Adversarially-Regularized Mixed Effects Deep Learning (ARMED) Models Improve Interpretability, Performance, and Generalization on Clustered (non-*iid*) Data

Kevin P. Nguyen, Alex H. Treacher, and Albert A. Montillo

***Abstract*—Natural science datasets frequently violate assumptions of independence. Samples may be clustered (e.g., by study site, subject, or experimental batch), leading to spurious associations, poor model fitting, and confounded analyses. While largely unaddressed in deep learning, this problem has been handled in the statistics community through mixed effects models, which separate cluster-invariant *fixed* effects from cluster-specific *random* effects. We propose a general-purpose framework for Adversarially-Regularized Mixed Effects Deep learning (ARMED) models through non-intrusive additions to existing neural networks: 1) an adversarial classifier constraining the original model to learn only cluster-invariant features, 2) a random effects subnetwork capturing cluster-specific features, and 3) an approach to apply random effects to clusters unseen during training. We apply ARMED to dense, convolutional, and autoencoder neural networks on 4 datasets including simulated nonlinear data, dementia prognosis and diagnosis, and live-cell image analysis. Compared to prior techniques, ARMED models better distinguish confounded from true associations in simulations and learn more biologically plausible features in clinical applications. They can also quantify inter-cluster variance and visualize cluster effects in data. Finally, ARMED matches or improves performance on data from clusters seen during training (5-28% relative improvement) and generalization to unseen clusters (2-9% relative improvement) versus conventional models.**

***Index Terms*—Generalization, interpretability, mixed effects model, multilevel model, biomedical imaging, clinical data.**

I. Introduction

In predictive modeling, one often assumes that data is independent and identically distributed (*iid*), such that no samples are correlated or interdependent. However, this assumption is frequently violated in the natural sciences when samples are clustered. For example, many multi-site neurological studies acquire cognitive scores using a different human rater at each site, which are subject to inter-rater differences [1], [2], [3]. As a result, these measurements have inherent intra-site correlation and inter-site variability. Another example is medical imaging, such as magnetic resonance imaging (MRI), where differences in imaging protocol and scanner hardware lead to substantial site effects in multi-site studies [4], [5]. Clustering also occurs in biological data, such as when measurements are collected across different experimental batches [6] or tissue samples [7], and in environmental data collected across locations [8].

If not properly handled in analysis, the cluster effects of non-*iid* data can lead to erroneous conclusions. The so-called Simpson's paradox occurs when an association between two variables appears, disappears, or even reverses when analysis is performed at the population level versus when analysis is stratified by cluster, indicating a confounding effect. This situation can lead to Type I (false positive) or Type II (false negative) findings in many situations, including clinical studies [9], proteomics [7], and economics [10].

Despite these consequences, the machine learning community has generally ignored the problems underlying non-*iid* data. Meanwhile, the traditional statistics community has addressed clustered data with *mixed effects* models, which learn a combination of *fixed* and *random* effects. The most common of these is the linear mixed effects (LME) model, which builds upon the basic linear regression model. Suppose that we have data $X \in \mathbb{R}^{n\times p}$ with n samples and p independent variables (features), originating from c clusters, and a dependent variable (target) $y \in \mathbb{R}^{n\times 1}$. We can define the following LME regression model; for a sample $i = 1, 2, \ldots, n$ originating from cluster $j = 1, 2, \ldots, c$ we have:

$$\begin{aligned}\hat{y}_i &= \beta_0 + x_{i,1}\beta_1 + \ldots + x_{i,p}\beta_p \\ &\quad + u_{j,0} + x_{i,1}u_{j,1} + \ldots + x_{i,p}u_{j,p} + \epsilon_i \\ &= \beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta} + u_{j,0} + \boldsymbol{x}_i^\top \boldsymbol{u}_j + \epsilon_i \end{aligned} \tag{1}$$

where $\hat{y}_i$ is the predicted target, $\boldsymbol{x}_i^\top = [x_{i,1}, \ldots, x_{i,p}]$ is the p-dimensional feature vector of the i^{th} sample from X, and ϵ_i is the residual. The model contains two types of weights.

Manuscript received 22 April 2022; revised 6 October 2022; accepted 18 December 2022. Date of publication 12 January 2023; date of current version 5 June 2023. This work was supported in part by Lyda Hill Foundation and in pat by the National Institute Of General Medical Sciences of the National Institutes of Health under Award Number R01GM144486. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. Recommended for acceptance by J. Bruna. *(Corresponding author: Albert A. Montillo.)*

The authors are with the Lyda Hill Department of Bioinformatics, University of Texas Southwestern Medical Center, Dallas, TX 75390 USA (e-mail: kevin3.nguyen@utsouthwestern.edu; alex.treacher@utsouthwestern.edu; albert.montillo@utsouthwestern.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TPAMI.2023.3234291.

Digital Object Identifier 10.1109/TPAMI.2023.3234291

The fixed effect intercept β_0 and slopes $\boldsymbol{\beta}^\top = [\beta_1, ..., \beta_p]$ are cluster-*invariant* and apply globally to all samples. The random effects weights include the intercept $u_{j,0}$ and slopes $\boldsymbol{u}_j^\top = [u_{j,1}, ..., u_{j,p}]$, whose values are *specific* to each cluster j. The random effect weight values are assumed to follow a random distribution, most often a multivariate normal distribution with mean 0, i.e. $\boldsymbol{u} \sim N(0, \sigma)$. Consequently, the random effect weights $\boldsymbol{u}$ can be interpreted as cluster-specific offsets from the fixed effect weights $\boldsymbol{\beta}$. The LME model separates the variance explained by global associations from the inter-cluster variance, controls for correlated samples, and improves weight estimates [8], [11]. Unfortunately, proper handling of mixed effects in deep learning, delivering all of these gains, has gone unanswered. In this work, we describe how appropriate handling of mixed effects can address the inadequacies of deep learning models when applied to clustered data.

A. Related Work

Previous deep learning approaches for clustered data have key limitations. A naive but prevalent strategy is to insert cluster information as an additional, one-hot encoded covariate [12]. This increases data dimensionality, which may cause overfitting with a high number of clusters c [12], [13], and it entangles the cluster-invariant and cluster-specific features within the model weights, hampering model interpretation. *Domain adaptation* techniques train a model on a source domain (i.e., cluster), then adapt it in a subsequent training step to a target domain [14]. This yields an adapted model for each target domain but not a single unified model. It also does not scale easily to many domains or separate domain-invariant from domain-specific features, which also limits interpretability. *Domain generalization* techniques address some of these weaknesses by producing a single generalized model agnostic to domain differences. Earlier approaches used gradient reversal layers, which modify backpropagation to maximize domain invariance [15], [16]. Other methods use *meta-learning* to guide gradient descent in a direction that reduces the loss for all domains [17], [18]. However, these involve second-order optimization which vastly increase computational cost. A third category of domain generalization methods uses an adversarial classifier [19], [20]. The adversarial classifier learns to classify domains from the latent features of the main model, while the main model learns features that maximize domain classification error. The common limitation of all domain generalization techniques is that they produce a model that has *only* learned the domain-invariant features (fixed effects), while domain-specific information (random effects), are discarded. Our proposed framework captures this ignored information in a separate random effects subnetwork, while an adversarially-regularized subnetwork captures global fixed effects. We show that this adds predictive value and allows users to understand more about cluster variance in their data.

To date, there have been three prior approaches to incorporate mixed effects into deep learning. Xiong et al. proposed MeNet, a mixed effects convolutional neural network (CNN), for a gaze estimation dataset containing repeated images per subject [21]. While improving accuracy, the method requires an expensive expectation-maximization algorithm with inversion of large covariance matrices ($n_j \times n_j$ where n_j is the number of samples within each cluster). MeNet also only models random slopes and not intercepts. Next, Tran et al. proposed DeepGLMM, a mixed effects approach for dense feedforward neural networks (DFNNs) using Bayesian deep learning and variational inference for more efficient training [22]. Though theoretically capable of modeling both random slopes and intercepts, their applications only used models with random intercepts. Their experiments also lacked comparisons with other deep learning methods. Finally, Simchoni et al. proposed LMMNN, a mixed effects approach for both DFNNs and CNNs, and demonstrated a performance benefit across multiple applications. However, LMMNN is trained using expensive covariance matrix inversions, and their real-world applications use only random intercepts [13].

There are several common limitations across the MeNet, DeepGLMM, and LMMNN approaches. These methods prioritize the improvement of predictive performance and ignore the additional interpretability afforded by mixed effects, such as quantification and visualization of inter-cluster variance. They also lack explicit guidance of the fixed effects to be cluster-invariant, so their resilience to confounded associations is unclear. Additionally, none of these works demonstrate models with both random slopes and intercepts or unsupervised learning models, such as autoencoders. Lastly, there are no specific recommendations for applying these models to new data that does not originate from the same clusters seen during training, which limits real-world utility where data from new clusters is frequently encountered.

B. Contributions

We propose an Adversarially-Regularized Mixed Effects Deep learning (ARMED) framework that generalizes across model archetypes and alleviates the shortcomings of the previous approaches. This framework contains three components that can be readily added to a conventional deep learning model with minimal modification of the existing architecture. First, inspired by domain generalization, we employ an adversarial classifier to regularize the model to learn cluster-invariant fixed effects. We show through simulations that this improves the separation of cluster-specific, potentially confounded features from cluster-invariant features. Second, we introduce a Bayesian random effects subnetwork to learn the cluster-specific features, and we demonstrate how it can quantify and visualize the variance across clusters. Third, we add another classifier which infers random effects for so-called “unseen cluster” data, where samples originate outside the clusters seen during training. We demonstrate the advantages of our framework across 4 test cases using DFNNs, CNNs, and convolutional autoencoders, including simulations and three biomedical examples. In each case, we achieve not only the separation and identification of fixed and random effects, but also better predictive performance on data from seen clusters and better generalization to unseen clusters.

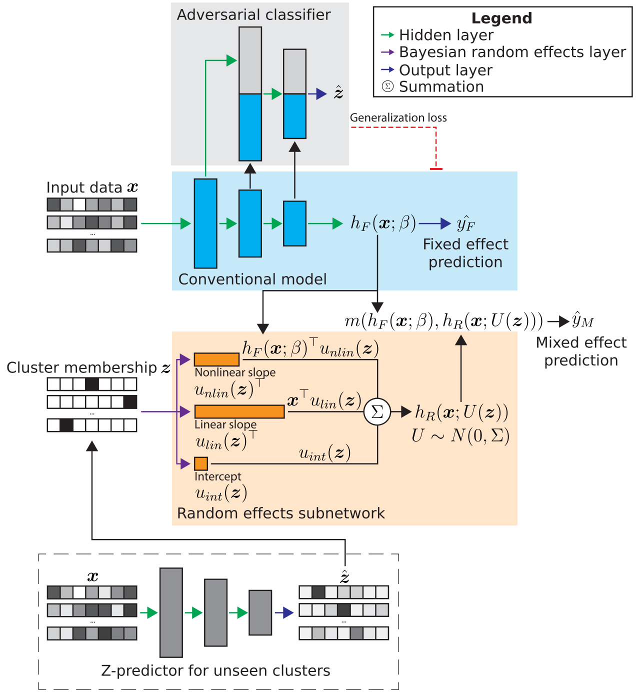
Adversarial classifier
Legend
Hidden layer
Bayesian random effects layer
Output layer
Summation
$\hat{\boldsymbol{z}}$
Generalization loss
Input data $\boldsymbol{x}$
$h_F(\boldsymbol{x};\beta) \rightarrow \hat{y}_F$
Fixed effect prediction
Conventional model
$m(h_F(\boldsymbol{x};\beta), h_R(\boldsymbol{x};U(\boldsymbol{z}))) \rightarrow \hat{y}_M$
Mixed effect prediction
Cluster membership $\boldsymbol{z}$
$h_F(\boldsymbol{x};\beta)^\top u_{nlin}(\boldsymbol{z})$
Nonlinear slope
$u_{nlin}(\boldsymbol{z})^\top$
$\boldsymbol{x}^\top u_{lin}(\boldsymbol{z})$
Σ
$h_R(\boldsymbol{x};U(\boldsymbol{z}))$
$U \sim N(0,\Sigma)$
Linear slope
$u_{lin}(\boldsymbol{z})^\top$
$u_{int}(\boldsymbol{z})$
Intercept
$u_{int}(\boldsymbol{z})$
Random effects subnetwork
$\boldsymbol{x}$
$\hat{\boldsymbol{z}}$
Z-predictor for unseen clusters

Fig. 1. The ARMED framework for a generic neural network. The conventional model (blue area) predicts $\hat{y}_F$ from the data sample $\boldsymbol{x}$. Cluster membership of the sample is one-hot encoded into $\boldsymbol{z}$. The fixed effects subnetwork (blue + gray areas) is constructed by adding an adversarial classifier (gray area) to predict cluster membership $\hat{\boldsymbol{z}}$. The original model is penalized through the generalization loss for learning features that allow cluster membership prediction. The random effects subnetwork (orange area) uses Bayesian layers to learn cluster-specific weights, dependent on $\boldsymbol{z}$, that follow zero-mean multivariate normal distributions. These weights can be formulated as nonlinear slopes multiplied by the fixed effects latent representation $h_F(X;\beta)$, linear slopes multiplied by X, and/or intercepts. The fixed and random effects are combined with a mixing function $m(\ldots)$. For prediction on data from clusters unseen during training, $\boldsymbol{z}$ is inferred with a classifier (Z-predictor) trained on data from seen clusters.

II. Methods

In general, a conventional feed-forward neural network computes a nonlinear transformation of the data X through its layers (Fig. 1, blue area). We denote the output of the penultimate layer as $h(X,\beta) \in \mathbb{R}^{n\times q}$, where β contains all learned weights up to and including this layer and q is the number of neurons. For a typical regression or classification task, a final linear or softmax output layer o then transforms $h(X,\beta)$ into the final prediction output $\hat{\boldsymbol{y}}$:

$$\hat{\boldsymbol{y}} = o(h(X,\beta))$$

During training, the model finds the weights, β, which minimize a given loss function quantifying the **e**rror for the predictive task, $\mathcal{L}_e(\boldsymbol{y},\hat{\boldsymbol{y}})$.

To encode cluster membership information for a dataset with n samples and c clusters, we introduce a one-hot encoded design matrix $Z \in \mathbb{R}^{n\times c}$, where $Z_{i,j} = 1$ if sample i belongs to cluster j and $Z_{i,j} = 0$ otherwise. The following sections present a description of the ARMED framework components, agnostic to model architecture. These components include the fixed effects subnetwork h_F, including a conventional neural network and an adversarial classifier a that together learn cluster-invariant features, the random effects subnetwork h_R for learning Z-dependent cluster-specific features, the mixing function m that combines the fixed and random effects for prediction, and the Z-predictor used to apply random effects to new clusters.

A. Fixed Effects Subnetwork

First, we add an adversarial classifier (Fig. 1, gray area) to the conventional model (Fig. 1, blue area) to enforce the learning of cluster-invariant fixed effects, creating the *F*ixed effects subnetwork $h_F(X;\beta)$. This is based on the adversarial learning technique for domain generalization [19], [20]. For a neural network with L layers, let

$$H_F(X;\beta_F) = [h_{F,1}(X;\beta_{F,1}),\ldots,h_{F,L}(X;\beta_{F,L})]$$

represent the collected outputs of each layer, where $\beta_{F,l}$ contains the weights up to the lth layer. We define an adversarial classifier a which predicts a sample's cluster membership from these layer outputs, $\hat{Z} = a(H_F(X;\beta_F);\beta_A)$, where β_A contains the weights for this adversary. The adversary is trained to minimize the categorical cross-entropy loss:

$$\begin{aligned}&\mathcal{L}_{CCE}(Z,\hat{Z})\\&= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} Z_{i,j}\log(\hat{Z}_{i,j}) + (1-Z_{i,j})\log(1-\hat{Z}_{i,j})\end{aligned}$$

Meanwhile, the main model is penalized for learning features that allow the adversary to predict cluster membership. It must *maximize* this cross-entropy, which we call the cluster generalization loss. The resulting training objective of the fixed effects subnetwork is

$$\mathcal{L}_e(\boldsymbol{y},\hat{\boldsymbol{y}}_F) - \lambda_g\mathcal{L}_{CCE}(Z,\hat{Z}) \tag{2}$$

where the hyperparameter λ_g controls the weight of the generalization loss. We use $\hat{\boldsymbol{y}}_F$ to denote the prediction output of this fixed effects subnetwork.

B. Random Effects Subnetwork

We next define a second subnetwork to learn the *R*andom effects, $h_R(X;U(Z))$ with cluster-specific weights $U(Z)$ (Fig. 1, orange area). The cluster-specific values for each individual weight $u(Z)$ in $U(Z)$ are assumed to follow a normal distribution with mean 0, i.e. $u(Z) \sim N(0,\sigma)$ where σ represents the inter-cluster variance of each weight. Collectively, Σ contains the inter-cluster variance for all weights in $U(Z)$. We implement these weights using a Bayesian formulation. We specify a zero-mean normal prior distribution for each weight $p(U) \sim N(0,\sigma_p)$ with the fixed prior variance σ_p as a global hyperparameter. The posterior distribution $p(U|X)$ is then learned through variational inference, which reframes Bayesian modeling as an optimization problem that can be efficiently handled through gradient descent [23], [24]. The objective of variational inference is to learn a surrogate posterior $q(U)$, here a multivariate normal distribution, which closely approximates the true posterior $p(U|X)$, where "closeness" is measured by the Kullback-Leibler (KL) divergence:

$$D_{\mathrm{KL}}(q(U)||p(U|X)) = \int q(U) \log \frac{q(U)}{p(U|X)} dU$$

Minimizing $D_{\mathrm{KL}}(q(U)||p(U|X))$ directly is impossible because computing the posterior through Bayes Rule, $p(U|X) = \frac{p(X|U)p(U)}{p(X)}$, involves the intractable marginalization $p(X)$. Instead, variational inference maximizes the Evidence Lower Bound (ELBO) which contains fully tractable and differentiable quantities:

$$\mathrm{ELBO} = \mathbb{E}_q[\log p(X|U)] - D_{\mathrm{KL}}(q(U)||p(U))$$

where the first right-hand term is the log-likelihood and the second term is the KL divergence between the surrogate posterior and the prior. For gradient descent, we minimize the negative ELBO and let $\mathcal{L}_e(\boldsymbol{y}, \hat{\boldsymbol{y}})$ represent the first term, i.e. the negative log-likelihood loss. This yields the following objective:

$$\mathcal{L}_e(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \lambda_K D_{\mathrm{KL}}(q(U)||p(U)) \tag{3}$$

with the hyperparameter λ_K controlling the strength of the regularization to the prior and $D_{\mathrm{KL}}(q(U)||p(U)) = \int q(U) \log \frac{q(U)}{p(U)} dU$. Note that our method based on variational inference does not require expensive inversions of covariance matrices as in MeNet and LMMNN [13], [21].

The architecture of this subnetwork will depend on the types of random effects to be modeled. Nonlinear random effects slopes can be modeled as weights multiplied by the fixed effects latent representation $h_F(X; \beta)$:

$$h_{R,nlin}(\boldsymbol{x}_i; u_{nlin}(\boldsymbol{z}_i)) = h_F(\boldsymbol{x}_i; \beta)^\top u_{nlin}(\boldsymbol{z}_i) \tag{4}$$

where $\boldsymbol{z}_i$ and $\boldsymbol{x}_i$ are the rows in Z and X for the ith sample and $u_{nlin}(\boldsymbol{z}_i) \in \mathbb{R}^{q\times 1}$ returns the slopes for cluster $\boldsymbol{z}_i$, q being the number of output neurons of $h_F(\boldsymbol{x}_i; \beta)$. A random intercept is modeled simply as a weight:

$$h_{R,int}(u(\boldsymbol{z}_i)) = u_{int}(\boldsymbol{z}_i) \tag{5}$$

where $u_{int}(\boldsymbol{z}_i)$ is a scalar value. Additionally, for tabular data, we can model linear random effects slopes multiplied directly with X, which allows each slope to be interpreted directly with respect to a corresponding input variable:

$$h_{R,lin}(\boldsymbol{x}_i; u_{lin}(\boldsymbol{z}_i)) = \boldsymbol{x}_i^\top u_{lin}(\boldsymbol{z}_i) \tag{6}$$

where $u_{lin}(\boldsymbol{z}_i) \in \mathbb{R}^{p\times 1}$ returns the slopes for cluster $\boldsymbol{z}_i$. The random effects subnetwork outputs the sum of these random effects:

$$\begin{aligned} h_R(\boldsymbol{x}_i; U(\boldsymbol{z}_i)) = {} & h_{R,nlin}(\boldsymbol{x}_i; u_{nlin}(\boldsymbol{z}_i)) \\ & + h_{R,lin}(\boldsymbol{x}_i; u_{lin}(\boldsymbol{z}_i)) \\ & + h_{R,int}(u_{int}(\boldsymbol{z}_i)) \end{aligned} \tag{7}$$

These three cases will apply to most models with a dense penultimate layer producing a vector-form $h_F(X; \beta)$. For models such as autoencoders, we describe in the Supplemental Materials, (available online), how random effects can be readily applied across multiple convolutional layers (Section 3.1.3, Fig. S5).

C. Combining Fixed and Random Effects

We construct the final ARMED model by combining the outputs of the fixed effects and random effects subnetworks. In the linear model of (1), random and fixed effects were combined through addition. For greater flexibility here, we substitute the addition in (1) with a more general mixing function $m(\ldots)$.

$$\hat{\boldsymbol{y}}_M = m(h_F(X; \beta), h_R(X; U(Z))) \tag{8}$$

For example, in the following binary classification applications, we use a nonlinear analog of (1). We add $h_R(X; U(Z))$ to the logit of $\hat{\boldsymbol{y}}_F$ (equal to $h_F(\boldsymbol{x}_i; \beta)^\top \beta_L$ where β_L are the weights of the output layer), then apply the sigmoid activation function:

$$\hat{\boldsymbol{y}}_M = sigmoid\left(h_F(\boldsymbol{x}_i; \beta)^\top \beta_L + h_R(\boldsymbol{x}_i; U(\boldsymbol{z}_i))\right)$$

The objective function is obtained by combining (2) and (3):

$$\begin{aligned} & \mathcal{L}_e(\boldsymbol{y}, \hat{\boldsymbol{y}}_M) + \lambda_F \mathcal{L}_e(\boldsymbol{y}, \hat{\boldsymbol{y}}_F) \\ & \quad - \lambda_g \mathcal{L}_{CCE}(Z, \hat{Z}) + \lambda_K D_{\mathrm{KL}}(q(U)||p(U)) \end{aligned} \tag{9}$$

The second term ensures that the fixed effect subnetwork will still be capable of prediction on its own so that the fixed effect features will be meaningful in later analyses. The loss weight $\lambda_F < 1$ balances the fixed effect error with the mixed effect error $\mathcal{L}_e(\boldsymbol{y}, \hat{\boldsymbol{y}}_M)$.

ARMED includes these hyperparameters: the generalization loss weight λ_g, the KL divergence weight λ_K, the fixed effect prediction error weight λ_F, and the prior distribution variance σ_p. Usage of linear versus nonlinear slopes must also be considered. In practice, we find that these can be easily tuned for model performance using standard hyperparameter optimization approaches, such as random search or Bayesian optimization, and appropriate cross-validation.

D. Prediction on Unseen Clusters

The previous mixed effects deep learning approaches provide no method for using the learned random effects when predicting on data not from clusters seen during training, i.e. not included in Z [13], [21], [22]. The authors of LMMNN propose to use only the fixed effects of their model on unseen clusters [13]. While the learned fixed effects, by definition, represent population-average associations, new data is not necessarily free of cluster effects and performance may be improved by fully utilizing the learned random effects. We propose to infer Z for unseen cluster data using a classifier we call the *Z-predictor*. We train this classifier to predict Z from X on the data from seen clusters, then use it to infer Z for data from unseen clusters. The unthresholded softmax predictions from the classifier provide a weighted combination of seen clusters that are most similar to each unseen cluster sample. In our applications, the Z-predictor uses the same architecture as the adversarial classifier.

E. Applications

1) Applications of ARMED to Dense Feedforward Neural Networks: Our first architectural application of ARMED is to a dense feedforward neural network (DFNN), which is suited to tabular data such as clinical measurements or pre-engineered image features. We describe the specifics of the ARMED-DFNN architecture in Fig. S1 and the Supplemental Materials 3.1.1, available online.

Spiral Classification Simulations: First, we evaluated the ARMED-DFNN on a simulated classification problem where cluster effects can be controlled, model-learned information can be compared to ground truth, and known confounded features can be added. The simulations are built upon the well-known spiral classification problem, where points must be classified into one of two spirals based on their coordinates x_1 and x_2 [25]. We simulated a nonlinear random effect by dividing the points into 10 clusters and randomly varied the spiral radius across clusters (Fig. S2). There were 3 variations of this simulation: 1) spiral radii varied across clusters, 2) spiral radii varied across clusters and spiral labels were inverted in half of the clusters (a more severe random effect), and 3) spiral radii varied across clusters and 2 known confounded probe features x_3 and x_4 were added. These probes created a spurious association between cluster and label but were not associated with the underlying spiral functions. Further details on these simulations can be found in the Supplemental Materials 3.2, available online. Because we have defined the random effects to be nonlinear, we used an ARMED-DFNN architecture with *nonlinear* random slopes ((4)) and a random intercept ((5)).

To test the ability of the fixed effects subnetwork to correctly downweight these confounded probes, we measured feature importance by computing the gradient of the model output with respect to the input features [26], [27]. Features with larger gradient magnitudes are more important in forming the model output. We compared the importance of each confounded probe (x_3 and x_4) to that of the least important true feature (x_1 or x_2).

Mild Cognitive Impairment Conversion Prediction: For a complementary real-world application, the ARMED-DFNN was used to predict the future development of full Alzheimer's Disease (AD) in subjects with mild cognitive impairment (MCI). MCI is an early stage of cognitive decline that may progress to dementia. Our target was to distinguish progressive MCI (pMCI), where a subject converts to AD within 24 months of baseline observation, from stable MCI (sMCI), where the subject does not convert within 24 months. We used data from the Alzheimer's Disease Neuroimaging Initiative, which includes baseline demographic information, cognitive scores, neuroimaging measurements, and biomarker measurements, as well as longitudinal diagnoses for each participant, acquired with informed consent and institutional review board approval (Supplemental Materials 3.3.1, available online). The training dataset came from the largest 20 study sites, and we used site as the random effect cluster. Inter-site variance has been shown to affect cognitive scores, which are sensitive to judgments by human raters, and neuroimaging, which is sensitive to MRI scanner parameters [1], [3], [28]. We held out the remaining 34 sites to evaluate model performance on sites unseen during training. Performance metrics included area under the receiver operating characteristic curve (AUROC), balanced accuracy, sensitivity, and specificity. For this application, we used an architecture with *linear* random slopes ((6)) and a random intercept ((5)). These were chosen to allow direct interpretation of the learned random slopes and inter-site variance for each input feature.

As with the spiral simulations, we subsequently added simulated confounded probe features to test how well each model could downweight known confounded features. We generated 5 confounded probes that were nonlinearly associated with site and with the probability of being labeled pMCI but had no real biological relevance (Supplemental Materials 3.3.1, available online). We then compared how highly each model ranked the probes based on feature importance (gradient magnitudes).

2) Application of ARMED to Convolutional Neural Networks: We next applied our approach to a convolutional neural network (CNN), another important deep learning archetype, creating an ARMED-CNN capable of learning nonlinear random slopes and random intercepts. Architecture details are described in Fig. S3 and Supplemental Materials 3.1.2, available online.

We applied the ARMED-CNN to the classification of AD versus cognitively normal (CN) structural MRI, with study site as the random effect cluster. We acquired T1-weighted MRI from 12 sites in the ADNI dataset (inclusion criteria and preprocessing details are in Supplemental Materials 3.3.2, available online). These 12 sites were selected to emphasize the confounding site effect, where sites using General Electric MRI scanners had a greater proportion of AD subjects compared to sites using Philips or Siemens scanners (Table S1). The remaining 51 sites were held out to evaluate performance on sites unseen during training. We extracted a two-dimensional coronal slice through the hippocampi from each image. Performance metrics included AUROC, balanced accuracy, sensitivity, and specificity.

3) Application of ARMED to Autoencoders: To demonstrate our framework on unsupervised learning models, we developed a mixed effects autoencoder. Our fourth application was the melanoma live-cell image compression and phenotypic classification problem described in [6]. In this work, the authors used a convolutional autoencoder to compress the images into a vector latent representation, then trained a classifier to label cells as having either high or low metastatic efficiency. They revealed that batch effects are prominent in this dataset, due to discrepancies between image batches acquired across different days, and that the latent representations strongly segregated by batch. The dataset is described further in Supplemental Materials 3.4, available online. The training data from the melanoma cell image dataset contained images acquired over 13 days (batches), and the remaining 11 days were held out as unseen batches.

We extended their autoencoder architecture by connecting the metastatic efficiency classifier directly to the autoencoder and training the autoencoder-classifier (AEC) end-to-end. We then applied our ARMED framework to create an ARMED-AEC, containing a fixed effects subnetwork that produces batch-invariant latent representations and a random effects subnetwork that learns how the batch effects alter image appearance (Fig. S4). Our hypothesis was that the modeling of mixed effects would improve classification performance over the base AEC. This architecture is described in Supplemental Materials 3.1.3, available online.

TABLE I
SPIRAL SIMULATION RESULTS WITH 10-FOLD CROSS-VALIDATION

	Simulation 1: cluster-specific radii		Simulation 2: cluster-specific radii with inversions		Simulation 3: Simulation 1 + confounded features	
Model	Mean acc. (%)	95% CI	Mean acc. (%)	95% CI	Mean acc. (%)	95% CI
Conventional DFNN	76.4	75.8 - 76.9	52.3	51.3 - 53.2	72.1	70.2 - 74.1
Cluster input DFNN	67.6	65.8 - 69.3	67.1	66.1 - 68.2	**76.4**	75.6 - 77.3
MLDG	62.8	60.9 - 64.7	50.2	49.9 - 50.5	65.8	64.9 - 66.7
DA-DFNN	75.3	72.5 - 78.0	49.5	48.8 - 50.2	66.3	63.6 - 69.0
MeNet	77.4	76.8 - 78.0	53.3	52.1 - 54.5	73.0	71.0 - 75.0
LMMNN	50.0	50.0 - 50.0	50.0	50.0 - 50.0	47.0	46.7 - 47.3
ARMED-DFNN	78.8	78.0 - 79.6	65.0	61.2 - 68.8	74.5	72.2 - 76.9
w/o Adv.	**79.3**	76.9 - 81.8	**69.9**	67.3 - 72.5	68.5	67.5 - 69.4
randomized Z	76.7	75.6 - 77.8	50.6	49.7 - 51.6	69.7	66.7 - 72.8

DFNN: dense feedforward neural network; MLDG: meta-learning domain generalization; DA: domain adversarial; Adv.: adversary; acc.: accuracy; CI: confidence interval.

The best results in each simulation are bolded.

In addition to evaluating the reconstruction error (MSE) and phenotype prediction performance (AUROC), we also measured how strongly each model's latent representations clustered by batch. We computed the Davies-Bouldin (DB) score, where lower values indicate stronger clustering [29], and the Calinksi-Harabasz (CH) score, where higher values indicate stronger clustering [30]. Consequently, we desire a higher DB score and lower CH score to achieve batch-invariant latent representations.

F. Compared Methods and Ablation Tests

In each application, we compared the proposed mixed effects model with the following approaches. First, we tested a conventional neural network where the cluster membership Z is disregarded and data is assumed to be *iid*. Second, for the DFNN and CNN, we tried the "cluster input" approach of treating the one-hot cluster membership Z as a categorical covariate, i.e. an additional model input. For the DFNN, Z was concatenated to X. For the CNN, Z was concatenated to flattened output of the last convolutional layer, before the dense hidden layer. We also created a "cluster input+" CNN where Z was expanded into a tensor and concatenated along the channel dimension before every convolutional layer. When evaluating on unseen clusters, we used the inferred Z from the Z-predictor. Third, we also compared to meta-learning domain generalization (MLDG) [17]. However, due to the high computational cost of second-order gradients in MLDG (training took 10 times longer than the conventional DFNN) and poor performance, we dropped the MLDG comparison for the other applications, after the spiral simulation application. Fourth, we tested a domain adversarial (DA) neural network, i.e. the fixed effect subnetwork by itself. Despite regularization to learn only fixed effects, it does not model any cluster-specific random effects. Finally, for the DFNN and CNN, we tested MeNet [21] and LMMNN [13]. For the autoencoder, only the proposed ARMED approach has a suitable adaptation.

Additionally, we performed two ablation tests of the proposed mixed effects approach. We first trained the ARMED models without the adversarial classifier ("w/o Adv.") to test the necessity of the generalization loss to learn fixed effects. Additionally, we evaluated the originally trained, complete ARMED model on held-out data with randomly-assigned cluster memberships ("randomized Z"). For data from seen clusters, this tested whether the model truly learned cluster-specific effects. For data from unseen clusters, this tested the impact of using the Z-predictor to infer cluster membership.

III. RESULTS

A. Spiral Classification Simulations

The classification accuracy of each model, with 10-fold cross-validation, is presented in Table I. In simulation 1 (random cluster-specific radii distributed around 1), the ARMED-DFNN outperformed all other models and had statistically significantly higher accuracy than the second-best model, MeNet (78.8% versus 77.4%, $p = 0.003$ in paired T-test). It was also uniquely able to learn appropriate cluster-specific decision boundaries that scaled in size with the cluster-specific spiral radii (Fig. 2). For example, cluster 1 (left column) has the smallest ground truth radius (green dashed line), and while cluster 2 (middle column) has the largest true radius, and the ARMED-DFNN uniquely learned this difference. In simulation 2 (greater inter-cluster variance, spiral labels inverted in half), only the cluster input DFNN, MeNet, and ARMED-DFNN achieved accuracy substantially higher than chance (50%), with 67.1%, 53.3%, and 65.0% respectively. The cluster input DFNN and ARMED-DFNN statistically significantly outperformed MeNet ($p \ll 0.001$), but did not differ significantly from each other at $p < 0.05$. In simulation 3 (confounded probe features added), the cluster input ranked first (76.4%), followed by the ARMED-DFNN (74.5%) and MeNet (73.0%). However, the ARMED-DFNN more effectively downweighted the 2 confounded features compared to the true features (T-statistic $=$ 12.631 and 18.173) compared to the cluster input DFNN (T-statistic $=$ 5.346 and 5.042) and MeNet (T-statistic $=$ 7.923 and 4.541) (Table II). The conventional and meta-learning models placed greater importance on the confounded than the true features.

In ablation tests, removing adversarial regularization non-significantly improved the accuracy of the ARMED-DFNN in

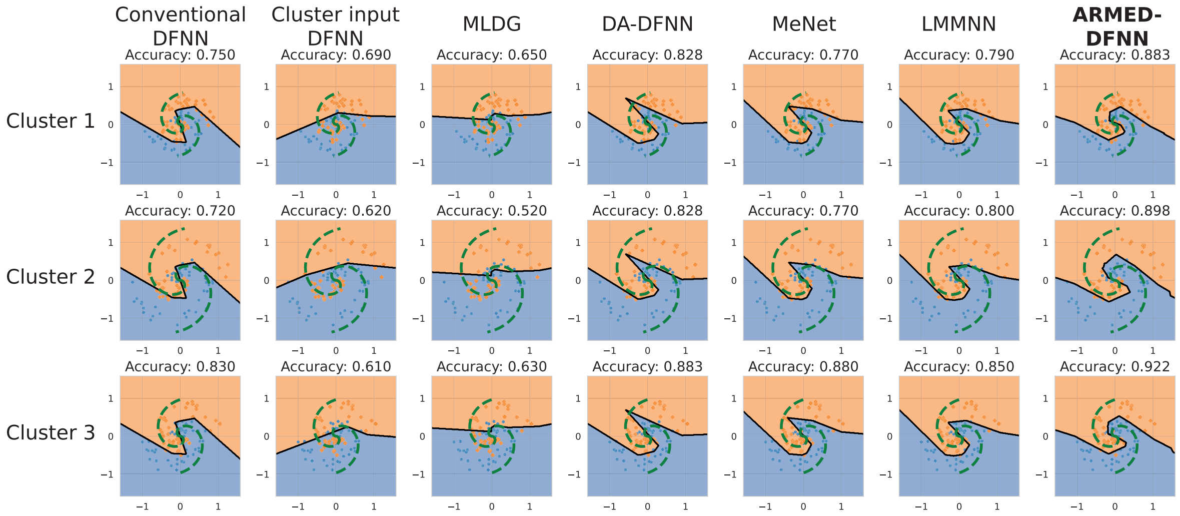

Fig. 2. Decision boundaries learned by each model in spiral simulation 1, where spiral radii varies across clusters as a random effect. Each row illustrates one of 3 representative clusters from 10 total simulated clusters. Each column contains the decision boundaries (black solid line) learned by one model. The green dashed line illustrates the true decision boundary, computed as the midpoint between the two spirals. Only the ARMED-DFNN was able to learn the appropriate cluster-specific decision boundaries.

TABLE II
SENSITIVITY TO CONFOUNDED PROBE FEATURES IN SPIRAL SIMULATION 3

	Probe x_3		Probe x_4	
Model	T-statistic	p-value	T-statistic	p-value
Conventional DFNN	-0.232	0.822	-0.861	0.411
Cluster input DFNN	5.346	<0.001	5.042	<0.001
MLDG	-16.573	<0.001	-14.535	<0.001
DA-DFNN	4.072	0.003	3.832	0.004
MeNet	7.923	<0.001	4.541	<0.001
LMMNN	8.369	<0.001	5.090	<0.001
ARMED-DFNN	**12.632**	<0.001	**18.173**	<0.001
w/o Adv.	1.390	0.198	0.343	0.740

Paired T-tests were computed to compare the feature importance of each confounded probe with the least important true feature. Positive and larger T-statistics are desired, indicating the model placed higher importance on the true feature than the confounded probe.

simulations 1 and 2, but decreased accuracy in simulation 3. It also worsened the separation of confounded and true features in simulation 3 (T $=$ 1.390 and 0.343). Using randomly assigned cluster memberships in Z uniformly decreased performance, confirming that the ARMED-DFNN learned necessary cluster-specific information.

B. *MCI Conversion Prediction*

The performance of each model in classifying pMCI versus sMCI, over 10×10 nested cross-validation folds, is compared in Table III. On study sites *seen* during training, the ARMED-DFNN outperformed all other models in AUROC, accuracy, and specificity (Table III, top). The AUROC of the ARMED-DFNN was statistically significantly higher than that of the second-best model, the conventional DFNN (0.926 versus 0.884, $p = 0.048$). On held-out study sites *unseen* during training, the ARMED-DFNN again outperformed all other models in AUROC, accuracy, and specificity (Table III, bottom). The AUROC of the ARMED-DFNN was statistically significantly higher than that of the second-best LMMNN (0.837 versus 0.811, $p \ll 10^{-3}$). The DA-DFNN performed the poorest on both seen and unseen sites, with AUROC of 0.811 and 0.723 respectively.

Removing the adversarial regularization of the ARMED-DFNN reduced AUROC (0.926 to 0.919) and accuracy (81.9% to 81.4%) on seen sites and accuracy (75.6% to 73.5%) and sensitivity (72.4% to 65.4%) on unseen sites. On seen sites, randomizing the site assignments reduced all metrics, including AUROC from 0.926 to 0.889. On unseen sites, using random instead of inferred site assignments also reduced all metrics including sensitivity from 72.4% to 69.8%.

We examined the feature importance ranking, based on the fixed effects subnetwork, and learned site-specific random slopes, based on the random effects subnetwork, of the ARMED-DFNN (Fig. 3). Demographic features including, race, ethnicity, and marital status had especially low inter-site variance. Cognitive scores such as the Clinical Dementia Rating Sum of Boxes (CDR-SB) and and Mini Mental State Exam (MMSE) had especially high inter-site variance. These results are further discussed in Section IV-C1. Feature importance rankings for all 6 DFNNs are presented in Fig. S6. We also examined the site-specific random intercepts of the ARMED-DFNN and found they correlated strongly with the percentage of pMCI subjects at each site (Pearson's $r = 0.860$, $p < 10^{-5}$), indicating the random intercepts captured the variability in class balance across sites, a major source of confounding effect.

When simulated confounded probe features were added, the ARMED-DFNN ranked these probes the lowest. The 10 highest ranked features for each model are shown in Fig. 4. The conventional DFNN, cluster input DFNN, LMMNN, and ARMED-DFNN without domain adversarial regularization all included 3 of the 5 confounded probes within the top 10 features. The DA-DFNN and MeNet included 1 confounded probe and

TABLE III
PREDICTION OF STABLE VERSUS PROGRESSIVE MILD COGNITIVE IMPAIRMENT

	AUROC		Balanced accuracy (%)		Sensitivity (%)		Specificity (%)	
Model	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean	95% CI
Seen sites								
Conventional DFNN	0.884	0.836 - 0.931	80.8	74.6 - 87.0	**81.2**	68.3 - 94.1	80.3	74.7 - 86.0
Cluster input DFNN	0.866	0.819 - 0.914	81.3	75.8 - 86.8	80.2	68.6 - 91.7	82.4	77.5 - 87.3
DA-DFNN	0.811	0.745 - 0.876	75.5	68.9 - 82.2	74.9	62.3 - 87.6	76.1	69.0 - 83.2
MeNet	0.830	0.780 - 0.880	75.5	68.3 - 82.7	73.7	59.0 - 88.4	77.3	71.7 - 82.9
LMMNN	0.860	0.824 - 0.896	79.4	72.2 - 86.6	73.9	59.7 - 88.1	84.9	81.6 - 88.1
ARMED-DFNN	**0.926**	0.901 - 0.951	**81.9**	77.7 - 86.1	76.5	67.6 - 85.3	**87.4**	84.5 - 90.2
w/o Adv.	0.919	0.891 - 0.946	81.4	76.8 - 86.1	74.5	64.6 - 84.4	**88.4**	85.4 - 91.4
randomized Z	0.889	0.862 - 0.916	79.1	73.9 - 84.2	73.9	64.0 - 83.9	84.2	80.2 - 88.2
Unseen sites								
Conventional DFNN	0.806	0.786 - 0.825	73.9	71.9 - 76.0	**76.2**	73.4 - 78.9	71.7	68.5 - 74.8
Cluster input DFNN	0.796	0.776 - 0.816	74.4	72.7 - 76.2	75.4	72.5 - 78.4	73.4	71.6 - 75.2
DA-DFNN	0.723	0.665 - 0.780	67.9	63.2 - 72.6	64.7	52.7 - 76.8	71.1	67.4 - 74.7
MeNet	0.750	0.693 - 0.807	70.2	65.6 - 74.9	66.0	57.7 - 74.4	74.5	69.8 - 79.1
LMMNN	0.811	0.805 - 0.817	74.6	73.6 - 75.7	71.1	68.1 - 74.2	78.1	76.9 - 79.3
ARMED-DFNN	**0.837**	0.833 - 0.842	**75.6**	74.1 - 77.1	72.4	67.6 - 77.1	78.8	76.6 - 80.9
w/o Adv.	**0.838**	0.827 - 0.848	73.5	72.5 - 74.5	65.4	62.9 - 67.8	**81.7**	80.7 - 83.3
randomized Z	0.830	0.822 - 0.837	74.6	73.3 - 75.9	69.8	65.0 - 74.5	79.5	77.0 - 82.0

DFNN: dense feedforward neural network; MLDG: meta-learning domain generalization; DA: domain adversarial; Adv.: adversary; AUROC: area under receiver operating characteristic curve; CI: confidence interval. Note: Cluster is inferred via our Z-predictor for the unseen sites for Cluster input DFNN model.

Confidence intervals were computed through 10×10-fold nested cross-validation. Sensitivity and specificity were computed at the Youden point. The best results for each metric are bolded.

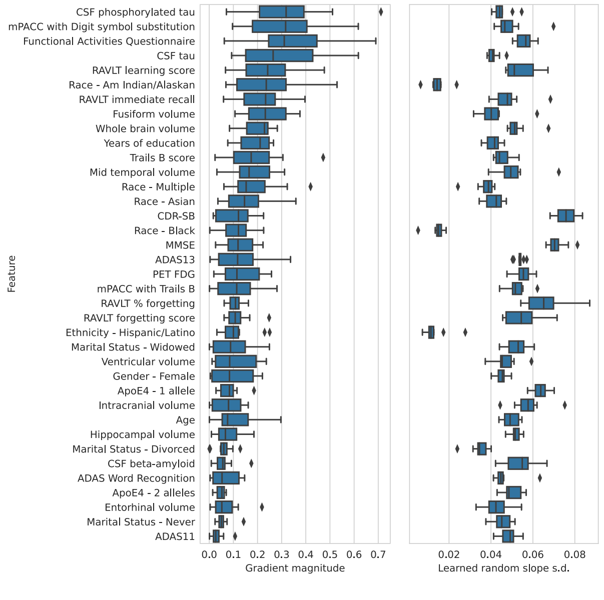

Fig. 3. Feature importance and random slope variance for the ARMED-DFNN predictor of stable versus progressive mild cognitive impairment. *left*) Features are ranked by descending median feature importance (gradient magnitude) across 10 cross-validation folds, measured from the fixed effects subnetwork. *right*) The inter-site variance of each feature's random slopes. See Supplemental Section 3.3.1, available online for abbreviations.

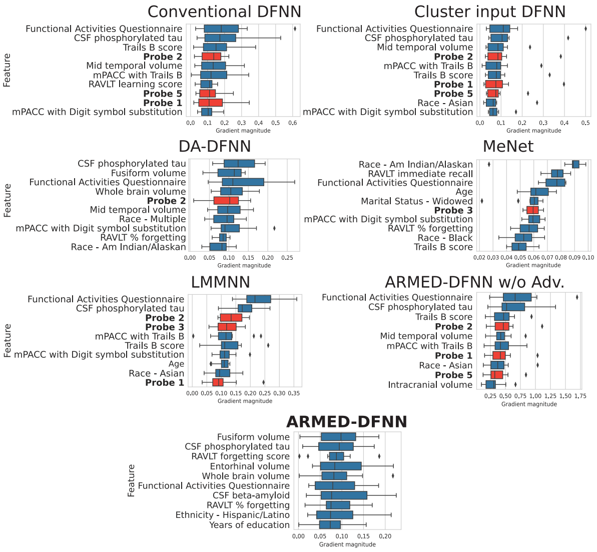

Fig. 4. MCI conversion prediction with 5 added confounded probes (**bolded** label, red bar). For each DFNN, the top 10 features are shown, ranked by median feature importance (gradient magnitude) across 10 cross-validation folds.

the full ARMED-DFNN included none in the top 10 features. Paired sign tests indicate that the ARMED-DFNN ranked the confounded probes significantly lower than any other model, e.g. $p = 0.031$ when compared to the second-best models, DA-DFNN and MeNet.

C. AD Diagnosis

The cross-validated performance of each model in classifying brain MRIs as CN versus AD is presented in Table IV. On study sites seen during training, LMMNN showed the highest AUROC, followed by MeNet, cluster input+, and the ARMED-CNN (Table IV4, top). Neither LMMNN, MeNet, nor cluster input+ significantly outperformed ARMED-CNN (paired T-test, $p = 0.064, 0.210, 0.255$, respectively). On unseen sites (Table IV, bottom), the ARMED-CNN performed second best after

TABLE IV
ALZHEIMER'S DISEASE DIAGNOSIS FROM MRI

	AUROC		Balanced accuracy (%)		Sensitivity (%)		Specificity (%)	
Model	Mean	95% CI	Mean	95% CI	Mean	95% CI	Mean	95% CI
				Seen sites				
Conventional CNN	0.703	0.621 - 0.785	69.7	63.7 - 75.7	69.3	56.7 - 81.8	70.1	58.8 - 81.5
Cluster input CNN	0.654	0.585 - 0.722	72.6	68.2 - 76.9	76.0	67.4 - 84.7	69.1	56.5 - 81.8
Cluster input+ CNN	0.901	0.871 - 0.930	89.1	86.0 - 92.1	89.1	82.6 - 95.6	89.0	84.4 - 93.7
DA-CNN	0.823	0.730 - 0.917	79.9	74.2 - 85.6	77.2	61.4 - 93.0	82.6	76.8 - 88.5
MeNet	0.923	0.894 - 0.952	89.6	86.7 - 92.5.	87.7	82.0 - 93.4	91.5	88.9 - 94.2
LMMNN	**0.938**	0.917 - 0.959	**90.4**	87.7 - 93.1	88.9	82.4 - 95.4	**91.9**	88.4 - 95.3
ARMED-CNN	0.900	0.861 - 0.939	88.7	83.8 - 91.6	**91.8**	87.0 - 96.7	83.6	75.2 - 92.0
w/o Adv.	0.816	0.729 - 0.903	79.7	73.3 - 86.2	**91.6**	86.1 - 97.2	67.8	52.6 - 83.1
randomized Z	0.585	0.506 - 0.664	63.3	58.3 - 68.2	65.5	48.3 - 82.7	61.0	47.2 - 74.8
				Unseen sites				
Conventional CNN	0.603	0.531 - 0.675	59.5	55.4 - 63.5	56.8	41.5 - 72.1	62.1	46.8 - 77.5
Cluster input CNN	0.587	0.520 - 0.653	58.3	54.4 - 62.2	57.6	42.3 - 72.9	59.0	42.9 - 75.2
Cluster input+ CNN	0.538	0.481 - 0.594	55.0	51.9 - 58.1	54.1	33.7 - 74.6	55.9	35.7 - 76.1
DA-CNN	**0.652**	0.614 - 0.690	**62.5**	59.7 - 65.3	**71.8**	66.7 - 77.0	53.1	48.9 - 57.4
MeNet	0.517	0.463 - 0.571	53.9	51.6 - 56.3	64.9	45.7 - 84.1	43.0	23.6 - 62.3
LMMNN	0.534	0.491 - 0.576	54.2	51.8 - 56.5	45.3	27.1 - 63.4	**63.1**	45.7 - 80.6
ARMED-CNN	0.645	0.606 - 0.684	61.2	58.6 - 63.9	65.9	60.3 - 71.6	56.6	50.8 - 62.3
w/o Adv.	**0.655**	0.608 - 0.701	**62.2**	58.9 - 65.6	68.6	61.6 - 75.6	55.9	46.6 - 65.1
randomized Z	0.551	0.526 - 0.576	54.6	53.0 - 56.2	47.5	33.9 - 61.1	61.8	47.4 - 76.1

CNN: convolutional neural network; DA: domain adversarial; Adv.: adversary; AUROC: area under ROC curve; CI: confidence interval. Note: Cluster is inferred via our Z-predictor for the unseen sites for Cluster input(+) CNN models.

Metrics were computed through 10 Monte Carlo cross-validation replicates. Sensitivity and specificity were computed at the Youden point. The best results for each metric are bolded.

the DA-CNN in AUROC, accuracy, and sensitivity. MeNet, LMMNN, and cluster input+ had the lowest AUROC on the unseen sites, indicating poor generalization. Without adversarial regularization, the performance of the ARMED-CNN increased slightly on unseen sites (mean AUROC 0.645 to 0.655) but decreased on seen sites (mean AUROC 0.900 to 0.816). Randomizing the site membership for seen sites drastically reduced all metrics, including mean AUROC from 0.900 to 0.585. On unseen sites, randomizing instead of inferring site membership reduced mean AUROC from 0.645 to 0.551.

Gradient-weighted Class Activation Mapping (Grad-CAM) visualizations from each model revealed differences in the features learned (Fig. 5) [31]. The conventional, cluster input, MeNet, and LMMNN CNNs attributed more weight to regions in the edges of each image, near the periphery of the brain. However, the DA-CNN emphasized medial brain areas, including the hippocampi and surrounding parahippocampal gyri. For the ARMED-CNN, we produced Grad-CAMs using the fixed effects subnetwork, which contains the learned cluster-invariant features. Like the DA-CNN, the ARMED-CNN also emphasized medial brain areas but gave additional weight to the superior regions including the lateral ventricles. Furthermore, we created separate Grad-CAMs to visualize the distinct site-specific features learned by the ARMED-CNN random effects subnetwork (Fig. S7), which involved the image periphery for some sites and more medial areas for others.

D. Cell Image Compression and Classification

The performance of each AEC model in compressing and classifying melanoma live-cell images is presented in Table V. For computational efficiency, the pre-trained and frozen DA-AEC was reused as the fixed effects subnetwork of the ARMED-AEC. For the ablation test without adversarial regularization ("w/o Adv."), the pre-trained conventional AEC was reused as the fixed effects subnetwork. Confidence intervals were computed using DeLong's method [32]. On *seen* batches, (Table V, first column group) the ARMED-AEC had the highest performance in classifying metastatic efficiency (AUROC 0.869), statistically significantly outperforming the second-best model, the conventional AEC ($p < 0.001$), and it had the lowest reconstruction error (MSE 0.0012). On *unseen* batches (Table V, second column group), the ARMED-AEC again showed the best classification performance (AUROC 0.789). This classification performance was statistically significantly higher than the second-best model, the conventional AEC ($p < 0.001$). All models had similar reconstruction error (MSE 0.0024) on unseen batches. Examining each AEC's latent representations, the DA-AEC and ARMED-AEC (using the DA-AEC as its fixed effects subnetwork) exhibited much less batch effect contamination. Compared to the conventional AEC, the DB score improved from 8.885 to 43.009 (484% relative increase) and the CH score improved from 545.9 to 20.4 (96% relative decrease).

In the ablation tests, removing the domain adversarial regularization of the fixed effects subnetwork in the ARMED-AEC

TABLE V
Melanoma Live Cell Image Compression and Classification, and Batch Effect Contamination of Latent Representations

	Seen batches		Unseen batches		Latent batch clustering	
Model	MSE	AUROC (95% CI)	MSE	AUROC (95% CI)	DB	CH
Conventional AEC	0.0019	0.817 (0.812 - 0.822)	0.0024	0.773 (0.764 - 0.781)	8.885	545.9
DA-AEC	0.0018	0.777 (0.771 - 0.783)	0.0024	0.759 (0.750 - 0.768)	**43.009**	**20.4**
ARMED-AEC	**0.0012**	0.869 (0.865 - 0.874)	0.0024	**0.789** (0.781 - 0.798)	**43.009**	**20.4**
w/o Adv.	**0.0012**	**0.876** (0.872 - 0.881)	0.0024	**0.791** (0.782 - 0.799)	8.885	545.9
randomized Z	0.0018	0.732 (0.726 - 0.738)	0.0024	0.712 (0.702 - 0.721)		

AEC: autoencoder-classifier; DA: domain adversarial; Adv.: adversary; MSE: mean squared error between original and reconstructed images; AUROC: area under receiver operating characteristic curve for phenotype classification; CI: confidence interval; DB: Davies-Bouldin score, lower values indicate stronger clustering; CH: Calinski-Harabasz score, higher values indicate stronger clustering

Confidence intervals were computed with DeLong's method. The best results for each metric are bolded.

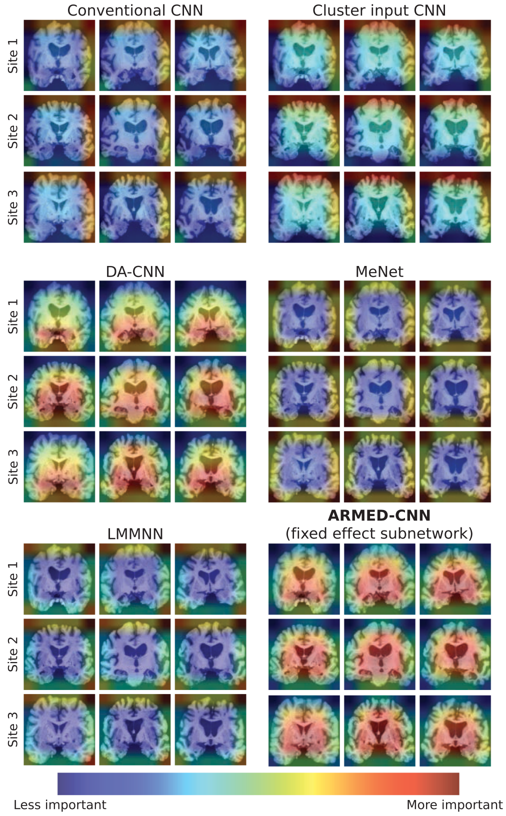

Fig. 5. Grad-CAM visualizations indicating important image regions for classifying Alzheimer's Disease versus cognitively normal individuals. Each row contains examples from one of three representative study sites.

(using the conventional AEC as the fixed effects subnetwork) slightly increased classification AUROC on *seen* batches (0.876 versus 0.869) and on unseen batches (0.791 versus 0.789). However, this came at the expense of greater batch contamination of the latent representations (DB score 8.885 and CH score 545.9). When cluster assignments were randomized instead of using the true cluster assignments on seen batches, reconstruction MSE worsened from 0.0012 to 0.0018 and classification AUROC decreased from 0.869 to 0.732. On *unseen* batches, randomized instead of Z-predictor-inferred cluster assignments reduced classification AUROC from 0.789 to 0.712.

To visualize the random effects learned by the ARMED-AEC, we generated image reconstructions from the random effects subnetwork with various learned batch-specific effects applied (Fig. 6). These simulate the appearance of an image if it had been acquired within different batches. We compared these with the image reconstructions from the fixed effects subnetwork, where batch effects have been removed. Some batches showed stronger specular highlights (e.g., batches 2 and 5), while others had greater contrast in the cell periphery (e.g., batches 1 and 3).

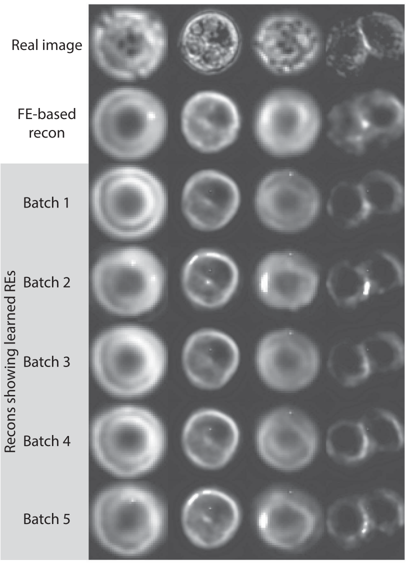

Fig. 6. Reconstructed melanoma cell images from the ARMED-AEC. The first row contains the real image, the second row contains the fixed effects-based reconstructions, and remaining rows show random effects-based reconstructions using different learned random effects.

IV. Discussion

A. General Observations

Our experiments across four applications illustrate the three critical contributions of ARMED. First, we demonstrated that the fixed effects subnetwork of ARMED models assigns feature importance more appropriately than the compared models. In the spiral simulations, the ARMED-DFNN most strongly separated the true and confounded features by feature importance. The conventional and MLDG models erroneously placed greater importance on the confounded probes than the true features, while the cluster input, DA-DFNN, MeNet, and LMMNN models downweighted the confounded probes to a lesser degree than the ARMED-DFNN. In MCI conversion prediction with simulated confounded probes, the ARMED-DFNN ranked the probes statistically significantly lower than any other model, including the DA-DFNN and MeNet. In contrast, the conventional, cluster input, and LMMNN models were most sensitive to the probes. In AD diagnosis, Grad-CAM visualizations showed that the ARMED-CNN highlighted more biologically plausible brain regions than the conventional, cluster input, MeNet, and LMMNN CNN's, which is further discussed in Section IV-C2.

Second, we demonstrated the ability of ARMED to visualize random effects learned by the random effects subnetwork. In MCI conversion prediction, we quantified the learned inter-site variance of the random slopes for each feature. This allowed us to identify which features are most contaminated by site effects, and we discuss these below (Section IV-C1). In AD diagnosis, we visualized site-specific differences in Grad-CAMs. Finally, in the cell imaging application, we generated image reconstructions showing the impact of learned batch effects.

Third, ARMED typically outperforms the compared non-mixed effects methods and outperforms or matches the other mixed effects methods. In the spiral simulations, the ARMED-DFNN had either the best or second-best accuracy, while being more discriminative between true and confounded features and learning the most cluster-appropriate decision boundaries. In the MCI conversion application, the ARMED-DFNN outperformed all other methods on both data from seen and unseen sites. In AD diagnosis, the ARMED-CNN performed similarly to MeNet and LMMNN methods on seen sites and competed favorably with the DA-CNN on unseen sites. Meanwhile, MeNet and LMMNN generalized poorly to unseen sites. In the cell imaging application, the ARMED-AEC had the best reconstruction error on data from seen batches, the best metastatic efficiency classification on both seen and unseen batches, and substantially reduced batch effects in its latent representations compared to the conventional AEC. In ablation tests, we found that ARMED models without DA often performed similarly to or non-significantly better than the full model with DA, but their fixed effects subnetworks were more sensitive to confounded probe features. Therefore, we recommend always using the full ARMED model with DA, as any small performance increase comes at the cost of confound susceptibility. We also found that randomizing cluster assignment reduced performance on seen clusters, confirming that the ARMED models had learned cluster-specific information in the random effects subnetworks. Similarly, performance on unseen clusters decreased when using randomized cluster assignment instead of using the Z-predictor to infer cluster membership. This indicates that the Z-predictor is needed to fully exploit the learned random effects when predicting on data from unseen clusters.

Though we have focused on biomedical data in this work, we anticipate that our approach will be of use to any case where data is non-*iid* and subject to random effects. Given its flexible and modular nature, the ARMED framework should apply readily to other architecture types besides the three demonstrated here.

B. Comparison to Prior Work

A common approach to handling clustered data is to include the cluster membership, which is an unordered categorical variable, as additional one-hot encoded covariates in X [12]. This approach is unable to disentangle the cluster-specific random effects and cluster-independent fixed effects, and we found it was more sensitive to simulated confounded probes than ARMED models. We also found inferior performance versus ARMED, likely due to the high cardinality of the added features which can lead to overfitting [13], [33]. For example, the MCI conversion application had 20 sites and 37 input features, meaning that to add cluster membership to X would increase the width of X by 35%. ARMED is better suited to handling this high-cardinality information by modeling clustering as a random effect, which imposes a normal distribution prior.

A more recent approach to handling differences across clusters is domain adversarial learning. We showed that DA does improve generalization to data from unseen clusters. However, ARMED improves upon DA, adding a random effects subnetwork to capture the cluster-specific information that DA discards, which results in better performance on clusters *seen* during training. Using the Z-predictor, this cluster-specific information can also be used when predicting on data from unseen clusters, allowing ARMED to outperform DA on *unseen* clusters as well.

This work remedies key weaknesses in previous approaches to incorporate mixed effects into deep learning. We described specific random effects architectures for random intercepts, linear slopes, and/or nonlinear slopes. This allows greater flexibility than DeepGLMM and LMMNN, which only learn random intercepts, and MeNet, which only learns nonlinear random slopes [13], [21], [22]. Another key improvement was adversarial regularization of the fixed effects subnetwork to learn generalizable, cluster-agnostic information. In our experiments with simulated confounders, this allowed ARMED models to appropriately upweight nonconfounded features and downweight confounded features, while MeNet and LMMNN, lacking adversarial regularization, were susceptible to the spurious confounded features. Next, we demonstrated interpretation and visualization of the learned random effects, which was not explored in these previous works. Finally, we evaluated ARMED models on data from clusters unseen during training and provided a method to infer cluster membership and apply learned random effects on this data. The previous works lacked such a method, meaning that the learned random effects cannot be utilized on new data. This is a major limitation for practical applications, where a deployed model may need to be applied to data from a new cluster, such as a new clinical site or patient.

C. Application-Specific Discussions

1) MCI Conversion Prediction: The ARMED-DFNN quantifies the inter-site variance of the learned random slope for each feature (Fig. 3). We found that demographic features such as race and ethnicity had the lowest inter-site variance, which in unsurprising as the association between these features and

MCI conversion should not be sensitive to measurement differences across sites. Certain cognitive measurements, however, had distinctly high inter-site variance. The CDR-SB score had the highest variance, which concurs with a previous report that CDR-SB has suboptimal inter-rater reliability in early dementia patients, such as those with MCI [34]. MMSE had the second highest variance in our ARMED-DFNN, again agreeing with previous findings of low inter-rater reliability [35].

Though we intentionally held out a large portion of the ADNI dataset to evaluate our models on unseen sites, our ARMED-DFNN performed similarly to or better than several published results on predicting 24-month MCI conversion in ADNI using deep learning. Lee et al. achieved 80% accuracy compared to the 81.9% of our ARMED-DFNN [36]. Shi et al. and Lian et al. achieved AUROC of 0.816 and 0.793, respectively, compared to our 0.926 [37], [38]. Note that neither of these studies held out entire study sites for evaluation, and our AUROC on *unseen* sites (0.837) still exceeded their results on *seen* sites.

2) AD Diagnosis: The Grad-CAMs of the DA-CNN and ARMED-CNN appropriately emphasized the importance of medial brain regions including the hippocampus and surrounding medial temporal lobe, which are involved in AD-related brain atrophy (Fig. 5) [39], [40], [41]. The ARMED-CNN Grad-CAMs also indicated the importance of the lateral ventricles, where enlargement has been connected to AD [40], [42]. The incorporation of these additional structures likely contributed to the better performance of the ARMED-CNN (AUROC 0.900) versus the DA-CNN (AUROC 0.823). Meanwhile, the conventional, cluster input, MeNet, and LMMNN models relied highly on likely spurious features in the image periphery. Such features appear to be related to site effects on imaging, since the random effects of the ARMED-CNN affect similar peripheral areas (Fig. S7).

The performance of our ARMED-CNN compares favorably to previous models using 2D MRI to diagnose AD in the ADNI dataset. We achieved 88.7% accuracy, while Kang et al. report 90.4% and Ebrahimi et al. report 87.5% [43], [44]. However, we trained on a fraction of the total ADNI data that these reports used, holding out the rest for evaluation of models on unseen sites. Consequently, our work focuses on comparisons across architectures, not with previous studies.

3) Cell Image Compression and Classification: We compare our ARMED-AEC results to the previous analysis published by Zaritsky et al. [6]. While they discussed the batch effect present in the latent representations produced by their autoencoder, their methods did not explicitly suppress this batch effect. In contrast, our proposed ARMED-AEC reduced the batch effect in the latent representations by 484% based on the DB score, compared to an unmodified AEC. We also improved classification AUROC to 0.876 compared to their reported 0.723, though this may be partially due to the direct incorporation of the phenotype classifier into the autoencoder, while Zaritsky et al. trained their classifier separately from their autoencoder.

D. Limitations

Mixed effects models generally require the presence of several clusters to accurately estimate the random effect distributions; with <4 clusters, LME models provide less of an advantage over generalized linear regression [8], [11]. Consequently, we suggest some caution when using our method for data with fewer than 4 clusters. Additionally, our method applies to datasets with a single level of random effects, but there are often cases with multiple levels of random effects, such as when multiple observations are collected per subject who are then clustered by study site. We plan to extend our methodology to such multi-level cases in future work. Finally, a practical limitation of ARMED is the additional complexity, which may increase training time by approximately 1.5-2x. However, we note that other methods have even greater computation cost, such as meta-learning domain generalization (MLDG) which uses second-order optimization and MeNet and LMMNN which involve expensive matrix inversions [13], [17], [21].

V. Conclusion

Our proposed approach uses mixed effects techniques from traditional statistics to improve the interpretability, reliability, and performance of deep learning models on non-*iid* data. ARMED models separately learn random and fixed effects in distinct subnetworks, with the fixed effects subnetwork more appropriately assigning feature importance with resilience to confounding effects, helping to avoid Type I and Type II errors. In biomedical applications, this allows better hypothesis formation and prevents waste of resources in following up confounded results. Meanwhile, the random effects subnetwork allows users to understand the cluster effects in their data, which can inform future research. For example, clinical study organizers could prioritize measurements with less inter-site variance in future studies. Besides these benefits, ARMED increases predictive performance on clustered data, including better generalization to clusters unseen during training. Given these advantages demonstrated across multiple model architectures and applications, we broadly recommend the ARMED framework to deep learning practitioners dealing with non-*iid* data. We make our code available at `tinyurl.com/ARMEDCode`.

Acknowledgments

This study was supported by NIH grant R01GM144486 and the Lyda Hill Foundation. Data used in this paper were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset. The investigators within the ADNI did not participate in analysis or writing of this study. A complete list of ADNI investigators can be found online.

References

[1] D. J. Connor and M. N. Sabbagh, "Administration and scoring variance on the ADAS-Cog," *J. Alzheimer's Dis.*, vol. 15, pp. 461–464, 2008.

[2] E. Kozora et al., "Effects of examiner error on neuropsychological test results in a multi-site study," *Clin. Neuropsychologist*, vol. 22, pp. 977–988, 2008.

[3] K. Schafer, S. de Santi, and L. S. Schneider, "Errors in ADAS-cog administration and scoring may undermine clinical trials results," *Curr. Alzheimer Res.*, vol. 8, pp. 373–376, 2011.

[4] F. Kruggel, J. Turner, and L. T. Muftuler, "Impact of scanner hardware and imaging protocol on image quality and compartment volume precision in the ADNI cohort," *NeuroImage*, vol. 49, pp. 2123–2133, 2010.

[5] C. Wachinger et al., “Quantifying confounding bias in neuroimaging datasets with causal inference,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2019, pp. 484–492.

[6] A. Zaritsky et al., “Interpretable deep learning uncovers cellular properties in label-free live cell images that are predictive of highly metastatic melanoma,” *Cell Syst.*, vol. 12, pp. 733–747.e6, 2021.

[7] A. Franks, E. Airoldi, and N. Slavov, “Post-transcriptional regulation across human tissues,” *PLoS Comput. Biol.*, vol. 13, 2017, Art. no. e1005535.

[8] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge, UK: Cambridge Univ. Press, 2007.

[9] G. B. Holt, “Potential Simpson’s paradox in multicenter study of intraperitoneal chemotherapy for ovarian cancer,” *J. Clin. Oncol.*, vol. 34, 2016, Art. no. 1016.

[10] C. H. Wagner, “Simpson’s paradox in real life,” *Amer. Statistician*, vol. 36, pp. 46–48, 1982.

[11] X. A. Harrison et al., “A brief introduction to mixed effects modelling and multi-model inference in ecology,” *PeerJ*, vol. 6, 2018, Art. no. e4794.

[12] J. T. Hancock and T. M. Khoshgoftaar, “Survey on categorical data for neural networks,” *J. Big Data*, vol. 7, pp. 1–41, 2020.

[13] G. Simchoni and S. Rosset, “Using random effects to account for high-cardinality categorical features and repeated measures in deep neural networks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 25111–25122.

[14] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[15] Y. Ganin et al., “Domain-adversarial training of neural networks,” *J. Mach. Learn. Res.*, vol. 17, pp. 2096–2030, 2016.

[16] T.-Y. Liu et al., “Bridging the generalization gap: Training robust models on confounded biological data,” 2018, *arXiv:1812.04778*.

[17] D. Li et al., “Learning to generalize: Meta-learning for domain generalization,” in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2018, pp. 3490–3497.

[18] Q. Liu, Q. Dou, and P.-A. Heng, “Shape-aware meta-learning for generalizing prostate MRI segmentation to unseen domains,” in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2020, pp. 475–485.

[19] E. Tzeng et al., “Adversarial discriminative domain adaptation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2962–2971.

[20] K. Kamnitsas et al., “Unsupervised domain adaptation in brain lesion segmentation with adversarial networks,” in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 597–609.

[21] Y. Xiong, H. J. Kim, and V. Singh, “Mixed effects neural networks (MeNets) with applications to gaze estimation,” in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7743–7752.

[22] M.-N. Tran et al., “Bayesian Deep Net GLM and GLMM,” *J. Comput. Graphical Statist.*, vol. 29, pp. 97–113, 2020.

[23] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2013, *arXiv:1312.6114*.

[24] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *J. Amer. Statist. Assoc.*, vol. 112, pp. 859–877, 2017.

[25] K. J. Lang and M. J. Witbrock, “Learning to tell two spirals apart,” *The 1988 Connectionist Models Summer Sch.*, pp. 52–59, 1989.

[26] Y. Dimopoulos, P. Bourret, and S. Lek, “Use of some sensitivity criteria for choosing networks with good generalization ability,” *Neural Process. Lett.*, vol. 2, pp. 1–4, 1995.

[27] J. D. Olden, M. K. Joy, and R. G. Death, “An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data,” *Ecological Modelling*, vol. 178, pp. 389–397, 2004.

[28] E. Thibeau-Sutre, B. Couvy-Duchesne, D. Dormont, O. Colliot, and N. Burgos, “MRI field strength predicts Alzheimer’s disease: A case example of bias in the ADNI data set,” in *Proc. IEEE 19th Int. Symp. Biomed. Imag.*, 2022, pp. 1–4.

[29] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[30] T. Calinski and J. Harabasz, “A dendrite method for cluster analysis,” *Commun. Statist. - Theory Methods*, vol. 3, pp. 1–27, 1974.

[31] R. R. Selvaraju et al., “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” *Int. J. Comput. Vis.*, vol. 128, pp. 336–359, 2020.

[32] E. R. DeLong, D. M. DeLong, and D. L. Clarke-Pearson, “Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach,” *Biometrics*, vol. 44, 1988, Art. no. 837.

[33] A. Rao, J. M. Monteiro, and J. Mourao-Miranda, “Predictive modelling using NeuroImaging data in the presence of confounds,” *NeuroImage*, vol. 150, pp. 23–49, 2017.

[34] K. Rockwood et al., “Interrater reliability of the clinical dementia rating in a multicenter trial,” *J. Amer. Geriatr. Soc.*, vol. 48, pp. 558–559, 2000.

[35] P. Bowie, T. Branton, and J. Holmes, “Should the Mini mental state examination be used to monitor dementia treatments?,” *Lancet*, vol. 354, pp. 1527–1528, 1999.

[36] G. Lee et al., “Predicting Alzheimer’s disease progression using multimodal deep learning approach,” *Sci. Rep.*, vol. 9, 2019, Art. no. 1952.

[37] H. Shi et al., “Early diagnosis of Alzheimer’s disease on ADNI data using novel longitudinal score based on functional principal component analysis,” *J. Med. Imag.*, vol. 8, 2021, Art. no. 024502.

[38] C. Lian, M. Liu, Y. Pan, and D. Shen, “Attention-guided hybrid network for dementia diagnosis with structural MR images,” *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 1992–2003, Apr. 2022.

[39] J. M. Schott et al., “Measuring atrophy in Alzheimer disease: A serial MRI study over 6 and 12 months,” *Neurology*, vol. 65, pp. 119–124, 2005.

[40] L. Ferrarini et al., “Shape differences of the brain ventricles in Alzheimer’s disease,” *NeuroImage*, vol. 32, pp. 1060–1069, 2006.

[41] B. Dubois et al., “Advancing research diagnostic criteria for Alzheimer’s disease: The IWG-2 criteria,” *The Lancet Neurol.*, vol. 13, pp. 614–629, 2014.

[42] L. G. Apostolova et al., “Hippocampal atrophy and ventricular enlargement in normal aging, mild cognitive impairment (MCI), and Alzheimer disease,” *Alzheimer Dis. Assoc. Disord.*, vol. 26, pp. 17–27, 2012.

[43] W. Kang et al., “Multi-model and multi-slice ensemble learning architecture based on 2D convolutional neural networks for Alzheimer’s disease diagnosis,” *Comput. Biol. Med.*, vol. 136, 2021, Art. no. 104678.

[44] A. Ebrahimi and S. Luo, “Convolutional neural networks for Alzheimer’s disease detection on MRI images,” *J. Med. Imag.*, vol. 8, 2021, Art. no. 024503.

Kevin P. Nguyen received the BS degree in biomedical engineering from Yale University. He is currently enrolled in the Medical Scientist Training Program with UT Southwestern, working toward both the MD and PhD degrees in biomedical engineering. His interests include the development of deep learning techniques to improve interpretability and performance on medical applications, especially in neuroradiology.

Alex H. Treacher received the BS degree in physics from the University of North Texas in 2014. He is currently working toward the PhD degree with the Bioinformatics Department, UT Southwestern. His interests include developing efficient approaches for the training of machine learning models and their applications for improving healthcare.

Albert A. Montillo received the BS and MS degrees from Rensselaer and the PhD degree in computer science and medical imaging from the University of Pennsylvania. He is an Assistant Professor with the Lyda Hill Department of Bioinformatics at UT Southwestern and head of the Deep Learning for Precision Health Lab. He develops the theory of machine learning for improved interpretability and generalization especially for prognostics from multimodal, multi-omic data of neurological disorders and improved disease mechanism insights.