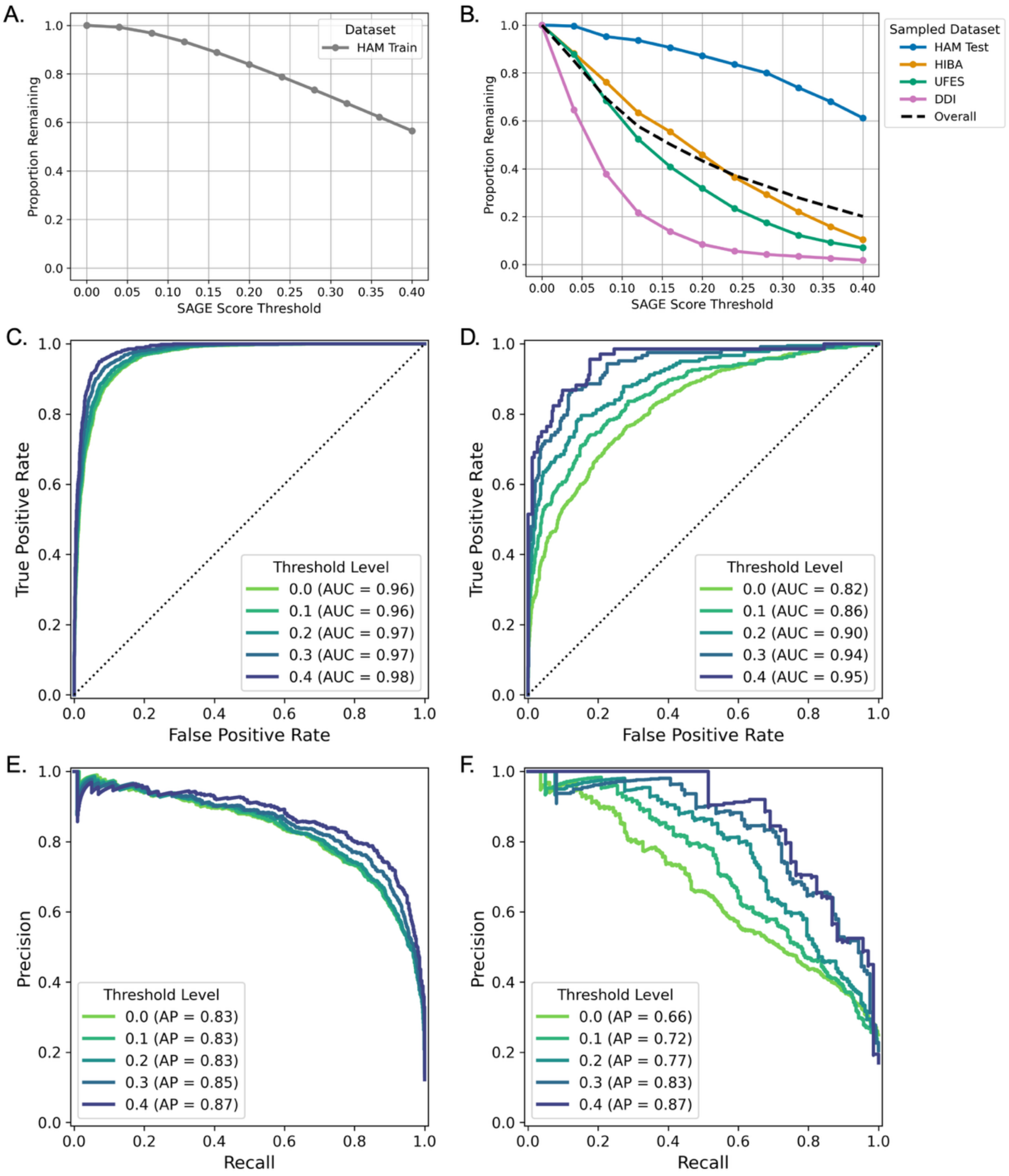


Figure 4.7 – SAGE score filtering improves performance of a pre-trained malignancy prediction model. A-B) Line plots for HAM10000 train and a balanced sample of test images (n=500 per test dataset) show retention rates as images below SAGE score thresholds are removed. C) Receiver Operating Characteristic (ROC) curve shows overall malignancy prediction performance after SAGE score thresholding at five levels for train images. D) ROC curves for sampled test dataset. E) Precision Recall (PR) curves show positive predictive performance after SAGE score thresholding for train images. F) PR curves for sampled test dataset.

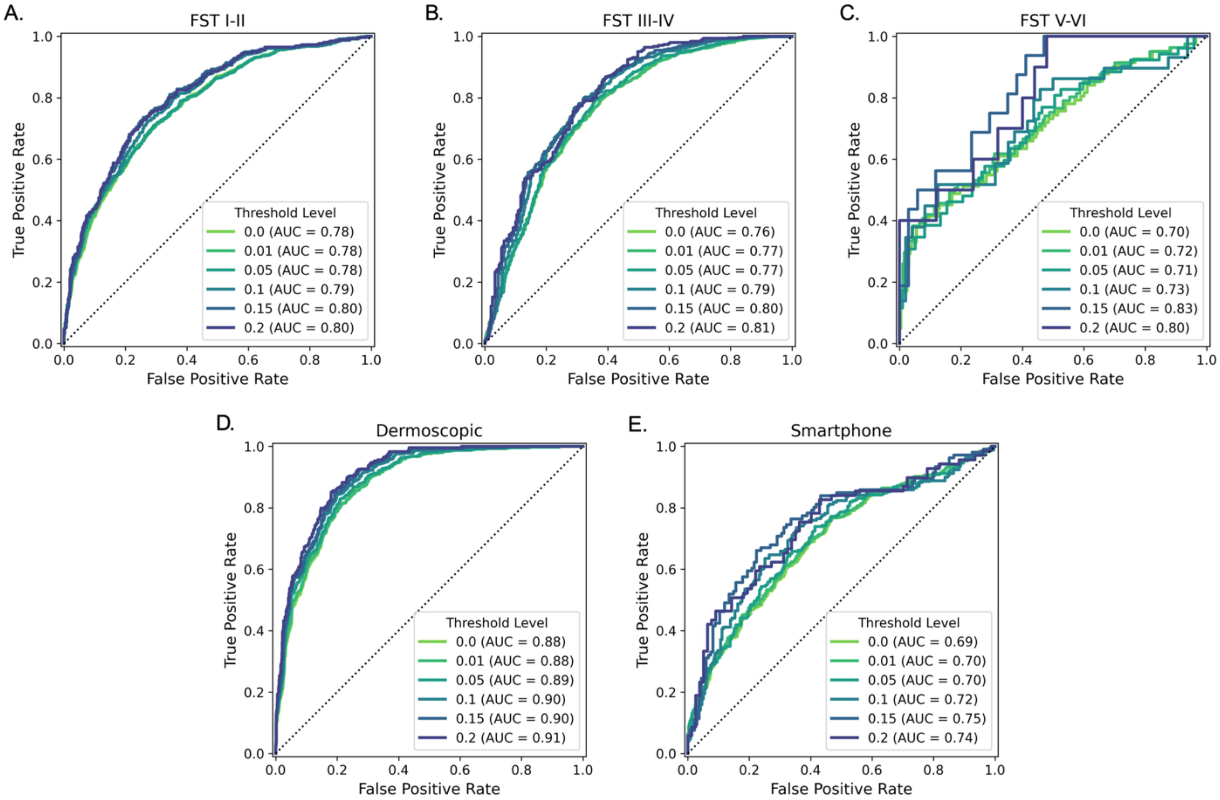


Figure 4.8 – SAGE thresholding improves performance of malignancy prediction across key metadata categories. A-C) ROC curves for three FST level groupings after filtering test images below five conservative SAGE score thresholds. D-E) ROC curves for test examples separated by imaging modality. Images below SAGE score thresholds are removed at same five threshold values.

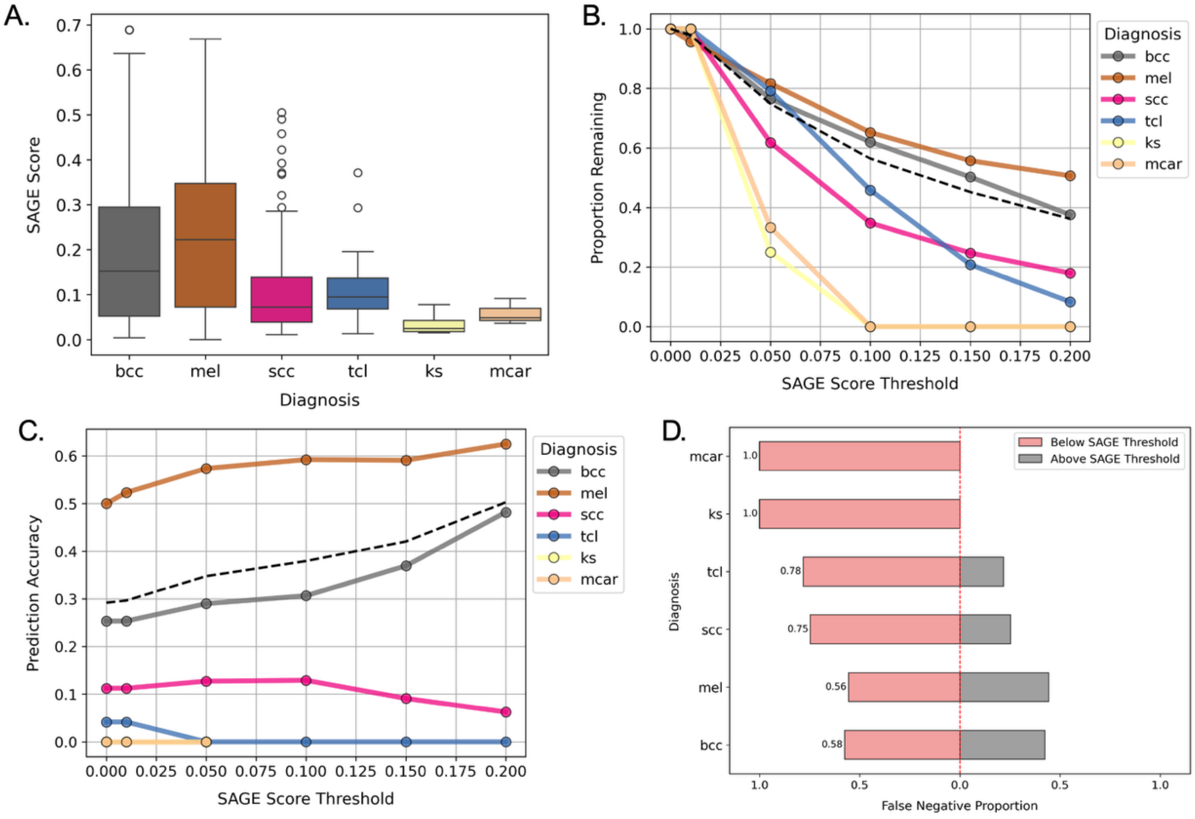


Figure 4.9 – Malignant classes missing from HAM10000 are enriched for low SAGE scores. A) SAGE score distributions are plotted as boxplots for each group. B) Line plots show the proportional loss of images as they are filtered at a series of SAGE score thresholds [0.0 - 0.2] for each malignant class. C) Line plots show changes to malignancy prediction accuracy across SAGE score thresholds. D) False negative (FN) predictions for malignancy classes show enrichment for SAGE scores below cutoff of 0.151, where 90% of train examples are retained. (bcc – basal cell carcinoma, mel – melanoma, scc – squamous cell carcinoma, tcl – T-cell lymphomas, ks – kaposi sarcoma, mcar – metastatic carcinoma)

4.4 Discussion

In this study we demonstrate the successful application of our novel uncertainty estimation method to enhance the reliability of downstream skin cancer malignancy detection. Our work uses the most popular benchmarking dataset of dermoscopic lesion images, HAM10000, as a reference to quantify out-of-distribution test images and identify detrimental image attributes without selection bias or requiring a classification label. To our knowledge, this is the first study to apply ensemble uncertainty estimation to dermatology imaging datasets in a model-agnostic manner. By incorporating four datasets from five countries in our analysis, we also address the critical need to reduce generalization gaps when applying skin cancer prediction models across global populations. Our method can pair with any dermatology ML task so long as the training data is known, which will improve integration of robust uncertainty estimation into more trustworthy automated pipelines. We envision SAGE as a powerful extension to or replacement for the concept of model cards, where users can view detailed information about a model's training data.[163] In our case, we not only quantify the differences between training data and prospective test data but also encourage users to interactively probe examples to see where and how they might differ from distributions of training image features.

However, this analysis has several limitations. Despite sourcing images from four independent datasets, the vast majority of images pertain to patients with lighter skin (FST I-IV) and lack significant representation from Asian and African populations. We also note that the FST scale itself has been shown to vary based on environmental conditions.[164,165] The data in this study consists of a relatively small cohort of ~15,000 training and test examples; inclusion of larger imaging datasets or pre-training on dermatological images could strengthen encoder fine-tuning and yield more relevant embeddings instead of initializing on default ImageNet weights from

PyTorch. We are also unable to assess the influence of identifiable markings such as scars or tattoos, as these were explicitly removed by the authors of each benchmarking dataset for patient de-identification purposes. We did not manually-annotate features for HAM10000 images due to the large dataset size and the incidence of these features in the SAGE training data was not analyzed. Furthermore, other image features that have been shown to affect ML model performance on skin imaging datasets such as color balance (e.g. blue or red pixel intensity shifts) were not annotated or assessed. Additionally, we did not study the effects of anatomical location on SAGE scoring which could provide additional guidance to users of the downstream malignancy prediction model.

While our model architecture and encoder embedding size of 32 dimensions was selected to lower overall training loss, we did not explore larger latent spaces >100 dimensions or other model architectures which might facilitate improved image reconstructions and better classification performance. As hardware support for training large ML models continues to decrease in cost, our work could expand to encompass larger foundational encoders and more complex latent space embeddings. Future work will also seek to pair SAGE with more difficult image segmentation, skin cancer diagnosis and lesion monitoring tasks across patient groups and will test a larger array of downstream models to develop an interval of task improvement facilitated by SAGE. Finally, while this work was explicitly focused on skin cancer detection using dermoscopy and clinical photography, SAGE could be broadly applicable for a wide array of clinical and non-clinical use cases.

4.5 Methods

4.5.1 Datasets

Humans Against Machine (HAM10000)[160] consists of 10,015 dermoscopy photos sourced from the Rosendahl dermatological practice in Australia and the ViDIR Group in Austria. We downloaded files from the Harvard Dataverse (https://doi.org/10.7910/DVN/DBW86T) containing JPEG images and associated metadata as a CSV file. The metadata file contains columns detailing lesion and image identifiers, patient age at time of image capture (median = 50 years old), body site of the lesion and diagnostic method. All images are of pigmented lesions and ground-truth labels are included in the metadata with six diagnostic classes: actinic keratosis (*ak*), basal cell carcinoma (*bcc*), benign keratosis (*bk*), melanocytic nevus (*nevi*), melanoma (*mel*) and vascular skin lesion (*vasc*). For the purposes of this study, only *bcc* and *mel* classes were considered malignant; all other classes were considered benign, although *ak* is known to progress to cancer in certain cases. All malignant lesions were histologically confirmed. Images have an original size of 600 x 450 pixels and were resized using bilinear interpolation where the smaller dimension of height or width was resized to 299 pixels, preserving aspect ratio. Images were cropped to 299 x 299 pixels after resizing to ensure consistency across datasets and compatibility with open-source ML architectures. Authors include multiple images of the same lesion with differing perspectives as a form of natural data augmentation and perform quality control to remove out-of-focus images and images with insufficient zoom of small lesions. We randomly split HAM10000 into training and test sets stratified by diagnostic class and removed any images of lesions from the test set that were also present in the training set, leaving 9,013 images for training and 600 for testing.

The **Hospital Italiano de Buenos Aires (HIBA)**[161] dataset contains 1,616 JPEG images of mixed dermoscopic and clinical smartphone images from Argentina which were downloaded from the ISIC archive (https://doi.org/10.34970/587329). The metadata file contains a row for each image which includes a unique image and lesion ID as well as an anonymized patient identifier. Separate metadata columns are also included for age (median = 65 years old), sex, family history of skin cancer, Fitzpatrick Skin Tone (FST) and image type differentiating between clinical smartphone and dermoscopic imaging technologies. 93% (*n*=566) of the 623 patients had skin phototype data and age and sex were recorded for over 99% of patients. Each image is assigned a ground-truth label from one of 10 diagnostic classes, 9 of which overlap with HAM10000 categories with the exception of squamous cell carcinoma (*scc*), an additional malignancy not included in the training data. The remaining image classes were basal cell carcinoma (*bcc*), melanoma (*mel*), melanocytic nevus (*nevi*), actinic keratosis (*ak*), dermatofibroma (*df*), vascular skin lesion (*vasc*), solar lentigo (*bk*), seborrheic keratosis (*bk*) and lichen planus-like keratosis (*bk*). All malignancies for *bcc, scc* and *mel* lesions were biopsy-confirmed. Images from the HIBA dataset vary in size between a maximum of 4,128 x 3,096 pixels and a minimum of 162 x 152 pixels. All files were preprocessed by converting to 8-bit RGB color channels, resized with bilinear interpolation and center-cropped to 299 x 299 pixels before use in our study.

The dermatology program at the **Universidade Federal do Espírito Santo (UFES)**[162] in Brazil published a dataset in 2020 containing 2,298 dermoscopic and clinical smartphone images with 6 diagnostic classes: basal cell carcinoma (*bcc*), squamous cell carcinoma (*scc*), actinic keratosis (*ak*), melanoma (*mel*), melanocytic nevus (*nevi*) and seborrheic keratosis (*bk*). All *bcc, scc and mel* images were considered malignant and lesions were biopsy-confirmed. We downloaded images in PNG format and the metadata CSV file from the Mendeley Data Commons link provided by the authors (https://doi.org/10.17632/zr7vgbcyr2.1). The metadata file contains patient, lesion and image identifiers as well as detailed lifestyle and living condition information such as access to piped water and sewage systems, smoking, drinking and exposure to pesticides. 100% of images have a skin type using the FST scale while only 65% of images contain patient sex. The median patient age is 62 years old with a minimum age of 6. During manual review of UFES data, we found 43 images with inconsistent or mixed labeling (e.g. duplicate images assigned to different ground-truth diagnoses: 6 instances where the ground-truth label switches between *scc* and *bcc* and 7 other instances where the label change impacts designation of the lesion as benign/malignant. All mislabeled examples were removed prior to analysis. Images from the UFES dataset have varying sizes with a maximum of 3,474 x 3,476 pixels and a minimum of 147 x 147 pixels. All files were converted to 8-bit RGB color channels, resized using bilinear interpolation and center-cropped to 299 x 299 pixels before use.

The **Diverse Dermatology Images (DDI)**[155] dataset consists of 656 images from Stanford dermatology clinics in the US captured between 2010 and 2020. All photos were taken on a clinic-issued smartphone and extracted retrospectively from electronic health records. We downloaded images and the associated metadata file from the Stanford AIMI Datasets Azure link (https://stanfordaimi.azurewebsites.net/datasets/35866158-8196-48d8-87bf-50dca81df965). The DDI dataset features over 40 unique diagnoses, including examples from all HAM10000 classes, but 36 others with low incidence in the general population and not contained in the other imaging datasets included in our analysis. (Supplementary Table 1) Metadata for DDI images is sparse with only a unique image identifier, diagnosis, malignancy and skin type information included for each lesion. Notably, sex and unique patient or lesion identifiers were absent. Images had varying sizes with a maximum of 1,914 x 1,424 pixels and a minimum of 163 x 79 pixels. We preprocessed DDI by converting images to 8-bit RGB color channels, resizing using bilinear interpolation and center-cropping to 299 x 299 pixels before use. Authors of DDI performed additional data augmentation including random rotation and vertical flipping of images before training their malignancy prediction model, however we were unable to replicate this process due to lack of access to the model's training code.

4.5.2 SAGE Model Training

We first normalized all input pixel values using mean and standard deviations from the ImageNet dataset. All image preprocessing was completed using pillow (v.10.3.0) and the torchvision (v0.17.2) transforms library. The SAGE model architecture was written in python (v3.10.14) using pytorch[166] (v2.2.2) and consists of an encoder as well as decoder and classifier modules that take the encoder's compressed embedding as input. The decoder contains 6 fully-connected layers, 6 deconvolutional layers and a learned upsampling layer while the classifier contains 5 fully-connected layers. We used a ResNet50 SAGE model encoder using the pre-trained 'IMAGENET1K_V1' weights available through the torchvision model library with an embedding size of 32 dimensions. The model was trained using the train split (90%) from the HAM1000 dataset in two stages with a balanced batch sampler and a batch size of 63. An initial warmup stage trained the encoder and classifier using the combined center loss of the latent space embedding and cross-entropy loss of classification until center loss no longer improved (maximum of 150 epochs). The second stage trained the encoder, decoder and classifier for 150 epochs and used cross-entropy loss as well as the decoder's mean-squared error (MSE) of reconstruction. We utilized a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5} with the AdamW optimizer and trained models using two Nvidia A40 GPUs.

4.5.3 t-SNE Visualization and Plotting

We generated the latent space embeddings of size 32 for images in all datasets using the trained SAGE model. For visualization purposes only, these embeddings were mapped to t-stochastic neighbor embedding (t-SNE) space using scikit-learn's manifold library, labeled according to the dataset of origin and plotted as scatterplots using matplotlib[116] (v3.9.0). To explore cluster identities, we plotted HAM10000 train and test images together and colored scatter points according to their diagnostic category. All plots were created using a combination of matplotlib and seaborn[115] (v0.13.2), and figures were assembled using BioRender.

4.5.4 SAGE Scoring

An identical scoring process was used to that described in Section 3.5.7 of this document. For this study, we calculated reference distributions of kNN distance, classifier confidence and reconstruction error using the HAM10000 dataset.

4.5.5 Manual Image Annotation and Parallel Coordinates Plot

We manually-annotated 12 independent image features (e.g. lesion contrast to skin, presence of non-skin background) that could impact quality and reliability of malignancy prediction as detailed in Table 4.1 for HIBA, UFES and DDI datasets. Images were reviewed by a single consistent observer (W.M.S.) with clinician oversight (E.B. and R.F.T.). Biasing information such as diagnosis, FST and malignancy status were removed prior to annotation. All annotations were appended to the metadata accompanying each dataset. After annotation was completed, we normalized the observed levels of 5 image features (lesion contrast, camera flash, hair level, presence of ruler and non-skin background) and plotted a line for each test image with the color denoted by the image's calculated SAGE score. We plotted images in order from low to high-scoring examples and added random jitter of 1.25% to improve visibility of line overlays.

Table 4.1 – Description of features used for manual annotation of test images.

Attribute	**Description**	**Variable**	**Level 0**	**Level 1**	**Level 2**	**Level 3**
Size	Lesion size w.r.t. cropped image frame	Ordinal	No discernable lesion	Very small to small size, nearly all skin	Majority of image is skin, lesion clearly visible	Majority of image is lesion, skin barely visible
Contrast	Lesion contrast with skin	Ordinal	No contrast	Lesion barely visible, blends with skin	Lesion clearly visible	Lesion starkly visible against skin
Hair Level	Hair density covering skin	Ordinal	No hair	Some hair visible	Denser hair (arm, chest, partial scalp)	Very dense hair (scalp)
Skew	Off-center position of lesion in image	Ordinal	Lesion centered	Some off-center skew	Lesion on periphery of image or cropped	–
Non-skin Background	Background visible in image (dermoscope border, floor, etc.)	Ordinal	No non-skin background	Any non-skin visible in image background	¼ - ½ of image consists of non-skin background	Over ½ of image is non-skin background
Cover	Cloth or bandage covering of skin (hospital gown, bandage, etc.)	Ordinal	No skin covering	Any skin covering visible	¼ - ½ of skin is covered	Over ½ of skin in image is covered
Marking	Colored marker on skin denoting lesion location	Binary	No marking	Marking present	–	–
Ruler	Measuring device in image	Binary	No ruler	Ruler present (white, black, green)	–	–
Flash	Camera flash used	Binary	No flash	Flash used	–	–
ENEM	Eyes, nose, ear or mouth visible in image	Binary	Not present	Present	–	–
Blurry	Unfocused image	Binary	Image focused	Image unfocused	–	–
Multiple	More than one lesion of similar size present	Binary	One lesion	Multiple lesions	–	–

4.5.6 Image Quality Control and FST Level Comparison

Using the manually annotated image features, we removed images from the entirety of the HIBA, DDI and UFES datasets (*n*=3,889) with low lesion pigmentation (contrast ≤ 1), camera flash, measuring devices, high hair density (hair level ≥ 2) and any non-skin background (background ≥ 1) leaving 2,168 high-quality images. We then partitioned these images into groups according to skin-tone levels (FST I-II, III-IV, V-VI) and compared the pairwise distributions of SAGE score values using a two-sided Wilcoxon rank-sum test and a significance threshold of 0.05.

4.5.7 Malignancy Prediction

We used a binary (i.e. malignant v. benign) deep-learning predictor from Daneshjou et al.[155] pre-trained on HAM10000 images and implemented in python according to instructions on the project's GitHub repository, as this was the only model from the paper with a publicly-available training dataset (https://github.com/DDI-Dataset/DDI-Code). Preprocessing steps used before malignancy prediction were the same as for training and evaluating our SAGE model. Only basal cell carcinoma and melanoma classes were considered malignant for HAM10000 images whereas HIBA and UFES also contained examples of malignant squamous cell carcinoma lesions. The DDI dataset came pre-annotated with histologically confirmed malignancies which were used as ground truth labels. All images with an output greater than or equal to the predetermined threshold of 0.733 were classified as malignant, with those falling below classified as benign lesions for all datasets.

4.5.8 Overall Malignancy Prediction Performance

After SAGE scoring and malignancy prediction, a random sample of images (*n* = 500) was taken from the HAM test, HIBA, UFES and DDI datasets and merged to form a mixed test set. Each random sample contained malignant and benign images in a 1:3

ratio mirroring that of the training data and what is encountered in clinical practice.[138] We used a series of progressive SAGE score values ([0.0, 0.1, 0.2, 0.3, 0.4]) and removed images falling below these thresholds from the HAM train and mixed test sets, plotting the proportions of each dataset remaining. At each threshold we calculated the area under the receiver operating characteristic (AUC) curve using scikit-learn[113] (v1.4.2) to visualize changes to overall performance. To measure changes to positive predictivity, we used the PrecisionRecallDisplay command to plot precision recall (PR) curves and calculate average precision (AP) values after SAGE score thresholding. For score thresholding on image type and FST levels, this process was repeated using SAGE thresholds of [0.0, 0.01, 0.05, 0.1, 0.2] and all (unsampled) test images after removal of rows with missing metadata values.

4.5.9 Prediction Performance on Test Set Malignancies

We grouped images in the sampled test dataset with a ground-truth malignant diagnosis by disease. For each diagnostic group, we repeated the process of SAGE score thresholding using a progressive set of values [0.0, 0.01, 0.05, 0.1, 0.2] and removed images falling below each level, plotting the proportion of images remaining. We plotted changes to malignancy prediction accuracy over the same thresholds, with each diagnostic group represented by a distinct line. We then plotted a joint grid of the raw malignancy prediction values against the calculated SAGE scores, isolating the false negative (FN) predictions falling below the malignancy predictor's threshold of 0.733. We used a split bar plot to show enrichment of FN examples for each test diagnostic group below a conservative SAGE score threshold (0.151) where 90% of the HAM10000 training dataset is retained.

5 Conclusion

5.1 Future Directions

5.1.1 Utilizing Predicted Radiation Courses for VA Research

The radiation course prediction project presented in Chapter 2 affords several promising directions for continued radiotherapy research within the VA network. However, there are a series of steps needed to make radiation course predictions operational for future VA research efforts. First, while electronic health record adoption within the VA was well underway at the turn of the 21st century there remains the question of how universal billing, procedure and diagnostic code usage may have changed over time. An unknown issue is whether shifting patterns of usage will have affected the reliability of predictions unequally in different periods. To estimate confidence, one can calculate the percentage of predicted 'Interior' dates falling between the start and end dates of a predicted radiation course where higher ratios indicate greater confidence that radiotherapy treatment did in fact occur during this time. The fraction of $\frac{\#\, predicted\ interior\ dates}{total\ \#\, interior\ dates}$ can be visualized by plotting on a per year basis and either a) only using predicted treatments from years with a sufficiently large ratios of interior to total dates or b) creating a cutoff value where only courses above the chosen value are retained for further analysis.

With high-confidence predictions identified, the occurrence of treatment during the predicted dates must be internally validated. This task will likely require the use of natural language processing (NLP) tools to find references to radiation in free text clinical notes that are available within the VA research computing infrastructure. As treatments are confirmed, NLP tools could also be used to capture key treatment details like site of delivery, fractionation and dosage which can be added to the existing radiation resource. Finally, the expanded treatment resource should be migrated to a queryable database environment such as SQL instead of living as an isolated document. This would improve access for VA researchers wishing to utilize our data for studies pertaining to radiation oncology. The VA offers researchers the unique advantage of accessing multimodal patient data where clinical notes and lifestyle survey data are connected with genetic data like SNP array panels collected as part of the Million Veterans Program (MVP). Using the patient IDs included in our resource to subsequently identify outcomes like secondary malignancy or immune-related adverse events from diagnostic codes or NLP searches could enable cohort-building for genome-wide association studies, the calculation of risk scores or treatment response prediction.

It should be noted that the future work involving our radiation treatment resource only pertains to patients with course data living within the VA's Corporate Data Warehouse. The majority of US Veterans are thought to receive radiation treatment from community care centers with data accessible through the Centers for Medicare and Medicaid services. Currently, the administrative data available for community care patients does not hold the same potential for integration with rich patient data like clinical text and genomic sequencing found within VHA databases. More detailed treatment data are retained by the healthcare organization that delivered radiotherapy and would therefore need to be requested and integrated into the VA before use in our shared resource. This is a non-trivial problem in aggregation of patient data from CMS that would require an automated request to community care providers for patients with high confidence predicted radiation courses asking for records in the specified date range. An ingestion mechanism would also be needed to process the result of those requests and standardize formatting between the various EHR systems for inclusion in our shared resource. The issue of limited access to CMS documents is not only a barrier to studying radiation oncology within the VA but affects all other aspects of oncology care that could benefit from additional patient data.

5.1.2 Standardizing SAGE Model Development

We have demonstrated that uncertainty quantification with SAGE is a flexible for adaptation to various downstream tasks like classification and regression, and that it can improve the quality of subsequent predictions when test samples with low generalization potential are removed. There are several steps needed beyond the proof-of-concept and application studies undertaken in Chapters 3 and 4 to standardize SAGE model development and accelerate adoption by users.

Establishing stronger OOD detection benchmarks: In Chapter 3 we showed SAGE can be used to identify OOD samples with two popular imaging datasets and an ecological dataset, however no test examples used in this chapter represent naturally occurring instances of out-of-sample data. The transforms we use are inspired by benchmarking studies like [167] which use a panel of artificial perturbations thought to represent common causes of data quality loss in ImageNet[168] examples. A stronger reflection of true OOD benchmark performance would use datasets with real-world examples of distribution shift instead of synthetic manipulations of imaging datasets. A prominent example is the WILDS[1] project which includes a package for loading various natural datasets containing domain shift (e.g. data from one source is used to train and a second source is used to test), subpopulation shift (e.g. test dataset class underrepresented in training data) and hybrid settings where both domain and subpopulation shift conditions apply. WILDS also provides additional data modalities and tasks that we have not included in prior SAGE analyses such as sentiment detection from online comment text and image segmentation from pictures of wheat plants. Illustrating how SAGE improves identification of OOD samples in a wider range of tasks, modalities and with more realistic examples of distribution shift would increase trust in our method.

Using improved encoder architectures: With a wider range of data modalities, SAGE would also need to incorporate new encoder architectures that have been fine-tuned for extracting high-quality embeddings using specific kinds of inputs. This contrasts with our previous work where we use popular convolutional architectures like ResNet[169] initialized with general purpose weights from ImageNet pretraining as our encoder of choice.

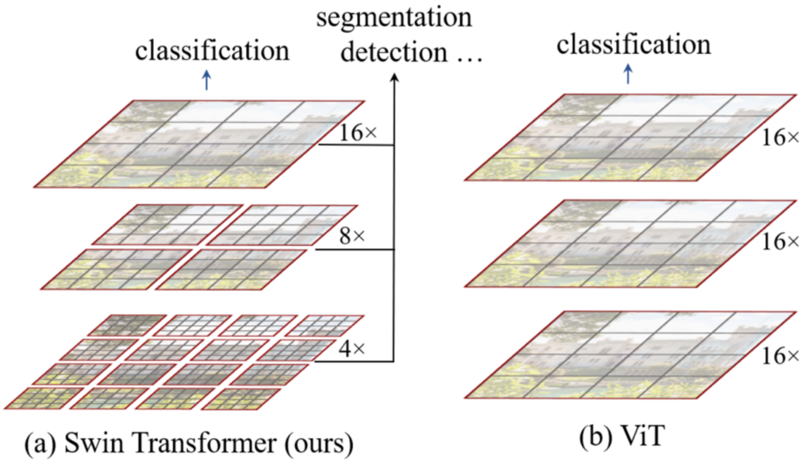


Figure 5.1 – Illustration of Swin Transformer with hierarchical stacking with different levels of resolution. In contrast, Vision Transformer resolution remains consistent. Image from [170].

While this was sufficient for detecting OOD samples, more powerful image encoders could better capture variation within the training dataset and provide stronger representations of near-distribution examples for delineation. Recent improvements to transformer-based architectures like the Swin Transformer[170] create hierarchical representations of input images that retain both local and distant relational maps between pixels, scaling linearly with image resolution. (Figure 5.1) Using Swin

Transformer encoding blocks could improve signal retention for near-distribution samples with more subtle feature variations and therefore enhance embedding quality.

Domain pretraining: Another approach for improving embedding quality is to engage in domain-specific pretraining of our encoders. Returning to the use of SAGE with dermoscopic images, a dataset of 400,000 skin lesion images was extracted from total body scans and recently released as SLICE-3D[171]. If instead of simply initializing encoders on default ImageNet weights we undertook a pretraining process with a large task-relevant dataset such as SLICE-3D, this could assist our encoder in capturing a better representation of the SAGE reference data during the secondary fine-tuning process e.g. when we train the SAGE encoder on HAM10000 dermoscopic images as was done in Chapter 4. A similar kind of transfer learning approach was recently proposed by the authors of PanDerm[172], a dermatology foundation model pretrained on over 2 million images from four modalities (dermoscopic, clinical smartphone, total body photographs, histology slides) who showed state-of-the-art performance on diverse dermatology tasks after further fine-tuning. Beyond skin lesion images, large pretrained models have been developed for NLP tasks in electronic health record tasks (Med-BERT[173]) and omics sequence data for various tasks such as predicting the effects of gene mutations on expression levels (Enformer[174]) and even broader general purpose genomics foundation models (Nucleotide Transformer[175]). Incorporating the use of these models as encoders for modality-specific applications of SAGE should greatly enrich the embedding quality and improve OOD detection.

An important caveat for inclusion of large foundational encoders is their size: the largest ResNet model we use to embed dermoscopic skin lesion images contains ~25 million trainable parameters whereas BERT models contain a minimum of 100 million (4x), Enformer contains 250 million (10x) and the smallest Nucleotide

Transformer has over 500 million parameters (20x). This poses considerable challenges to the scalability of training a SAGE model as a companion to a separate downstream task, for example in clinical use or with devices with limited processing capabilities like smartphone or tablets. Incorporating robust uncertainty estimation as part of a decision-making pipeline is needed to reduce resource burdens, and long times to inference and expensive compute could have the opposite effect. A possible alternative to our proposed schema of simultaneously training or fine-tuning the encoder with the decoder and classifier modules is to produce high-quality embeddings with a powerful foundation encoder only once, using the subsequent representations to train a decoder and classifier model. Ultimately, the value of using large pretrained encoders entails a tradeoff – foundational models could easily surpass 10 times the size of the largest encoder used in our studies whereas the smallest Swin Transformer, Swin-T, has only 28 million parameters and is comparable in size. Adapting SAGE to use a stronger modality-specific architectures like Swin-T in combination with pretraining on open source, domain-relevant datasets such as SLICE-3D could provide a boost in embedding quality that makes near-distribution outlier detection feasible while retaining the scalability benefits of our proposed design.

Integrating other methods of uncertainty estimation: Although we focus on the inclusion of three diverse measures of uncertainty in our proposed ensemble metric – distance to k-Nearest Neighbors, reconstruction error and classification confidence – we discuss other methods of uncertainty estimation in Section 1.1.3 that could be incorporated into SAGE. There is of course a practical limit to the number of uncertainty measures that could be used to generate an ensemble metric, although some ML approaches like random forest have been optimized to employ hundreds of smaller models and yield a stronger prediction when averaging model outputs together. In our case, we choose three measures of uncertainty for feasibility keeping in mind that these components can be mixed and matched to improve diversity of uncertainty estimators or to adapt to downstream task as needed.

For instance, an iteration of SAGE with fewer test samples and with higher computing resources could opt to include probabilistic measures of uncertainty like Bayesian methods instead of a standard neural network output like normalized confidence of class predictions. MC dropout[36] would use multiple forward passes from the latent embedding vector through the classifier module to approximate a Bayesian posterior. A measure such as variance of the sampled posterior could improve the measure of epistemic uncertainty in the combined SAGE score and add robustness at the cost of additional compute. A similar idea is to train an ensemble of networks using the latent embedding as input and take the average of the confidence. This approach would also enable use of alternative OOD metrics like MaxLogit that are inherently less prone to overconfidence.

Tuning SAGE with downstream information: A further consideration in the design approach we put forward with SAGE is that calculation of ensemble uncertainty scores is task *aware* but model *agnostic*. Because of this intentional separation, a gap can exist between performance of the downstream model of choice and the estimated ID vs. OOD determinations made using SAGE score thresholds, causing correctly predicted examples to be erroneously removed. One way to integrate information from the downstream model into SAGE training would be to weight examples based on accuracy or to penalize them in the latent space by incorporating a contrastive term and maximizing the distance to correct vs. incorrect pairs. If separate model predictions are available, the logits could also be used as an additional loss term (e.g. binary cross entropy loss) during SAGE model training to condition the uncertainty estimates to be higher when downstream prediction is incorrect. Another idea is distill the larger model’s knowledge into a student learner which mimics the downstream prediction process and is more computationally efficient than using the foundation model for inference.[176] (Figure 5.2)

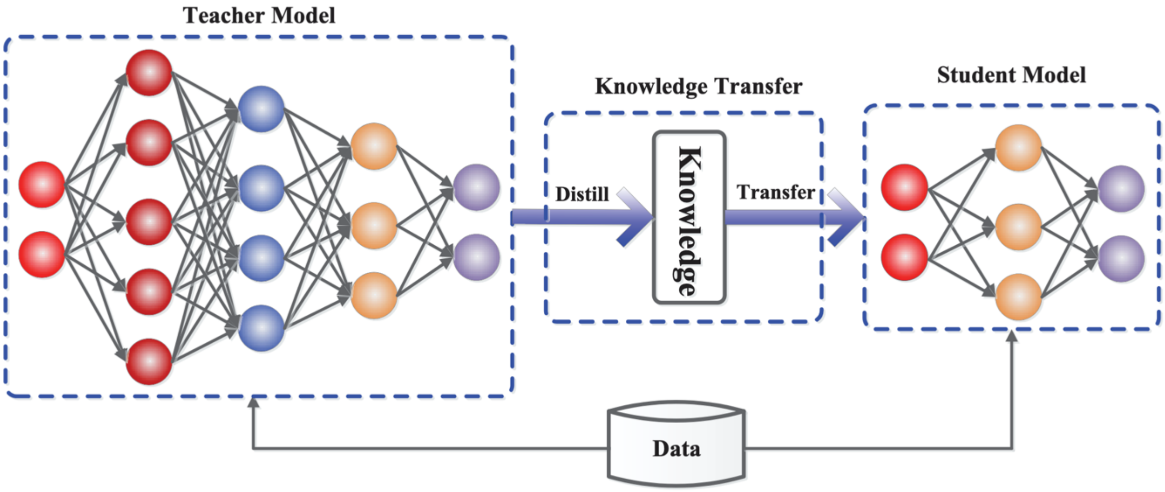


Figure 5.2 – Knowledge distillation from large "teacher" model to more computationally efficient "student" model. The student is trained to mimic the outputs of the teacher. Figure adapted from [176].

If the student model accepts embedding vectors of the training data as input, uncertainty estimates from the student model could then be incorporated into the ensemble SAGE score metric. In the case of large foundation models like PanDerm, student models could focus on distilling knowledge for one specific task (e.g. malignancy prediction from histology slide images) which could possibly yield better task and modality-aware estimates of uncertainty for the downstream model than training a SAGE module (decoder, classifier, etc.) from the general-purpose embeddings alone.

Like our omission of downstream prediction information during training, we also chose to avoid further augmentation of the training data during SAGE model development. This is a common preprocessing step in image modeling that can improve performance on both rote prediction and OOD detection tasks as demonstrated by Hendrycks et al. with AugMix[20]. It is unclear if our choice to exclude additional augmentations assists in better matching the SAGE model to the downstream task or if this limits the ability to discern near-distribution samples. Further testing of image augmentation can take place during the establishment of stronger OOD benchmarks and will determine our future recommendations to users.

Towards a SAGE Platform: The code for SAGE is available on the project's GitHub repository (https://github.com/pdxgx/sage), however adapting our python scripts to a given use case requires a low level understanding of deep learning packages like PyTorch which may be a barrier to those with minimal coding experience. Furthermore, even using SAGE for the purpose-built application of dermoscopic dataset comparison requires running our scripts via command line interfaces (CLI) like the Mac Terminal application or as part of an integrated development environment. To improve accessibility to our approach and standardize implementation of SAGE, we could create a python package which serves as a wrapper to more code-heavy implementations of PyTorch and torchvision. This package would provide users with the option to select from several preset encoders like vision transformers and CNNs of varying sizes (e.g. ResNet18, 34, 50, 100), a manual input of latent dimension sizes, and pair them with several choices for the other SAGE components like decoders and classifiers with the inclusion of Bayesian approaches like MC dropout. Additionally, we could expand our offerings of the approaches used for combining uncertainty measures beyond the conservative geometric mean method that we implemented in Chapters 3 and 4 like arithmetic mean or coefficients for a weighted average. This package would be offered for use in python development environments and integrated into the backend of a web platform where users can upload their datasets and choose SAGE model settings without needing to write any code themselves. A suite of pretrained, domain-specific encoders would be hosted on a cloud storage system and connected to our web platform via an API, with the ability to mix and match encoders with the SAGE components chosen by users. Finally, custom development could be expedited by connecting our web platform to a high-performance computing cluster where SAGE models are efficiently trained, used to generate data embeddings and output an uncertainty score for both the uploaded training and test data with the remote use of hardware such as GPUs. The outputs of the trained model could then be retained as part of the user's SAGE project for faster inference when new test data are uploaded to the platform for evaluation.

5.2 Ethics of ML Generalization in Oncology Informatics

As a response to George E.P. Box's aphorism, "All models are wrong, but some are useful", the need for ML advancement can simplistically be formulated as the desire to improve the utility of statistical models. With success too often determined by metrics like Top-1 or Top-5 accuracy on benchmarking datasets, it is common to forego wider discussion about the choices made by developers during model development and the implications of such tradeoffs when models are later deployed in settings that affect real people. In the field of oncology informatics, we must not only acknowledge performance-based definitions of what makes ML "useful" but consider the ethics of model imperfections and their consequences for human health. Despite the large literature dedicated to improving generalization reviewed in Section 1.1.3, the logical result of George Box's truism is that *all* models will retain some form of deficit in practice. Indeed, an over-emphasis on maintaining model performance in every possible deployment situation has even been described as chasing the "myth" of generalizability by Futoma et al.[177]: a more principled approach would allocate greater effort to learning how to identify and cope with these inevitable deficiencies instead of obsessing over their eradication.

Considering the ethical questions embodied in such tradeoffs is still in a stage of infancy. Prominent issues like underspecification[178], where a population sample is missing from a model's training dataset and therefore underperforms when testing on people from that population, are clearly problematic in oncology settings and have therefore been discussed in some detail. A 2024 paper by Vandersluis and Savulescu[179] considers the problem of underspecification in the context of a breast cancer diagnosis model trained only on females which therefore underperforms on male patients – should this model be deployed in clinical settings despite its known weaknesses? They conclude that the benefit to the majority of patients (females) outweighs the exclusion of a small minority group while commenting that the simple exclusionary criteria offered by this example (e.g. if male, remove from evaluation) are not available when more complicated subpopulations of patient cohorts are silently underspecified. Furthermore, they note that the minority group in this instance doesn't bear a history of injustice and medical treatment bias that could change the moral calculations of such a dilemma.

Yet opaque underspecifications exist even in settings where minority patient populations are clearly visible *and* bear such a history of injustice. A 2021 review paper[180] highlighted these concerns in automated dermatology diagnosis algorithms where only 10% (*n* = 7) of studies included any information on skin type despite the fact that patients with darker skin typically have lower disease detection rates and higher mortality in cases of malignant melanoma[181]. Implementing AI systems for diagnosis could therefore exacerbate existing health disparities if the same systems are used across patients of all skin types.

Dismayingly, we have seen the propagation of such biases in high-notoriety projects like Google Health's DermAssist tool[182], where only 0.28% of training cases included patients with the darkest Fitzpatrick Skin Type (FST), and DermaSensor, the only currently FDA-approved dermatology AI diagnostic tool where 97.7% of patients in its DERM-ASSESSIII validation trial[183] identified as white and over 84% of participants were classified as FST level III or lower. DermaSensor's conditional FDA approval in 2024 came with a requirement to monitor performance of their device in underrepresented patient populations. The clinical follow-up study timeline remains unknown. While the FDA's approval requirements are commendable, we can't predict the extent to which such requirements will result in the revocation of approval if they are shown to be biased against patients with darker skin types. The presence of skin type disparities in DermAssist and DermaSensor training data underscores the urgent need for quantification methods that can help users determine when a patient sample is inappropriate for AI-assisted diagnosis and refrain from making recommendations. Groh et al.[184] demonstrate that even the use of "fair" decision support algorithms can exacerbate human diagnostic accuracy disparities for non-specialists when reviewing images from light and dark skin patients. Furthermore, they show that humans are sometimes influenced by automated diagnostic tools to include incorrect predictions in their differential diagnoses. We therefore posit that a robust uncertainty estimation tool like SAGE (which we show does not suffer OOD detection loss as FST levels vary) must accompany diagnostic support tools as indicators of model failure and algorithmic bias *before* testing.

Ultimately, the gains from developing methods for stronger model generalization and efforts to preempt generalization failures are not zero-sum: both are needed to prevent the ethical failures of algorithmic bias propagation and silent underspecifications that can diminish the prospect of using ML to achieve better outcomes in cancer care settings.

5.3 Summary

Enabling machine learning models to generalize safely to diverse samples beyond the original training distributions is an exciting prospect for expanding the reach of AI systems. Undoubtedly of equal (if not superior) importance is our need to develop a keen understanding of where, when and why these systems fail. The significance of such failures is epitomized in medical disciplines like oncology where lapses in performance are poised to affect quality of care, research efforts and human health outcomes.

To summarize, the contributions presented in the body of this thesis are:

1. A structured data extraction method for attributing radiology course dates using machine learning that takes universal billing, procedure and diagnostic codes as input. The method was retrospectively applied to build a resource of over 1 million predicted radiation courses linked to US Veteran patients.
2. A robust, task-aware and model agnostic method (SAGE) for dataset comparison based on uncertainty estimation. We use SAGE to detect and remove corrupted samples in two benchmark imaging datasets and one ecological dataset.
3. Application of SAGE to real world skin lesion imaging datasets enables the identification of artifacts and outliers and improves the generalization of a downstream malignancy prediction model.

Our goal with this work is to advance the adoption of techniques for safe and sustainable ML in two oncology settings, namely identifying radiotherapy treatments from widely generalizable administrative data and creating a method for standardizing uncertainty estimation and OOD detection to preempt generalization failures when using automated diagnostic models. Although limited to these two applications, the principles we demonstrate are crucial to advancing model safety in medical applications writ large by identifying training features that transfer between medical centers and creating stronger measures of OOD detection that are valid even under conditions of data drift. Furthermore, we seek to promote uncertainty estimation as a key steppingstone to risk-aware machine learning applications not just in healthcare but in virtually any setting where a human operator needs to guard against avoidable errors in ML decision-making.

Bibliography

[1] Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain. Psychol Rev 1958;65:386–408. https://doi.org/10.1037/h0042519.

[2] Tech-AI-Math. The Perceptron: A Foundational Building Block of Neural Networks. Medium 2023. https://ai.plainenglish.io/the-perceptron-a-foundational-building-block-of-neural-networks-6c4bdd29a456 (accessed August 1, 2025).

[3] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw 1989;2:359–66. https://doi.org/10.1016/0893-6080(89)90020-8.

[4] Rumelhart DE, McClelland JL. Learning Internal Representations by Error Propagation. Parallel Distrib. Process. Explor. Microstruct. Cogn. Found., MIT Press; 1987, p. 318–62.

[5] Fradkov AL. Early History of Machine Learning. IFAC-Pap 2020;53:1385–90. https://doi.org/10.1016/j.ifacol.2020.12.1888.

[6] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–44. https://doi.org/10.1038/nature14539.

[7] Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. Adv. Neural Inf. Process. Syst., vol. 25, Curran Associates, Inc.; 2012.

[8] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions, IEEE Computer Society; 2015, p. 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

[9] Techniques for Interpretable Machine Learning – Communications of the ACM 2020. https://cacm.acm.org/research/techniques-for-interpretable-machine-learning/ (accessed August 1, 2025).

[10] Breiman L. Random Forests. Mach Learn 2001;45:5–32. https://doi.org/10.1023/A:1010933404324.

[11] Freund Y, Schapire RE. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. J Comput Syst Sci 1997;55:119–39. https://doi.org/10.1006/jcss.1997.1504.

[12] Weng L. From Autoencoder to Beta-VAE 2018. https://lilianweng.github.io/posts/2018-08-12-vae/ (accessed August 3, 2025).

[13] Bourlard H, Kamp Y. Auto-association by multilayer perceptrons and singular value decomposition. Biol Cybern 1988;59:291–4. https://doi.org/10.1007/BF00332918.

[14] Hinton GE, Zemel RS. Autoencoders, minimum description length and Helmholtz free energy. Proc. 7th Int. Conf. Neural Inf. Process. Syst., San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1993, p. 3–10.

[15] Vincent P, Larochelle H, Bengio Y, Manzagol P-A. Extracting and composing robust features with denoising autoencoders. Proc. 25th Int. Conf. Mach. Learn., New York, NY, USA: Association for Computing Machinery; 2008, p. 1096–103. https://doi.org/10.1145/1390156.1390294.

[16] Kingma DP, Welling M. Auto-Encoding Variational Bayes 2022. https://doi.org/10.48550/arXiv.1312.6114.

[17] Le L, Patterson A, White M. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. Adv. Neural Inf. Process. Syst., vol. 31, Curran Associates, Inc.; 2018.

[18] Ben-David S, Blitzer J, Crammer K, Kulesza A, Pereira F, Vaughan JW. A theory of learning from different domains. Mach Learn 2010;79:151–75. https://doi.org/10.1007/s10994-009-5152-4.

[19] Quionero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND. Dataset Shift in Machine Learning. The MIT Press; 2009.

[20] Hendrycks D, Mu N, Cubuk ED, Zoph B, Gilmer J, Lakshminarayanan B. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty 2020. https://doi.org/10.48550/arXiv.1912.02781.

[21] Hendrycks D, Lee K, Mazeika M. Using Pre-Training Can Improve Model Robustness and Uncertainty 2019. https://doi.org/10.48550/arXiv.1901.09960.

[22] Goetz L, Seedat N, Vandersluis R, van der Schaar M. Generalization—a key challenge for responsible AI in patient-facing clinical applications. Npj Digit Med 2024;7:126. https://doi.org/10.1038/s41746-024-01127-3.

[23] Krogh A, Hertz J. A Simple Weight Decay Can Improve Generalization. Adv. Neural Inf. Process. Syst., vol. 4, Morgan-Kaufmann; 1991.

[24] Tibshirani R. Regression Shrinkage and Selection Via the Lasso. J R Stat Soc Ser B Methodol 1996;58:267–88. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[25] Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. Technometrics 2000;42:80–6. https://doi.org/10.2307/1271436.

[26] Loshchilov I, Hutter F. Decoupled Weight Decay Regularization 2019. https://doi.org/10.48550/arXiv.1711.05101.

[27] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. J Mach Learn Res 2014;15:1929–58.

[28] SJÖBERG J, LJUNG L. Overtraining, regularization and searching for a minimum, with application to neural networks. Int J Control 1995;62:1391–407. https://doi.org/10.1080/00207179508921605.

[29] Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent. In: Lechevallier Y, Saporta G, editors. Proc. COMPSTAT2010, Heidelberg: Physica-Verlag HD; 2010, p. 177–86. https://doi.org/10.1007/978-3-7908-2604-3_16.

[30] Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning requires rethinking generalization 2017. https://doi.org/10.48550/arXiv.1611.03530.

[31] Hendrycks D, Gimpel K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks 2018. https://doi.org/10.48550/arXiv.1610.02136.

[32] Guo C, Pleiss G, Sun Y, Weinberger KQ. On Calibration of Modern Neural Networks. Proc. 34th Int. Conf. Mach. Learn., PMLR; 2017, p. 1321–30.

[33] Müller R, Kornblith S, Hinton G. When Does Label Smoothing Help? 2020. https://doi.org/10.48550/arXiv.1906.02629.

[34] Hendrycks D, Basart S, Mazeika M, Zou A, Kwon J, Mostajabi M, et al. Scaling Out-of-Distribution Detection for Real-World Settings 2022. https://doi.org/10.48550/arXiv.1911.11132.

[35] Liang S, Li Y, Srikant R. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks 2020. https://doi.org/10.48550/arXiv.1706.02690.

[36] Gal Y, Ghahramani Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. Proc. 33rd Int. Conf. Mach. Learn., PMLR; 2016, p. 1050–9.

[37] Lakshminarayanan B, Pritzel A, Blundell C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles n.d.

[38] Sun Y, Ming Y, Zhu X, Li Y. Out-of-Distribution Detection with Deep Nearest Neighbors n.d.

[39] Venkataramanan A, Benbihi A, Laviale M, Pradalier C. Gaussian Latent Representations for Uncertainty Estimation using Mahalanobis Distance in Deep Classifiers. 2023 IEEECVF Int. Conf. Comput. Vis. Workshop ICCVW, Paris, France: IEEE; 2023, p. 4490–9. https://doi.org/10.1109/ICCVW60793.2023.00483.

[40] Lee K, Lee K, Lee H, Shin J. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks 2018. https://doi.org/10.48550/arXiv.1807.03888.

[41] Rabanser S, Günnemann S, Lipton ZC. Failing Loudly: An Empirical Study of Methods for Detecting Dataset Shift 2019. https://doi.org/10.48550/arXiv.1810.11953.

[42] Hesse BW, Ahern D, Beckjord E. Oncology Informatics: Using Health Information Technology to Improve Processes and Outcomes in Cancer. Academic Press; 2016.

[43] Cook JA, Collins GS. The rise of big clinical databases. Br J Surg 2015;102:e93–101. https://doi.org/10.1002/bjs.9723.

[44] Weinstein JN, Collisson EA, Mills GB, Shaw KRM, Ozenberger BA, Ellrott K, et al. The Cancer Genome Atlas Pan-Cancer analysis project. Nat Genet 2013;45:1113–20. https://doi.org/10.1038/ng.2764.

[45] Clark K, Vendt B, Smith K, Freymann J, Kirby J, Koppel P, et al. The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository. J Digit Imaging 2013;26:1045–57. https://doi.org/10.1007/s10278-013-9622-7.

[46] Bycroft C, Freeman C, Petkova D, Band G, Elliott LT, Sharp K, et al. The UK Biobank resource with deep phenotyping and genomic data. Nature 2018;562:203–9. https://doi.org/10.1038/s41586-018-0579-z.

[47] Schwartz WB, Patil RS, Szolovits P. Artificial Intelligence in Medicine. N Engl J Med 1987;316:685–8. https://doi.org/10.1056/NEJM198703123161109.

[48] Kann BH, Hosny A, Aerts HJ. Artificial Intelligence for Clinical Oncology. Cancer Cell 2021;39:916–27. https://doi.org/10.1016/j.ccell.2021.04.002.

[49] Mavaddat N, Michailidou K, Dennis J, Lush M, Fachal L, Lee A, et al. Polygenic Risk Scores for Prediction of Breast Cancer and Breast Cancer Subtypes. Am J Hum Genet 2019;104:21–34. https://doi.org/10.1016/j.ajhg.2018.11.002.

[50] Seibert TM, Fan CC, Wang Y, Zuber V, Karunamuni R, Parsons JK, et al. Polygenic hazard score to guide screening for aggressive prostate cancer: development and validation in large scale cohorts. BMJ 2018;360:j5757. https://doi.org/10.1136/bmj.j5757.

[51] Seletkov D, Starck S, Mueller TT, Zhang Y, Steinhelfer L, Rueckert D, et al. AI-driven preclinical disease risk assessment using imaging in UK biobank. Npj Digit Med 2025;8:480. https://doi.org/10.1038/s41746-025-01771-3.

[52] Shen L, Margolies LR, Rothstein JH, Fluder E, McBride R, Sieh W. Deep Learning to Improve Breast Cancer Detection on Screening Mammography. Sci Rep 2019;9:12495. https://doi.org/10.1038/s41598-019-48995-4.

[53] Urban G, Tripathi P, Alkayali T, Mittal M, Jalali F, Karnes W, et al. Deep Learning Localizes and Identifies Polyps in Real Time With 96% Accuracy in Screening Colonoscopy. Gastroenterology 2018;155:1069-1078.e8. https://doi.org/10.1053/j.gastro.2018.06.037.

[54] Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature 2017;542:115–8. https://doi.org/10.1038/nature21056.

[55] Vermeulen C, Pagès-Gallego M, Kester L, Kranendonk MEG, Wesseling P, Verburg N, et al. Ultra-fast deep-learned CNS tumour classification during surgery. Nature 2023;622:842–9. https://doi.org/10.1038/s41586-023-06615-2.

[56] Parker JS, Mullins M, Cheang MCU, Leung S, Voduc D, Vickery T, et al. Supervised Risk Predictor of Breast Cancer Based on Intrinsic Subtypes. J Clin Oncol 2009;27:1160–7. https://doi.org/10.1200/JCO.2008.18.1370.

[57] Ellrott K, Wong CK, Yau C, Castro MAA, Lee JA, Karlberg BJ, et al. Classification of non-TCGA cancer samples to TCGA molecular subtypes using compact feature sets. Cancer Cell 2025;43:195-212.e11. https://doi.org/10.1016/j.ccell.2024.12.002.

[58] Sammut S-J, Crispin-Ortuzar M, Chin S-F, Provenzano E, Bardwell HA, Ma W, et al. Multi-omic machine learning predictor of breast cancer therapy response. Nature 2022;601:623–9. https://doi.org/10.1038/s41586-021-04278-5.

[59] Peterson DJ, Ostberg NP, Blayney DW, Brooks JD, Hernandez-Boussard T. Machine Learning Applied to Electronic Health Records: Identification of Chemotherapy Patients at High Risk for Preventable Emergency Department Visits and Hospital Admissions. JCO Clin Cancer Inform 2021:1106–26. https://doi.org/10.1200/CCI.21.00116.

[60] Heilbroner SP, Few R, Neilan TG, Mueller J, Chalwa J, Charest F, et al. Predicting cardiac adverse events in patients receiving immune checkpoint inhibitors: a machine learning approach. J Immunother Cancer 2021;9:e002545. https://doi.org/10.1136/jitc-2021-002545.

[61] Zhang X, Zhang Y, Zhang Q, Ren Y, Qiu T, Ma J, et al. Extracting comprehensive clinical information for breast cancer using deep learning methods. Int J Med Inf 2019;132:103985. https://doi.org/10.1016/j.ijmedinf.2019.103985.

[62] Forsyth AW, Barzilay R, Hughes KS, Lui D, Lorenz KA, Enzinger A, et al. Machine Learning Methods to Extract Documentation of Breast Cancer Symptoms From Electronic Health Records. J Pain Symptom Manage 2018;55:1492–9. https://doi.org/10.1016/j.jpainsymman.2018.02.016.

[63] Singh R, Bapna M, Diab AR, Ruiz ES, Lotter W. How AI is used in FDA-authorized medical devices: a taxonomy across 1,016 authorizations. Npj Digit Med 2025;8:388. https://doi.org/10.1038/s41746-025-01800-1.

[64] Cassidy B, Kendrick C, Brodzicki A, Jaworek-Korjakowska J, Yap MH. Analysis of the ISIC image datasets: Usage, benchmarks and recommendations. Med Image Anal 2022;75:102305. https://doi.org/10.1016/j.media.2021.102305.

[65] Yang J, Shi R, Wei D, Liu Z, Zhao L, Ke B, et al. MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. Sci Data 2023;10:41. https://doi.org/10.1038/s41597-022-01721-8.

[66] Hussain S, Naseem U, Ali M, Avendaño Avalos DB, Cardona-Huerta S, Bosques Palomo BA, et al. TECRR: a benchmark dataset of radiological reports for BI-RADS classification with machine learning, deep learning, and large language

model baselines. BMC Med Inform Decis Mak 2024;24:310. https://doi.org/10.1186/s12911-024-02717-7.

[67] Garrucho L, Kushibar K, Reidel C-A, Joshi S, Osuala R, Tsirikoglou A, et al. A large-scale multicenter breast cancer DCE-MRI benchmark dataset with expert segmentations. Sci Data 2025;12:453. https://doi.org/10.1038/s41597-025-04707-4.

[68] Doerrich S, Di Salvo F, Brockmann J, Ledig C. Rethinking model prototyping through the MedMNIST+ dataset collection. Sci Rep 2025;15:7669. https://doi.org/10.1038/s41598-025-92156-9.

[69] Varoquaux G, Cheplygina V. Machine learning for medical imaging: methodological failures and recommendations for the future. Npj Digit Med 2022;5:48. https://doi.org/10.1038/s41746-022-00592-y.

[70] Hendrycks D, Basart S, Mu N, Kadavath S, Wang F, Dorundo E, et al. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization 2021. https://doi.org/10.48550/arXiv.2006.16241.

[71] Schömig-Markiefka B, Pryalukhin A, Hulla W, Bychkov A, Fukuoka J, Madabhushi A, et al. Quality control stress test for deep learning-based diagnostic model in digital pathology. Mod Pathol 2021;34:2098–108. https://doi.org/10.1038/s41379-021-00859-x.

[72] Petrie TC, Larson C, Heath M, Samatham R, Davis A, Berry EG, et al. Quantifying Acceptable Artefact Ranges for Dermatologic Classification Algorithms. Skin Health Dis 2021;1:ski2.19. https://doi.org/10.1002/ski2.19.

[73] Chlap P, Min H, Vandenberg N, Dowling J, Holloway L, Haworth A. A review of medical image data augmentation techniques for deep learning applications. J Med Imaging Radiat Oncol 2021;65:545–63. https://doi.org/10.1111/1754-9485.13261.

[74] Wu M, Wang S, Pan S, Terentis AC, Strasswimmer J, Zhu X. Deep learning data augmentation for Raman spectroscopy cancer tissue classification. Sci Rep 2021;11:23842. https://doi.org/10.1038/s41598-021-02687-0.

[75] Ovadia Y, Fertig E, Ren J, Nado Z, Sculley D, Nowozin S, et al. Can you trust your model' s uncertainty? Evaluating predictive uncertainty under dataset shift. Adv. Neural Inf. Process. Syst., vol. 32, Curran Associates, Inc.; 2019.

[76] Linmans J, Elfwing S, van der Laak J, Litjens G. Predictive uncertainty estimation for out-of-distribution detection in digital pathology. Med Image Anal 2023;83:102655. https://doi.org/10.1016/j.media.2022.102655.

[77] Guha Roy A, Ren J, Azizi S, Loh A, Natarajan V, Mustafa B, et al. Does your dermatology classifier know what it doesn't know? Detecting the long-tail of unseen conditions. Med Image Anal 2022;75:102274. https://doi.org/10.1016/j.media.2021.102274.

[78] Health C for D and R. Methods and Tools for Effective Postmarket Monitoring of Artificial Intelligence (AI)-Enabled Medical Devices. FDA 2024.

[79] Windecker D, Baj G, Shiri I, Kazaj PM, Kaesmacher J, Gräni C, et al. Generalizability of FDA-Approved AI-Enabled Medical Devices for Clinical Use. JAMA Netw Open 2025;8:e258052. https://doi.org/10.1001/jamanetworkopen.2025.8052.

[80] Feng J, Xia F, Singh K, Pirracchio R. Not All Clinical AI Monitoring Systems Are Created Equal: Review and Recommendations. NEJM AI 2025;2:AIra2400657. https://doi.org/10.1056/AIra2400657.

[81] Feng J, Phillips RV, Malenica I, Bishara A, Hubbard AE, Celi LA, et al. Clinical artificial intelligence quality improvement: towards continual monitoring and updating of AI algorithms in healthcare. Npj Digit Med 2022;5:66. https://doi.org/10.1038/s41746-022-00611-y.

[82] Orth M, Lauber K, Niyazi M, Friedl AA, Li M, Maihöfer C, et al. Current concepts in clinical radiation oncology. Radiat Environ Biophys 2014;53:1–29. https://doi.org/10.1007/s00411-013-0497-2.

[83] Wilkins A, Parker C. Treating prostate cancer with radiotherapy. Nat Rev Clin Oncol 2010;7:583–9. https://doi.org/10.1038/nrclinonc.2010.135.

[84] Wolff RF, Ryder S, Bossi A, Briganti A, Crook J, Henry A, et al. A systematic review of randomised controlled trials of radiotherapy for localised prostate cancer. Eur J Cancer 2015;51:2345–67. https://doi.org/10.1016/j.ejca.2015.07.019.

[85] Effects of radiotherapy and of differences in the extent of surgery for early breast cancer on local recurrence and 15-year survival: an overview of the randomised trials. The Lancet 2005;366:2087–106. https://doi.org/10.1016/S0140-6736(05)67887-7.

[86] Vinod SK, Hau E. Radiotherapy treatment for lung cancer: Current status and future directions. Respirology 2020;25:61–71. https://doi.org/10.1111/resp.13870.

[87] Sonke J-J, Belderbos J. Adaptive Radiotherapy for Lung Cancer. Semin Radiat Oncol 2010;20:94–106. https://doi.org/10.1016/j.semradonc.2009.11.003.

[88] Locke J, Karimpour S, Young G, Lockett MA, Perez CA. Radiotherapy for epithelial skin cancer. Int J Radiat Oncol 2001;51:748–55. https://doi.org/10.1016/S0360-3016(01)01656-X.

[89] Amos EH, Mendenhall WM, McCarty PJ, Gage JO, Emlet JL, Lowrey GC, et al. Postoperative radiotherapy for locally advanced colon cancer. Ann Surg Oncol 1996;3:431–6. https://doi.org/10.1007/BF02305760.

[90] Bartelink H, Roelofsen F, Eschwege F, Rougier P, Bosset JF, Gonzalez DG, et al. Concomitant radiotherapy and chemotherapy is superior to radiotherapy alone in the treatment of locally advanced anal cancer: results of a phase III randomized

trial of the European Organization for Research and Treatment of Cancer Radiotherapy and Gastrointestinal Cooperative Groups. J Clin Oncol 1997;15:2040–9. https://doi.org/10.1200/JCO.1997.15.5.2040.

[91] van Dijk TH, Tamas K, Beukema JC, Beets GL, Gelderblom AJ, de Jong KP, et al. Evaluation of short-course radiotherapy followed by neoadjuvant bevacizumab, capecitabine, and oxaliplatin and subsequent radical surgical treatment in primary stage IV rectal cancer†. Ann Oncol 2013;24:1762–9. https://doi.org/10.1093/annonc/mdt124.

[92] Twyman-Saint Victor C, Rech AJ, Maity A, Rengan R, Pauken KE, Stelekati E, et al. Radiation and dual checkpoint blockade activate non-redundant immune mechanisms in cancer. Nature 2015;520:373–7. https://doi.org/10.1038/nature14292.

[93] James ND, Hussain SA, Hall E, Jenkins P, Tremlett J, Rawlings C, et al. Radiotherapy with or without Chemotherapy in Muscle-Invasive Bladder Cancer. N Engl J Med 2012;366:1477–88. https://doi.org/10.1056/NEJMoa1106106.

[94] Tsao MN, Wara WM, Larson DA. Radiation therapy for benign central nervous system disease. Semin Radiat Oncol 1999;9:120–33. https://doi.org/10.1016/S1053-4296(99)80002-2.

[95] research.VA.gov | Cancer n.d. https://www.research.va.gov/topics/cancer.cfm (accessed October 5, 2023).

[96] VA.gov | Veterans Affairs n.d. https://www.va.gov/health/aboutvha.asp (accessed October 5, 2023).

[97] Xiao C, Choi E, Sun J. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. J Am Med Inform Assoc 2018;25:1419–28. https://doi.org/10.1093/jamia/ocy068.

[98] Jamian L, Wheless L, Crofford LJ, Barnado A. Rule-based and machine learning algorithms identify patients with systemic sclerosis accurately in the electronic health record. Arthritis Res Ther 2019;21:305. https://doi.org/10.1186/s13075-019-2092-7.

[99] Bronsert M, Singh AB, Henderson WG, Hammermeister K, Meguid RA, Colborn KL. Identification of postoperative complications using electronic health record data and machine learning. Am J Surg 2020;220:114–9. https://doi.org/10.1016/j.amjsurg.2019.10.009.

[100] Cher BAY, Dykstra M, Wang C, Schipper M, Hayman JA, Mayo CS, et al. Trends in Radiation Oncology Treatment Fractionation at a Single Academic Center, 2010 to 2020. Adv Radiat Oncol 2022;7:101032. https://doi.org/10.1016/j.adro.2022.101032.

[101] Schroeder S, Wilhelm H, Wallace C, Rodgers K, Lin MH, Pompos A, et al. Contemporary Fractionation Trends within a Large United States Academic

Radiation Oncology Department. Int J Radiat Oncol Biol Phys 2018;102:e428–9. https://doi.org/10.1016/j.ijrobp.2018.07.1249.

[102] Goldberg SI, Niemierko A, Turchin A. Analysis of Data Errors in Clinical Research Databases. AMIA Annu Symp Proc 2008;2008:242–6.

[103] Bourhis J, Overgaard J, Audry H, Ang KK, Saunders M, Bernier J, et al. Hyperfractionated or accelerated radiotherapy in head and neck cancer: a meta-analysis. The Lancet 2006;368:843–54. https://doi.org/10.1016/S0140-6736(06)69121-6.

[104] Mirnezami R, Chang GJ, Das P, Chandrakumaran K, Tekkis P, Darzi A, et al. Intraoperative radiotherapy in colorectal cancer: Systematic review and meta-analysis of techniques, long-term outcomes, and complications. Surg Oncol 2013;22:22–35. https://doi.org/10.1016/j.suronc.2012.11.001.

[105] Barry M, Kell MR. Radiotherapy and breast reconstruction: a meta-analysis. Breast Cancer Res Treat 2011;127:15–22. https://doi.org/10.1007/s10549-011-1401-x.

[106] Luo Y, Chen S, Valdes G. Machine learning for radiation outcome modeling and prediction. Med Phys 2020;47:e178–84. https://doi.org/10.1002/mp.13570.

[107] Appelt AL, Elhaminia B, Gooya A, Gilbert A, Nix M. Deep Learning for Radiotherapy Outcome Prediction Using Dose Data – A Review. Clin Oncol 2022;34:e87–96. https://doi.org/10.1016/j.clon.2021.12.002.

[108] Isaksson LJ, Pepa M, Zaffaroni M, Marvaso G, Alterio D, Volpe S, et al. Machine Learning-Based Models for Prediction of Toxicity Outcomes in Radiotherapy. Front Oncol 2020;10.

[109] Wallis CJD, Mahar AL, Choo R, Herschorn S, Kodama RT, Shah PS, et al. Second malignancies after radiotherapy for prostate cancer: systematic review and meta-analysis. BMJ 2016;352:i851. https://doi.org/10.1136/bmj.i851.

[110] Kapoor R, Moghanaki D, Rexrode S, Monzon B, Ray M, Hulick PR, et al. Quality Improvements of Veterans Health Administration Radiation Oncology Services Through Partnership for Accreditation With the ACR. J Am Coll Radiol 2018;15:1732–7. https://doi.org/10.1016/j.jacr.2018.06.029.

[111] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature 2020;585:357–62. https://doi.org/10.1038/s41586-020-2649-2.

[112] McKinney W. Data Structures for Statistical Computing in Python, Austin, Texas: 2010, p. 56–61. https://doi.org/10.25080/Majora-92bf1922-00a.

[113] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. J Mach Learn Res 2011;12:2825–30.

[114] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems n.d.

[115] Waskom M. seaborn: statistical data visualization. J Open Source Softw 2021;6:3021. https://doi.org/10.21105/joss.03021.

[116] Hunter JD. Matplotlib: A 2D Graphics Environment. Comput Sci Eng 2007;9:90–5. https://doi.org/10.1109/MCSE.2007.55.

[117] Amodei D, Olah C, Steinhardt J, Christiano P, Schulman J, Mané D. Concrete Problems in AI Safety 2016. https://doi.org/10.48550/arXiv.1606.06565.

[118] Hendrycks D, Carlini N, Schulman J, Steinhardt J. Unsolved Problems in ML Safety 2022. https://doi.org/10.48550/arXiv.2109.13916.

[119] Guo C, Pleiss G, Sun Y, Weinberger KQ. On Calibration of Modern Neural Networks 2017. https://doi.org/10.48550/arXiv.1706.04599.

[120] Nguyen A, Yosinski J, Clune J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. 2015 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR, Boston, MA, USA: IEEE; 2015, p. 427–36. https://doi.org/10.1109/CVPR.2015.7298640.

[121] McInnes L, Healy J, Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction 2020. https://doi.org/10.48550/arXiv.1802.03426.

[122] Maaten L van der, Hinton G. Visualizing Data using t-SNE. J Mach Learn Res 2008;9:2579–605.

[123] Chari T, Pachter L. The specious art of single-cell genomics. PLOS Comput Biol 2023;19:e1011288. https://doi.org/10.1371/journal.pcbi.1011288.

[124] UMAP Reproducibility — umap 0.5 documentation n.d. https://umap-learn.readthedocs.io/en/latest/reproducibility.html (accessed December 3, 2024).

[125] Betthauser L, Chajewska U, Diesendruck M, Pesala R. Discovering Distribution Shifts using Latent Space Representations 2022. https://doi.org/10.48550/arXiv.2202.02339.

[126] Henning C, D'Angelo F, Grewe BF. Are Bayesian neural networks intrinsically good at out-of-distribution detection? 2021. https://doi.org/10.48550/arXiv.2107.12248.

[127] Tohme T, Vanslette K, Youcef-Toumi K. Reliable neural networks for regression uncertainty estimation. Reliab Eng Syst Saf 2023;229:108811. https://doi.org/10.1016/j.ress.2022.108811.

[128] Rippel O, Mertens P, Merhof D. Modeling the Distribution of Normal Data in Pre-Trained Deep Features for Anomaly Detection. 2020 25th Int. Conf. Pattern Recognit. ICPR, 2021, p. 6726–33. https://doi.org/10.1109/ICPR48806.2021.9412109.

[129] Hendrycks D, Mazeika M, Dietterich T. Deep Anomaly Detection with Outlier Exposure 2019. https://doi.org/10.48550/arXiv.1812.04606.

[130] Oehri S, Ebert N, Abdullah A, Stricker D, Wasenmüller O. GenFormer -- Generated Images are All You Need to Improve Robustness of Transformers on Small Datasets 2024. https://doi.org/10.48550/arXiv.2408.14131.

[131] LeCun Y, Burges CJC, Cortes C. The MNIST Database of Handwritten Digits. Http://YannLecunCom/Exdb/Mnist/ n.d.

[132] Krizhevsky A. Learning Multiple Layers of Features from Tiny Images n.d.

[133] W. Nash, T. Sellers, S. Talbot, A. Cawthorn, W. Ford. The Population Biology of Abalone (Haliotis species) in Tasmania. I. Blacklip Abalone (H. rubra) from the North Coast and Islands of Bass Strait. Marine Research Laboratorties - Taroona: Department of Primary Industry and Fisheries, Tasmania; 1994.

[134] Omohundro SM. Five Balltree Construction Algorithms. ICSI Tech Rep TR-89-063 1989.

[135] Wightman R. PyTorch Image Models. GitHub Repos 2019. https://doi.org/10.5281/zenodo.4414861.

[136] Langselius O, Rumgay H, de Vries E, Whiteman DC, Jemal A, Parkin DM, et al. Global burden of cutaneous melanoma incidence attributable to ultraviolet radiation in 2022. Int J Cancer n.d.;n/a. https://doi.org/10.1002/ijc.35463.

[137] Zhou L, Zhong Y, Han L, Xie Y, Wan M. Global, regional, and national trends in the burden of melanoma and non-melanoma skin cancer: insights from the global burden of disease study 1990–2021. Sci Rep 2025;15:5996. https://doi.org/10.1038/s41598-025-90485-3.

[138] Viola KV, Tolpinrud WL, Gross CP, Kirsner RS, Imaeda S, Federman DG. Outcomes of Referral to Dermatology for Suspicious Lesions: Implications for Teledermatology. Arch Dermatol 2011;147:556–60. https://doi.org/10.1001/archdermatol.2011.108.

[139] Goyal M, Knackstedt T, Yan S, Hassanpour S. Artificial intelligence-based image classification methods for diagnosis of skin cancer: Challenges and opportunities. Comput Biol Med 2020;127:104065. https://doi.org/10.1016/j.compbiomed.2020.104065.

[140] Vaidya T, Zubritsky L, Alikhan A, Housholder A. Socioeconomic and geographic barriers to dermatology care in urban and rural US populations. J Am Acad Dermatol 2018;78:406–8. https://doi.org/10.1016/j.jaad.2017.07.050.

[141] Nazari S, Garcia R. Automatic Skin Cancer Detection Using Clinical Images: A Comprehensive Review. Life 2023;13:2123. https://doi.org/10.3390/life13112123.

[142] Grignaffini F, Barbuto F, Piazzo L, Troiano M, Simeoni P, Mangini F, et al. Machine Learning Approaches for Skin Cancer Classification from Dermoscopic Images: A Systematic Review. Algorithms 2022;15:438. https://doi.org/10.3390/a15110438.

[143] Xia M, Kheterpal MK, Wong SC, Park C, Ratliff W, Carin L, et al. Lesion identification and malignancy prediction from clinical dermatological images. Sci Rep 2022;12:15836. https://doi.org/10.1038/s41598-022-20168-w.

[144] Mundada MR, Seema S, J SB, Student SF. DeepDerm: Elevating Precision in Dermatological Diagnosis with Enhanced CNN. 2024 15th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT, 2024, p. 1–7. https://doi.org/10.1109/ICCCNT61001.2024.10724699.

[145] Han SS, Kim MS, Lim W, Park GH, Park I, Chang SE. Classification of the Clinical Images for Benign and Malignant Cutaneous Tumors Using a Deep Learning Algorithm. J Invest Dermatol 2018;138:1529–38. https://doi.org/10.1016/j.jid.2018.01.028.

[146] Salido JA, Ruiz C. Using deep learning for melanoma detection in dermoscopy images. Int J Mach Learn Comput 2018;8:61–8. https://doi.org/10.18178/ijmlc.2018.8.1.664.

[147] Fujisawa Y, Otomo Y, Ogata Y, Nakamura Y, Fujita R, Ishitsuka Y, et al. Deep-learning-based, computer-aided classifier developed with a small dataset of clinical images surpasses board-certified dermatologists in skin tumour diagnosis. Br J Dermatol 2019;180:373–81. https://doi.org/10.1111/bjd.16924.

[148] Tschandl P, Rosendahl C, Akay BN, Argenziano G, Blum A, Braun RP, et al. Expert-Level Diagnosis of Nonpigmented Skin Cancer by Combined Convolutional Neural Networks. JAMA Dermatol 2019;155:58–65. https://doi.org/10.1001/jamadermatol.2018.4378.

[149] Combalia M, Codella N, Rotemberg V, Carrera C, Dusza S, Gutman D, et al. Validation of artificial intelligence prediction models for skin cancer diagnosis using dermoscopy images: the 2019 International Skin Imaging Collaboration Grand Challenge. Lancet Digit Health 2022;4:e330–9. https://doi.org/10.1016/S2589-7500(22)00021-8.

[150] Wolf JA, Moreau JF, Akilov O, Patton T, English JC III, Ho J, et al. Diagnostic Inaccuracy of Smartphone Applications for Melanoma Detection. JAMA Dermatol 2013;149:422–6. https://doi.org/10.1001/jamadermatol.2013.2382.

[151] Wongvibulsin S, Yan MJ, Pahalyants V, Murphy W, Daneshjou R, Rotemberg V. Current State of Dermatology Mobile Applications With Artificial Intelligence Features. JAMA Dermatol 2024;160:646–50. https://doi.org/10.1001/jamadermatol.2024.0468.

[152] Sun MD, Kentley J, Mehta P, Dusza S, Halpern AC, Rotemberg V. Accuracy of commercially available smartphone applications for the detection of melanoma. Br J Dermatol 2022;186:744–6. https://doi.org/10.1111/bjd.20903.

[153] Wang SQ, Dusza SW, Scope A, Braun RP, Kopf AW, Marghoob AA. Differences in Dermoscopic Images from Nonpolarized Dermoscope and

Polarized Dermoscope Influence the Diagnostic Accuracy and Confidence Level: A Pilot Study. Dermatol Surg 2008;34:1389.

[154] Hanlon KL, Wei G, Correa-Selm L, Grichnik JM. Dermoscopy and skin imaging light sources: a comparison and review of spectral power distribution and color consistency. J Biomed Opt 2022;27:080902. https://doi.org/10.1117/1.JBO.27.8.080902.

[155] Daneshjou R, Vodrahalli K, Novoa RA, Jenkins M, Liang W, Rotemberg V, et al. Disparities in dermatology AI performance on a diverse, curated clinical image set. Sci Adv 2022;8:eabq6147. https://doi.org/10.1126/sciadv.abq6147.

[156] Mehta D, Primiero C, Betz-Stablein B, Nguyen TD, Gal Y, Bowling A, et al. Multi-task AI models in dermatology: Overcoming critical clinical translation challenges for enhanced skin lesion diagnosis. J Eur Acad Dermatol Venereol n.d.;n/a. https://doi.org/10.1111/jdv.20551.

[157] Kim S, Gaibor E, Matejek B, Haehn D. Melanoma Detection with Uncertainty Quantification 2024. https://doi.org/10.48550/arXiv.2411.10322.

[158] Shamsi A, Asgharnezhad H, Bouchani Z, Jahanian K, Saberi M, Wang X, et al. A novel uncertainty-aware deep learning technique with an application on skin cancer diagnosis. Neural Comput Appl 2023;35:22179–88. https://doi.org/10.1007/s00521-023-08930-1.

[159] Schreyer WM, Anderson C, Thompson RF. Generalization is not a universal guarantee: Estimating similarity to training data with an ensemble out-of-distribution metric 2025. https://doi.org/10.48550/arXiv.2502.16329.

[160] Tschandl P, Rosendahl C, Kittler H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci Data 2018;5:180161. https://doi.org/10.1038/sdata.2018.161.

[161] Ricci Lara MA, Rodríguez Kowalczuk MV, Lisa Eliceche M, Ferraresso MG, Luna DR, Benitez SE, et al. A dataset of skin lesion images collected in Argentina for the evaluation of AI tools in this population. Sci Data 2023;10:712. https://doi.org/10.1038/s41597-023-02630-0.

[162] Pacheco AGC, Lima GR, Salomão AS, Krohling B, Biral IP, de Angelo GG, et al. PAD-UFES-20: A skin lesion dataset composed of patient data and clinical images collected from smartphones. Data Brief 2020;32:106221. https://doi.org/10.1016/j.dib.2020.106221.

[163] Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, et al. Model Cards for Model Reporting. Proc. Conf. Fairness Account. Transpar., New York, NY, USA: Association for Computing Machinery; 2019, p. 220–9. https://doi.org/10.1145/3287560.3287596.

[164] Howard JJ, Sirotin YB, Tipton JL, Vemury AR. Reliability and Validity of Image-Based and Self-Reported Skin Phenotype Metrics. IEEE Trans Biom

Behav Identity Sci 2021;3:550–60. https://doi.org/10.1109/TBIOM.2021.3123550.

[165] Weir VR, Dempsey K, Gichoya JW, Rotemberg V, Wong A-KI. A survey of skin tone assessment in prospective research. Npj Digit Med 2024;7:191. https://doi.org/10.1038/s41746-024-01176-8.

[166] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library 2019. https://doi.org/10.48550/arXiv.1912.01703.

[167] Hendrycks D, Dietterich T. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations 2019. https://doi.org/10.48550/arXiv.1903.12261.

[168] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conf. Comput. Vis. Pattern Recognit., 2009, p. 248–55. https://doi.org/10.1109/CVPR.2009.5206848.

[169] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition 2015. https://doi.org/10.48550/arXiv.1512.03385.

[170] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows 2021. https://doi.org/10.48550/arXiv.2103.14030.

[171] Kurtansky NR, D'Alessandro BM, Gillis MC, Betz-Stablein B, Cerminara SE, Garcia R, et al. The SLICE-3D dataset: 400,000 skin lesion image crops extracted from 3D TBP for skin cancer detection. Sci Data 2024;11:884. https://doi.org/10.1038/s41597-024-03743-w.

[172] Yan S, Yu Z, Primiero C, Vico-Alonso C, Wang Z, Yang L, et al. A multimodal vision foundation model for clinical dermatology. Nat Med 2025;31:2691–702. https://doi.org/10.1038/s41591-025-03747-y.

[173] Rasmy L, Xiang Y, Xie Z, Tao C, Zhi D. Med-BERT: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. Npj Digit Med 2021;4:86. https://doi.org/10.1038/s41746-021-00455-y.

[174] Avsec Ž, Agarwal V, Visentin D, Ledsam JR, Grabska-Barwinska A, Taylor KR, et al. Effective gene expression prediction from sequence by integrating long-range interactions. Nat Methods 2021;18:1196–203. https://doi.org/10.1038/s41592-021-01252-x.

[175] Dalla-Torre H, Gonzalez L, Mendoza-Revilla J, Lopez Carranza N, Grzywaczewski AH, Oteri F, et al. Nucleotide Transformer: building and evaluating robust foundation models for human genomics. Nat Methods 2025;22:287–97. https://doi.org/10.1038/s41592-024-02523-z.

[176] Gou J, Yu B, Maybank SJ, Tao D. Knowledge Distillation: A Survey. Int J Comput Vis 2021;129:1789–819. https://doi.org/10.1007/s11263-021-01453-z.

[177] Futoma J, Simons M, Panch T, Doshi-Velez F, Celi LA. The myth of generalisability in clinical research and machine learning in health care. Lancet Digit Health 2020;2:e489–92. https://doi.org/10.1016/S2589-7500(20)30186-2.

[178] D'Amour A, Heller K, Moldovan D, Adlam B, Alipanahi B, Beutel A, et al. Underspecification Presents Challenges for Credibility in Modern Machine Learning 2020. https://doi.org/10.48550/arXiv.2011.03395.

[179] Vandersluis R, Savulescu J. The selective deployment of AI in healthcare: An ethical algorithm for algorithms. Bioethics 2024;38:391–400. https://doi.org/10.1111/bioe.13281.

[180] Daneshjou R, Smith MP, Sun MD, Rotemberg V, Zou J. Lack of Transparency and Potential Bias in Artificial Intelligence Data Sets and Algorithms: A Scoping Review. JAMA Dermatol 2021;157:1362–9. https://doi.org/10.1001/jamadermatol.2021.3129.

[181] Adamson AS, Smith A. Machine Learning and Health Care Disparities in Dermatology. JAMA Dermatol 2018;154:1247–8. https://doi.org/10.1001/jamadermatol.2018.2348.

[182] Liu Y, Jain A, Eng C, Way DH, Lee K, Bui P, et al. A deep learning system for differential diagnosis of skin diseases. Nat Med 2020;26:900–8. https://doi.org/10.1038/s41591-020-0842-3.

[183] Hartman RI, Trepanowski N, Chang MS, Tepedino K, Gianacas C, McNiff JM, et al. Multicenter prospective blinded melanoma detection study with a handheld elastic scattering spectroscopy device. JAAD Int 2024;15:24–31. https://doi.org/10.1016/j.jdin.2023.10.011.

[184] Groh M, Badri O, Daneshjou R, Koochek A, Harris C, Soenksen LR, et al. Deep learning-aided decision support for diagnosis of skin disease across skin tones. Nat Med 2024;30:573–83. https://doi.org/10.1038/s41591-023-02728-3.