**Highlights**

- Adversarial attacks can confidently fool ensemble networks in ECG analysis.
- Diversifying the extracted features between networks is crucial for robustness.
- Feature decorrelation and Fourier partitioning diversify network features.

**Authors**

Christopher Wiedeman, Ge Wang

**Correspondence**

wangg6@rpi.edu

**In brief**

In patient-critical applications like electrocardiogram (ECG) analysis, it is essential for AI to identify instances where it cannot operate effectively. Ensemble methods, which employ multiple networks, can estimate prediction uncertainty but are still deceived by targeted input noise known as adversarial attacks. The authors propose two methods—feature decorrelation and Fourier partitioning—that diversify individual networks, improving robustness against adversarial attacks in ECG analysis. Both methods are scalable and transferable, offering a promising approach to improve robustness across various tasks.
Decorrelative network architecture for robust electrocardiogram classification

Christopher Wiedeman¹ and Ge Wang²,3,*
¹Department of Electrical and Computer Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
²Department of Biomedical Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
³Lead contact
*Correspondence: wangg6@rpi.edu
https://doi.org/10.1016/j.patter.2024.101116

SUMMARY

To achieve adequate trust in patient-critical medical tasks, artificial intelligence must be able to recognize instances where they cannot operate confidently. Ensemble methods are deployed to estimate uncertainty, but models in an ensemble often share the same vulnerabilities to adversarial attacks. We propose an ensemble approach based on feature decorrelation and Fourier partitioning for teaching networks diverse features, reducing the chance of perturbation-based fooling. We test our approach against white-box attacks in single- and multi-channel electrocardiogram classification and adapt adversarial training and DVERGE into an ensemble framework for comparison. Our results indicate that the combination of decorrelation and Fourier partitioning maintains performance on unperturbed data while demonstrating superior uncertainty estimation on projected gradient descent and smooth adversarial attacks of various magnitudes. Furthermore, our approach does not require expensive optimization with adversarial samples during training. These methods can be applied to other tasks for more robust models.

INTRODUCTION

The exponential increase in high-dimensional patient datasets and constant demand for personalized healthcare justify the urgent need for artificial intelligence (AI) in medicine. As an excellent example, electrocardiograms (ECGs) are commonly used for inpatient monitoring of cardiac conditions and are now available in smart or implantable devices. While the proper application of continuous ECG monitoring requires further clinical investigation, large-scale collection and analysis of ECGs, in either inpatient or outpatient populations, has the potential to improve healthcare by monitoring for signs of heart problems or alerting medical services to emergency situations. Deeper analysis of numerous samples is necessary to extract more healthcare-relevant information hidden in these signals. Big data can be leveraged in this instance, but it is infeasible for human clinicians to individually analyze all these recordings, making AI a natural solution to this problem.¹–³

THE BIGGER PICTURE

The rising demand for medical services has accelerated a healthcare shortage across multiple countries. Artificial intelligence (AI) can potentially help ease this shortage by automating data processing and analysis in clinical workflows. However, current models lack the transparency and robustness required to be trusted with patient-critical tasks. Adversarial attacks are targeted input perturbations that can cause errors in a model’s predictions. These attacks are concerning, having been demonstrated in contexts like electrocardiogram analysis, and could undermine model reliability and facilitate medical fraud. Current methods for resolving this weakness, such as adversarial training, do not scale well to high-dimensional inputs, such as most medical signals.

One alternative is to detect unstable inputs using an ensemble of networks, allowing uncertain cases to be reviewed by human experts. However, adversarial attacks can often fool multiple networks simultaneously, as these models often share the same vulnerabilities. We propose two methods for diversifying these weaknesses among an ensemble. Our electrocardiogram classification experiments show that ensembles trained using these methods are more resilient to adversarial attacks, exhibiting higher uncertainty under unstable inputs while maintaining performance on unperturbed samples.
For this purpose, many researchers have applied deep learning to ECG classification. The 2017 PhysioNet Challenge is a milestone in this field, where deep neural networks (DNNs) were trained to classify atrial fibrillation (AFib) from single-lead ECGs.5 Similarly, the 2018 China Physiological Signal Challenge (CPSC 2018) observed classification of several rhythm abnormalities from 12-lead ECGs.6 Top-scoring models can often achieve high classification accuracies on test data, but their interpretability and robustness are major concerns.6

Robustness in machine learning is a persistent topic, yet it lacks a unified definition. Broadly, it can be defined as the ability of a model to adapt and perform under changing conditions, but there is no consensus on exactly how these conditions should be defined. For example, a review from Drenkow et al. proposes a unified definition of robustness in computer vision as a model’s performance on images where “environmental, sensor, or rendering conditions are low-probability” (e.g., unusually poor lighting conditions).7 Taori et al. focus on computer vision robustness in the context of natural distribution shifts caused by unanticipated changes in the scene.8 Another line of work defines robustness with deep belief networks as the ability of the model to accurately approximate a variety of underlying functions or control laws.9,10 This work specifically studies robustness against adversarial attacks, which have been demonstrated both in machine learning broadly and in specific healthcare tasks.

Adversarial attacks background
Adversarial attacks are small input perturbations that do not change the semantic content yet cause massive errors in a network output, for example, imperceptible noise patterns that, when added to an image, cause a model to misclassify the image.11 Given a target model and input, projected gradient descent (PGD) is the most common algorithm for finding adversarial perturbations under an $\ell_\infty$ bound.12,13 Various other algorithms exist, including the use of generative adversarial ensembles.14–16 Impressively, universal adversarial perturbations can be crafted to fool a network when added to any sample.17,18

The understanding of adversarial attacks has rapidly developed over the past several years. Akhtar and Mian wrote a broad survey on adversarial attacks in computer vision.19 Although adversarial instability may relate to overfitting, DNNs often generalize well to unseen data yet fail on previously seen data that are only slightly altered.20 Furthermore, it has been shown that linear models and other machine learning methods are also vulnerable to adversarial attacks.21 Early research attributed this phenomenon to a lack of data in high-dimensional problems, which leaves large portions of the total “data manifold” unstable.22,23 The literature has also reported a relationship between large local Lipschitz constants (with regard to the loss function) and adversarial instability.24–26 To our knowledge, the most unifying explanation is the robust features model, where it is shown that data distributions often exhibit statistical patterns that are meaningless to humans but correlate well with different classes.27 From a human’s perspective, these patterns are arbitrary and easily perturbed, but since models are trained to maximize only distributional accuracy, they have no reason to prioritize human-favored features over these patterns.

Training models for defending against adversarial attacks remains an open problem, affecting nearly every application of machine learning. Early attempts at defense methods by obfuscating the loss gradient were found to beat only weak attackers, proving ineffective for sophisticated attackers.28–32 Adversarial training, in which a model is iteratively trained on strong adversarial samples, has shown the best results in terms of adversarial robustness.12,33 However, the network size and computational time required are considerable for small problems, and improving adversarial robustness appears to sacrifice performance on clean data.34 Satisfactory performance on larger problems has not been achieved due to these limitations.

Another troubling, well-documented characteristic of these attacks is their transferability: models trained on the same task will often be fooled by the same attacks, despite having different parameters.12,21,35 This phenomenon is largely congruent with the robust features model, since these models are likely learning the same useful, but non-robust features. Nevertheless, transferability makes black-box attacks viable, where a malicious attacker does not need detailed knowledge of the target model.

Adversarial instability is particularly alarming for AI deployment in healthcare, where there is ample incentive and opportunity to leverage these attacks for medical fraud, as explained by Finlayson et al.36 To summarize, unsavory clinicians and institutions can already engage in “upcoding,” where unnecessary diagnoses are selected to maximize profits.37,38 This malpractice not only increases national healthcare costs, but also burdens patients with inappropriate treatments. To counter this, insurance companies set standards of proof for these decisions and will likely require algorithmic guarantees when AI models are involved. In workflows where deep learning models augment decision-making, malware that intercepts an input before it is analyzed can adversarially perturb the signal, triggering a profitable inference with no change to the model. Detecting such cases of foul play, even with trained readers, is difficult because the ground truth is often ambiguous in medical signal analysis, and the corrupted samples can be indistinguishable from benign samples.

Aside from targeted attacks, the mere existence of adversarial vulnerability breeds a “confidence crisis” in deep models: a model’s adversarial vulnerability reflects its reliance on spurious correlations in a dataset rather than true semantic understanding of the input.27 As such, it is difficult to trust these models in patient-critical tasks, where external factors, such as changes in recording conditions, could alter these correlations in unanticipated ways. In other words, there is no guarantee that perturbations along adversarial subspaces would not naturally occur during deployment. Consequently, protection against adversarial attacks is critical for public trust in and responsible deployment of medical AI.

Han et al. have shown that models trained for ECG classification are concerningly susceptible to natural-looking adversarial attacks.39 In short, the authors observed that traditional $\ell_\infty$-bounded PGD attacks produce square-wave artifacts that are not physiologically plausible in ECGs; to rectify this, the perturbation space was modified by applying Gaussian smoothing kernels in the attack objective, rendering plausible yet still highly effective adversarial samples.
Electrocardiography background
The ECG is a front-line, non-invasive tool for monitoring heart health. Skin electrodes measure electrical signals originating from the heart; the behavior of these signals over time corresponds to various events during the cardiac cycle. A healthy rhythm consists of a P wave, QRS complex, and T wave, which correspond to atrial depolarization, ventricular depolarization, and ventricular repolarization, respectively. Clinical ECGs have traditionally used 12 or 5–7 (Holter) leads, but single-channel ECGs have become more prevalent for continuous monitoring.40 These devices are either external or implantable loop recorders (ILRs) designed to record for multiple years.

A variety of downstream analytical tasks are associated with ECG, including biometric identification, respiratory estimation, emotional monitoring, and even fetal heartbeat monitoring. However, the most common application is detection of various arrhythmias, which can indicate disorders or disease risk.40 AFib, or an abnormally rapid atrial firing rate, is commonly assessed (such as in the 2017 PhysioNet Challenge3), but many arrhythmic classes exist, including left or right bundle branch block, premature atrial or ventricular contraction, ventricular fibrillation, tachycardia, and myocardial infarction (heart attack) itself. Certain classes require immediate and serious medical intervention, while others, such as AFib, are not immediately harmful but could still indicate risk of disease. The clinical utility of AFib detection is still an active area of investigation; a systematic review found AFib to be associated with increased risk of myocardial infarction in patients without coronary heart disease and increased risk of all-cause mortality and heart failure in all patients, implying value in detecting it.41

On the other hand, a randomized control trial using ILR in patients with at least one risk factor for stroke concluded that continuously monitoring for AFib in this population did not reduce the risk of stroke.42 This suggests that detecting AFib early may not provide additional information for managing stroke in patients who are already known to be at risk of the event. Nevertheless, ECG and its subsequent analysis are widely applied and investigated for their implications on patient cardiovascular health.

Clinician review is often required for ECG analysis; this process is resource consuming, especially in the case of continuous monitoring or large patient samples. Thus, automatic classification of these signals is desirable, but simple rule-based classifications often fail to generalize due to data heterogeneity between patients and the non-stationary nature of the signal within patients. As such, researchers have turned to data-driven techniques and machine learning to build ECG classification models. Convolutional neural networks (CNNs) have been the most dominant architecture for ECG arrhythmia classification, but deep belief networks, recurrent neural networks, long short-term memory, and gated recurrent units have all been investigated for the same task.43 For an extensive background, Merdjanovska et al. provide a comprehensive review of applications, public datasets, and deep learning research for ECG, and Ebrahimi et al. further survey common deep learning architectures for ECG.40,43

Uncertainty estimation in healthcare applications
As misdiagnosis in healthcare contexts can cause serious harm, the standard of trust required for AI to operate in this space is high. Rather than replacing clinicians, we envision AI tools augmenting clinical workflows by monitoring inputs over a large population, flagging abnormal or low-confidence instances for human review. To achieve this synergy, models must be capable of gauging their own confidence, recognizing conditions where they can and cannot perform well.44 Bayesian deep learning (BDL) is a promising field that models the parameters of a DNN as a distribution rather than a point estimation; sampling this distribution at inference time then allows one to estimate model certainty in an inference.45,46 Approaches for approximating and sampling the parameter distribution, including variational inference and Markov chain Monte Carlo with Hamiltonian dynamics, are often difficult to scale to large spaces.47,48 One simple approach is to train an ensemble of networks for the same task, with each network acting as a sample of the parameter space.49 However, this approach does not guarantee robustness: adversarial attacks in particular are known to transfer between different models, because these models (even with vastly different parameters) often learn the same unstable features. Furthermore, in high-dimensional problems with large parameter spaces, training, storing, and running inferences from numerous models quickly become infeasible. As such, the goal in training such ensembles should be to achieve adequate robustness and feature diversity with a small number of models.

Diversifying features in deep ensembles
To our knowledge, prior work on adversarial robustness has primarily quantified either white-box accuracy or black-box transferability but has not evaluated uncertainty via BDL. Furthermore, works in this field primarily test methods on lower-dimension datasets, such as MNIST or CIFAR10. Our goal is to efficiently train small but diverse deep ensembles capable of gauging uncertainty in worst-case scenarios, i.e., adversarial attacks. We contextualize this in the aforementioned ECG classification, a problem that is much higher in dimension. Adversarial training as a means of diversifying an ensemble is explored, and we also introduce two diversification methods that do not require adversarial sample computations, adding almost no overhead to the regular training process.

According to the robust features model, simply training networks with different parameters in isolation does not achieve adversarial robustness, as networks trained under the same conditions tend to converge toward the same learned features and vulnerabilities.27,50 As such, rather than achieving diversity in the parameter space, we turn the conversation to diversity in the feature space. A mechanism for incentivizing networks to learn different features is necessary. To this end, Yang et al. conceived DVERGE, which diversifies the learned features and adversarial weaknesses in a classification ensemble.51 However, this method requires full or partial computations of adversarial samples and round-robin-style training of networks, which necessitate considerable computation capability. In this work, we experiment with ensemble diversification methods that are based on adversarial gradients and other methods that are agnostic to these calculations.

Adversarial training is the best known defense against adversarial attacks and essentially consists of training a network on adversarial samples.12 Unfortunately, adversarial training is computationally expensive and reduces accuracy on natural data.34
Conventional adversarial training is the best known defense against adversarial attacks, but it does not detect attacks or quantify uncertainty; rather, it attempts to make a single network more robust to attacks. Furthermore, it achieves satisfactory performance only on small problems using large networks to fit more complex decision boundaries. As such, we adapt adversarial trained ensembles, in which individual networks are adversarially trained for additional time after natural training. Consequently, we are able to study adversarial training in the Bayesian ensemble framework and observe how vulnerabilities may be diversified between models.

We propose two distinct methods for diversifying learned features and test these against adversarial ECG attacks in Han et al. The first method, linear feature decorrelation, is based on previous work, which not only found a strong linear correlation in the latent space of networks trained on the same task but also found that adding a loss term to reduce the linear correlation greatly decreased the transferability of adversarial attacks. However, the decorrelation process proposed in that work is expensive, as it requires large batch sizes and parallel training of networks. We modify this decorrelation process to make it scalable to larger problems. The second method, which we refer to as Fourier partitioning, is heuristically simpler, employing linear time-invariant filters to partition the input space by frequency, forcing networks to learn features in different frequency bands. This method is inspired by recently discovered connections between the Fourier space and adversarial vulnerability, which demonstrated not only that neural networks can make accurate inferences by relying only on low- or high-frequency characteristics but also that most robustifying training methods only shift a network’s sensitivity to different frequency bands. As such, we find that a crude but efficient way to teach networks different features is to partition the original inputs by frequency, feeding data in different bands to different networks, and integrate their outputs via ensemble learning.

RESULTS

Overview

Two ECG datasets were used for all experiments: the 2017 PhysioNet challenge data (single channel, four classes) and the 2018 CPSC data (12 channels, nine classes). The following ensemble training strategies were tested, as well as combinations of methods (e.g., dec + part):

1. Baseline: conventionally trained ensemble, where all models are identically and independently trained
2. Dec: ensemble trained with the proposed linear feature decorrelation to diversify the model features
3. Part: ensemble trained using the proposed Fourier partitioning scheme
4. Adv: a baseline ensemble that undergoes additional ensemble adversarial training
5. Dverge: a baseline ensemble that undergoes additional DVERGE training

Ensembles produce multiple inferences, which can be processed in various ways to gauge epistemic and aleatoric uncertainty. Here, we adopt a normalized uncertainty approach from Mobiny et al. which calculates a normalized measure of certainty \( I_{\text{norm}} \) based on the mutual information between the sample and the model parameters (see methods, “ensemble training and inference”).

We tested each ensemble using validation data perturbed by both PGD and physiologically feasible smooth adversarial perturbations (SAP) attacks of varying magnitude \( \epsilon \), targeting the first model in each ensemble. Both attack algorithms were adapted to target all models in an ensemble as a white-box attack (see methods, “adversarial attacks,” for details). Figure 1 displays several example attacks from the PhysioNet 2017 data along with inferences, probability, and uncertainty values outputted by the baseline, dec, part, and dec + part ensembles.

Notably, we found that implementing the decorrelation step only slightly increased training time: conventional training took about 352 and 358 min per model on average for PhysioNet and CPSC, respectively; decorrelation added only about 10 min on average to this time in both cases (see methods, “experimental details,” for training parameters). Fourier partitioning did not noticeably increase training time.

Uncertainty and accuracy performance

To classify an ensemble inference as either certain or uncertain, a threshold, \( I_T \in [0, 1] \), can be applied to differentiate certain \( I_{\text{norm}} \leq I_T \) and uncertain \( I_{\text{norm}} > I_T \) predictions. A robust model is generally certain when it is correct and uncertain when it is incorrect. Thus, in addition to inference accuracy, we adopt the following three evaluation metrics from Mobiny et al.:

1. Correct-certain ratio \( R_{\text{cc}}(I_T) = P_{I_T \text{ correct} | \text{certain}} \): probability that the model inference is correct when it is certain
2. Incorrect-uncertain ratio \( R_{\text{iu}}(I_T) = P_{I_T \text{ incorrect} | \text{uncertain}} \): probability that the model is uncertain when it is incorrect
3. Uncertainty accuracy \( UA(I_T) = P_{I_T \text{ correct} \cap \text{certain} \cup I_T \text{ incorrect} \cap \text{uncertain}} \): probability of a desired outcome (either correct and certain or uncertain and incorrect).

All three measures depend on the variable uncertainty threshold \( I_T \). Thus, similar to a binary classifier, the overall efficacy can be found by integrating the measure as a function of \( I_T \in [0, 1] \) (i.e., finding the area under the curve [AUC], where larger values are more desirable).

Table 1 summarizes the average prediction accuracy and AUCs for the correct-certain ratio, incorrect-uncertain ratio, and uncertainty accuracy (UA) for natural, PGD, and SAP adversarial datasets for the PhysioNet 2017 experiments. Similarly, Table 2 reports the same metrics for the CPSC 2018 experiments. Some initial observations from these numbers are as follows: (1) for the PhysioNet 2017 experiments, dec, part, and dec + part generally have metrics that are comparable or superior to the baseline on natural (\( \epsilon = 0 \)) samples, but dec achieves better performance on stronger \( \epsilon = 50, 75, 100 \) PGD and SAP attacks. Although part does not benefit accuracy, the method shows increased UA, with dec + part yielding the best UA on stronger attacks. Adv slightly decreases natural accuracy but improves performance on all adversarial attacks, with the combination of dec + adv generally leading to superior performance in these instances. Dverge is comparable to the baseline
performance in this instance. (2) On the CPSC 2018 experiments, adv and dec + adv achieve improved robustness on perturbed data but sacrifice considerable accuracy on natural samples. dverge experiences a mild reduction in natural accuracy for a moderate increase on perturbed data, although this advantage diminishes with higher magnitude attacks. The combination of dec + part poses better performance on perturbed data relative to the baseline without sacrificing performance on natural samples.

Uncertainty difference between incorrect and correct predictions

Since uncertainty $I_{\text{norm}}$ should be a useful discriminative feature for distinguishing correct and incorrect predictions, it is desirable that incorrect predictions, on average, output higher uncertainty than correct predictions. Thus, we simply define the average difference in uncertainty between incorrectly and correctly classified samples:

$$\Delta I_{\text{norm}} = E[I_{\text{norm}}|\text{incorrect}] - E[I_{\text{norm}}|\text{correct}].$$

Figure 2 compares this $\Delta I_{\text{norm}}$ for all PhysioNet experiments and ensembles as a function of attack strength $\varepsilon$. For CPSC 2018 experiments, the average uncertainty for incorrect samples is observed, since ensemble accuracy at higher attack magnitudes is too low to estimate $E[I_{\text{norm}}|\text{correct}]$. Higher values are desirable for both measures. While comparisons between different ensembles are largely heterogeneous, the dec + part most consistently scores the highest across different attack conditions, suggesting better detection of adversarial samples via uncertainty estimation. adv also outperforms other methods (excluding dec + part) in the majority of cases with regard to these measures.

Performance on a partially attacked dataset

In a clinical setting, deep models should be used to augment clinical workflows. When the amount of data needing analysis outstrips the available time of qualified clinicians, deep ensembles can initially assess all inputs and defer the most uncertain samples to human readers. This begs the question, “If a deep ensemble is budgeted a certain number of cases that it can refer to human experts, then how many cases would still be misclassified?” To investigate this, we ran the following hypothetical experiment: (1) a dataset of all natural samples and a partially perturbed dataset were drawn. In the partially perturbed dataset, only 50% of the data were unperturbed, and 25%, 15%, and 10% of the data had $\varepsilon = 10, 50, 75$ and $\varepsilon = 0.025, 0.05, 0.15$ perturbations for the PhysioNet and CPSC data, respectively. (2) Each ensemble evaluated all samples, ordering the inferences from most to least confident. (3) Starting with the most confident samples, varying amounts of samples would defer to the deep model’s classification, while all other (less confident) samples would be correctly classified, presumably reviewed by clinicians. Note that this experiment is not meant to exactly reflect the actions and metrics of a true clinical workflow; rather, it is to investigate and compare the potential benefit of the aforementioned deep ensemble methods in augmenting human workers, particularly in the face of adversarially corrupted data.

Figures 3 and 4 plot the percentage of misclassified instances in the sample as a function of the percentage of cases referred to the deep ensemble for the natural and partially perturbed datasets, respectively. In addition, the total AUCs are shown, with lower values being better in this instance. Initially, one can see that adv, dec + adv, and dec + part + adv perform poorly on
the natural datasets, particularly the CPSC 2018 data. dec, part, and dec + part perform better than the baseline on the partially perturbed datasets while still performing comparable to or better than the baseline on the natural PhysioNet 2017 dataset. Baseline performs marginally better than dec, dec + part, and dverge in the natural CPSC data, but dec + part outperforms all other methods considerably in the perturbed instance.

**DISCUSSION**

Table 1 reflects the accuracy and uncertainty scores of tested ensembles on the PhysioNet 2017 data. Overall, it can be seen that decorrelation and partitioning do not negatively impact natural accuracy in this instance: in fact, overall ensemble accuracy, $R_{cc}$ and $R_{iu}$, are marginally higher on natural data in the dec, part, and dec + part compared to baseline. Furthermore, all ensembles, including the baseline, are somewhat robust to small-magnitude perturbations: Han et al.\(^{39}\) reported that their $\varepsilon = 10$ SAP attacks fooled a single network 74% of the time, but our results show only a 33.18% error rate for the baseline ensemble under these conditions. This indicates that in this instance, fooling multiple models is more difficult than fooling a single model, even without diversifying measures.\(^{12}\) However, this initial

| Attack strength $\epsilon$ (PGD) | 0 | 10 | 50 | 75 | 100 |
|-------------------------------|---|----|----|----|-----|
| baseline                      | 83.82 | 66.82 | 2.81 | 1.41 | 0.82 |
| dec                           | 84.41 | 68.23 | 15.12 | 6.68 | 3.52 |
| part                          | 86.75* | 58.97 | 2.58 | 0.35 | 0.35 |
| dec + part                    | 85.58 | 58.97 | 5.63 | 1.06 | 0.59 |
| adv                           | 83.35 | 78.55* | 53.34 | 25.79 | 9.61 |
| dverge                        | 84.17 | 61.43 | 3.05 | 1.06 | 0.82 |
| dec + adv                     | 80.07 | 76.79 | 58.62* | 41.74* | 23.68* |
| dec + part + adv              | 81.48 | 77.49 | 53.93 | 31.42 | 15.36 |

| Attack strength $\epsilon$ (SAP) | 0 | 10 | 50 | 75 | 100 |
|-------------------------------|---|----|----|----|-----|
| baseline                      | 83.82 | 66.82 | 2.81 | 1.41 | 0.82 |
| dec                           | 84.41 | 68.23 | 15.12 | 6.68 | 3.52 |
| part                          | 86.75* | 58.97 | 2.58 | 0.35 | 0.35 |
| dec + part                    | 85.58 | 58.97 | 5.63 | 1.06 | 0.59 |
| adv                           | 83.35 | 78.55* | 53.34 | 25.79 | 9.61 |
| dverge                        | 84.17 | 61.43 | 3.05 | 1.06 | 0.82 |
| dec + adv                     | 80.07 | 76.79 | 58.62* | 41.74* | 23.68* |
| dec + part + adv              | 81.48 | 77.49 | 53.93 | 31.42 | 15.36 |

R_{cc} (% AUC)

| Attack strength $\epsilon$ (PGD) | 0 | 10 | 50 | 75 | 100 |
|-------------------------------|---|----|----|----|-----|
| baseline                      | 85.46 | 69.27 | 2.30 | 1.11 | 0.65 |
| dec                           | 86.18 | 71.96 | 10.98 | 4.76 | 2.66 |
| part                          | 88.18* | 61.96 | 2.04 | 0.27 | 0.33 |
| dec + part                    | 87.64 | 67.17 | 11.22 | 2.99 | 2.02 |
| adv                           | 85.08 | 80.74* | 57.24 | 28.28 | 9.95 |
| dverge                        | 86.25 | 64.59 | 2.29 | 0.88 | 0.76 |
| dec + adv                     | 81.20 | 77.98 | 58.18* | 41.74* | 23.68* |
| dec + part + adv              | 83.68 | 80.15 | 57.22 | 32.25 | 12.91 |

R_{iu} (% AUC)

| Attack strength $\epsilon$ (PGD) | 0 | 10 | 50 | 75 | 100 |
|-------------------------------|---|----|----|----|-----|
| baseline                      | 13.77 | 16.84 | 15.93 | 15.09 | 14.72 |
| dec                           | 15.91 | 25.77 | 43.74 | 46.92 | 49.49 |
| part                          | 15.27 | 23.27 | 31.36 | 30.58 | 38.70 |
| dec + part                    | 19.57 | 42.05* | 68.27* | 69.66* | 67.92* |
| adv                           | 16.97 | 18.30 | 22.81 | 27.66 | 29.04 |
| dverge                        | 18.59 | 19.50 | 19.67 | 15.68 | 11.65 |
| dec + adv                     | 10.58 | 10.64 | 10.89 | 14.69 | 20.04 |
| dec + part + adv              | 21.05* | 22.05 | 32.04 | 43.34 | 55.07 |

UA (% AUC)

| Attack strength $\epsilon$ (PGD) | 0 | 10 | 50 | 75 | 100 |
|-------------------------------|---|----|----|----|-----|
| baseline                      | 81.65 | 66.39 | 17.49 | 15.85 | 15.17 |
| dec                           | 81.31 | 66.80 | 44.15 | 46.84 | 49.34 |
| part                          | 82.74* | 59.94 | 32.11 | 30.68 | 38.79 |
| dec + part                    | 81.07 | 63.47 | 67.72* | 69.63* | 67.98* |
| adv                           | 78.04 | 74.68* | 57.19 | 42.04 | 33.64 |
| dverge                        | 80.51 | 62.38 | 20.98 | 16.28 | 12.24 |
| dec + adv                     | 77.43 | 74.35 | 56.33 | 42.35 | 32.93 |
| dec + part + adv              | 74.40 | 71.31 | 55.60 | 49.25 | 52.93 |

R_{cc}, correct-certain ratio; R_{iu}, incorrect-uncertain ratio; UA, uncertainty accuracy. Asterisks indicate the best score for a metric.
robustness plummets in the face of more challenging, larger magnitude attacks, as seen in Table 1. Introducing decorrelation seems to improve robustness in both a regular and an adversarially trained ensemble, with dec + adv achieving the best accuracy on stronger attacks. While partitioning does not benefit classification accuracy, it improves uncertainty estimation, as seen with higher UA, with dec + part outperforming even the more expensive adversarially trained ensembles on this metric. These observations all suggest that improved network diversification that via features decorrelation can improve accuracy and uncertainty estimation, while Fourier partitioning improves only uncertainty estimation.

Table 1 also indicates little to no benefit from DVERGE training on the PhysioNet 2017 data; a small trade-off of natural accuracy for adversarial accuracy is seen in the CPSC 2018 data (Table 2). It should be noted that the original intent and testing of DVERGE was on black-box attacks, where the adversary has no direct knowledge of the target model’s parameters. Thus, it is possible that this method’s effectiveness is reduced when subject to much stronger white-box attacks, as is the case in our experiments, where the adversary can still find points of common instability among all models.

One can see that while adversarial training minimally affects natural performance on the PhysioNet data, natural performance...
is greatly degraded for adv, dec + adv, and dec + part + adv in the CPSC data in Table 2. This degradation has been observed for adversarial training and is likely due to the network’s limited capacity, which forces a trade-off between natural discriminative features and adversarially robust features. Indeed, as the dimension of the input space increases, as is the case of the higher-dimensional CPSC data, much larger networks and more training (both adversarial and natural) are needed to fit the robust but complex decision boundaries, rendering adversarial training less feasible for high-dimensional problems.

Decorrelation alone also provides little to no benefit on the CPSC 2018 experiments; we suspect that this is due to the increase in classes and input dimension. As the number of classes increases, the final feature layer, where decorrelation takes place, must increase in dimension such that feature vectors extracted from different networks can be uncorrelated while still correlating with the correct class. In addition, the expanded input space increases the overlap of unstable points among models, which the decorrelation mechanism must reduce. Despite this, combining dec with other methods such as adv or part yields more robust results compared to any individual method: dec + part generally shows the best UA, while dec + adv or dec + part + adv has superior accuracy against perturbed data. Thus, decorrelation still benefits other robustness methods in these higher-dimensional, higher-class count problems. However, as the benefits of using dec + part + adv over dec + adv or dec + part are not consistent across attack strengths in our experiments, benefiting from three methods simultaneously may always correlated with the utility of its uncertainty estimation: some ensembles may pose better overall accuracy but worse UA, $R_u$, and/or $R_{cc}$. It is important to evaluate Bayesian neural networks (BNNs) in a method similar to how they could be deployed to observe potential trade-offs between these qualities. This is the motivation behind the mixed dataset experiments (Figures 3 and 4), as it initially explores how well the uncertainty measures of different ensembles can prioritize clinician attention.

Unlike adversarial training, the combination of dec + part does not substantially change natural or adversarial accuracy, but results in the best UA in all adversarial instances. This suggests that the combination dec + part can improve adversarial uncertainty estimation without sacrificing natural accuracy, even in higher-dimensional problems.

While the metrics in Tables 1 and 2 overall suggest robustness benefits for linear feature decorrelation and Fourier partitioning, these results are admittedly heterogeneous. It is clear that, especially in higher-dimensional data, there is a strong trade-off between natural performance and adversarial robustness. Furthermore, an ensemble’s inference accuracy is not require more sensitive hyperparameter tuning and more research efforts.

While the metrics in Tables 1 and 2 overall suggest robustness benefits for linear feature decorrelation and Fourier partitioning, these results are admittedly heterogeneous. It is clear that, especially in higher-dimensional data, there is a strong trade-off between natural performance and adversarial robustness. Furthermore, an ensemble’s inference accuracy is not always correlated with the utility of its uncertainty estimation: some ensembles may pose better overall accuracy but worse UA, $R_u$, and/or $R_{cc}$. It is important to evaluate Bayesian neural networks (BNNs) in a method similar to how they could be deployed to observe potential trade-offs between these qualities. This is the motivation behind the mixed dataset experiments (Figures 3 and 4), as it initially explores how well the uncertainty measures of different ensembles can prioritize clinician attention.

Unlike adversarial training, the combination of dec + part does not substantially change natural or adversarial accuracy, but results in the best UA in all adversarial instances. This suggests that the combination dec + part can improve adversarial uncertainty estimation without sacrificing natural accuracy, even in higher-dimensional problems.

Figure 2. Uncertainty differences from various adversarial attacks
Top: PhysioNet 2017 difference in average normalized uncertainty between incorrect and correct samples with varying attacks. Bottom: CPSC 2018 average uncertainty on incorrectly classified samples with varying attacks. The x axis for all plots is the magnitude $\varepsilon$ of PGD (left) or SAP (right) adversarial attack.

Figure 2. Uncertainty differences from various adversarial attacks
Top: PhysioNet 2017 difference in average normalized uncertainty between incorrect and correct samples with varying attacks. Bottom: CPSC 2018 average uncertainty on incorrectly classified samples with varying attacks. The x axis for all plots is the magnitude $\varepsilon$ of PGD (left) or SAP (right) adversarial attack.
using the fast Fourier transform, Fourier partitioning is an efficient way to force ensemble models to extract different features, which improves uncertainty estimation against white-box ensemble attacks. Previous work mentioned several challenges with scaling decorrelated ensembles to larger problems, such as the need to train models in parallel and the large batch size needed to overdetermine the feature space with each training step. We find that our modifications, such as selecting the final hidden layer for decorrelation and compressing the feature space with random projections, have allowed us to scale this mechanism to higher dimensions (see details in the methods).

Furthermore, we found that both methods added little to no extra training time and penalized natural accuracy less than adversarial training. It should also be noted that both methods are orthogonal to gradient-based methods such as adversarial training and DVERGE and thus can be combined with these methods. Indeed, our results generally support that using decorrelation can improve the results of adversarial training.

Limitations of the study
A number of limitations exist for this study, which can be explored in future work:

1. Optimization of the design space: the introduction of Fourier partitioning and decorrelation introduces new hyperparameters for tuning. For decorrelation, the compression ratio for the features, dimension of the feature space, and batch size are all critical considerations. As previously discussed, we believe that expanding the feature space may be necessary as the number of discriminative classes increases. Fourier partitioning was inspired by the discovery that neural networks can often solve computer vision tasks with only partial frequency information. Thus, each ensemble filter should be designed to preserve sufficient information for the task but have non-overlapping vulnerabilities. Our experiments simply used two “ring filters,” which summed to an impulse response, but many other schemes could be explored in the same spirit. In addition, these methods can be used in combination with DVERGE or adversarial training. Future experiments should explore all these considerations extensively and introduce new applications as well.

2. Investigation of robustness against various attacks, corruptions, and shifts: this work focuses on adversarial attacks in ECG of varying magnitude, using both PGD and the more domain-specific SAP algorithms. However, robust uncertainty quantification is desirable in many other contexts, such as domain-specific noise corruption, data domain shifts, and out-of-sample detection. Since neither decorrelation nor Fourier partitioning explicitly optimizes against adversarial attacks, we hypothesize that their diversification benefits may extend to these other contexts.

3. Experiments mimicking clinical deployment: the trade-offs between a model's inference accuracy and UA make it necessary to test how deep models can work in synergy with clinicians. Our experiments compare the number of misclassified samples when uncertainty is used to theoretically prioritize clinician attention. In reality, however, clinical workflows are more complicated, and desirable outcomes will depend on the cost of false positives/negatives for different diagnoses, clinical resources, and the ability of human clinicians to more accurately diagnose certain diseases relative to a model. Thus, all these aspects should be considered in future translative work. Furthermore, we tested only a natural sample and an attacked sample where the distribution of attack magnitudes was roughly based on the assumption that larger perturbation attacks will be less common. Models should be rigorously tested specifically with the kinds and magnitudes of perturbations that one might expect in deployment.

Conclusion
Efficient and accurate confidence measurement is necessary for trust in AI systems. We have presented a novel approach for diverse network ensembles using two unique training methods that add little to no training time: a streamlined and accelerated decorrelation training strategy and a Fourier partitioning scheme. These ensembles achieve robustness by focusing on feature diversity between models. In addition, we adapt adversarial training to ensembles and test all methods along with DVERGE in the Bayesian ensemble framework. All approaches are applied to ECG classification with uncertainty estimation and tested for stability against state-of-the-art adversarial ECG attacks, demonstrating their merits and potential...
in solving large problems. Incorrect diagnoses can cause major harm in many healthcare tasks where AI can work alongside clinicians; predicting model confidence in these contexts is crucial. Thus, we speculate that diverse ensembles will play a key role in elevating trustworthiness and confidence in AI for applications such as tomographic image processing, radiomics, and multimodal diagnosis. We see applications of this approach for robust uncertainty estimation with a diversified ensemble, which discourages different models from extracting redundant features.

**METHODS**

**Ensemble training and inference**

Basic ensembles consist of multiple DNNs trained for the same task. An ensemble inference on sample $x$ is simply the average output of each model in the ensemble. Thus, for an ensemble with models $f_1, f_2, \ldots, f_k$:

$$\hat{y} = \frac{1}{R} \sum_{k=1}^{K} f_k(x).$$  \hspace{1cm} \text{(Equation 1)}$$

Note that for a classification task, this output is a discrete probability distribution.

For estimating epistemic uncertainty, we adopt the approach from Mobiny et al.,\(^5\) which defines the uncertainty of sample $x$ as the mutual information between the inferred label $\hat{y}$ and the underlying parameter distribution, in other words, how much additional information sample $x$ tells us about the true parameters:

$$I(y, \theta|x, D) = H(y|x, D) - H(y|x, \theta, D) = H(y|x, D) - \mathbb{E}_{\hat{y}|D}[H(y|x, \theta)].$$

$D$ is the training data. The first term is intractable, but can be estimated using the network ensemble as the entropy of the expected inference.$\(^5\) Thus, for an ensemble with $f_1, f_2, \ldots, f_k$ models, each of which outputs a discrete probability distribution over $C$ classes:

$$I(y, \theta|x, D) = - \sum_{c=1}^{C} f_{\text{ens}}(x)[c] \log f_{\text{ens}}(x)[c] + \frac{1}{R} \sum_{k=1}^{K} \sum_{c=1}^{C} f_k(x)[c] \log f_k(x)[c].$$

The scale of $I$ is relative and can vary between models and ensembles. Thus, we normalize the uncertainty with the minimum and maximum uncertainty values found during training.

$$I_{\text{norm}} = \frac{I - I_{\min}}{I_{\max} - I_{\min}}.$$

Note that test samples can have greater or less uncertainty than any sample encountered in the training set. Thus, values for $I_{\text{norm}}$ are not necessarily limited to $[0, 1]$. For a threshold $I_{\text{r}}$, which classifies samples as either “certain” or “uncertain,” metrics $R_{\text{sc}}, R_{\text{uc}},$ and $U_{\text{A}}$ were calculated empirically over an adversarial dataset based on the number of correct and certain, correct and uncertain, incorrect and certain, and incorrect and uncertain samples.

For the baseline ensemble, each model was trained independently in sequence. Other ensembles were trained using the methods described below.

**Decorrelation training**

**Decorrelation of two networks**

The intent of decorrelation training between two networks is to minimize the Pearson correlation coefficient between the latent features extracted by the two networks, as explained in Wiedeman and Wang.$^{50}$ Minimizing this value incentivizes the networks to extract different features for a task, reducing the transferability of network weaknesses.$^{27,50}$ Assume two classification models, $f_1$ and $f_2$, and sample batch $X$ of inputs with accompanying labels $Y$. Next, $Z_i = f_i(x)$ for $i = 1, 2$ are the latent feature extracted by model $i$ at layer $l$ from batch $X$. Note that $Z_i \in \mathbb{R}^{N \times D}$ where $N$ is the batch size and $D$ is the dimension of the latent space. A linear relationship estimating $Z_2$ from $Z_1$ with weights $W$ can be found using ordinary least-squares regression:

$$\text{minimize} \|Z_2 - Z_1 W\|_2^2$$

where $Z_1 = |Z_1, 1|$\(^w\) The Pearson correlation coefficient is then the ratio between the residual and the total sum of squares:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{total}}} = 1 - \frac{\|I - Z_1 (Z_2^T Z_1)^{-1} Z_1^T \|^2}{\|Z_2 - Z_1\|^2_2}.$$

Figure 4. Misclassification and deferral analysis on a partially attacked dataset

Percentage of misclassified cases with respect to percentage of cases deferred to the deep ensemble in the partially attacked dataset experiments for PhysioNet 2017 (left) and CPSC 2018 (right). Numbers are AUCs (lower is better).
To reduce this term during training, the decorrelation loss is defined as:

$$L_R = \log(SS_{\text{total}}) - \log(SS_{\text{res}}) + \epsilon. \quad \text{(Equation 2)}$$

$$L_R(Z_1, Z_2) = \log\left(\|Z_2 - Z_1\|^2 + \epsilon\right)$$

$$- \log\left(\left(1 - Z_2 (Z_2^T Z_1)^{-1} Z_1^T\right) Z_1^2\right) + \epsilon,$$

$$\text{where } \epsilon \text{ is some small constant for stability (set to } 10^{-5} \text{ in our experiments) and } Z_2 \text{ is an } N \times D \text{ matrix where each row is the sample mean of } Z_2. \text{ This decorrelation can be applied to model training simply by weighting and adding this loss to a conventional training objective (e.g., cross-entropy loss) for both networks, balancing feature decorrelation and individual network performance.}

**Scaling decorrelation to ensembles**

While decorrelation of two networks was shown to reduce transferability of adversarial attacks in Wiedeman and Wang, several issues impede its use in larger ensembles and higher-dimensional problems. First, this method requires training multiple networks in parallel to obtain $Z_1$ and $Z_2$, which linearly scales the memory needed for training and quickly becomes infeasible for more networks, larger networks, and larger data sizes. Second, calculating the loss in Equation 3 requires taking the pseudo-inverse of $Z_1 \in R^{N \times D+1}$, requiring batch size $N > D + 1$ for a sufficiently overdetermined system. Higher-dimensional problems often necessitate higher-dimensional latent spaces and smaller batch sizes, rendering this condition impractical.

To solve the above problems and extend the method to ensembles consisting of more than two networks, we employ the following modifications:

1. **Selecting the regression layer:** most deep classification networks can be divided into a multi-layer feature extractor and a linear layer followed by a softmax function (i.e., a logistic regression). We select the network layer just before this final linear layer, as it represents the highest-level features and is typically lower dimension than previous layers.

2. **Dimensionality reduction via random projections:** prior to computing the pseudo-inverse of the regressor, we compress its $D + 1$ dimensionality to $r < D + 1$ by applying a random projection, $R \in R^{D+1 \times r}$. To balance the asymmetry in this relationship, we randomly select which network’s extracted features act as the regressor and which as the regressand with each training batch. Our new loss is expressed as follows:

$$L_R(Z_1, Z_2) = \begin{cases} L_R(Z_1, Z_2 R) & \text{with prob. 0.5} \\ L_R(Z_2, Z_1 R) & \text{with prob. 0.5} \end{cases} \quad R \in R^{D+1 \times r} \sim N(0, 1/\sqrt{D}).$$

Although these projections individually do not capture all information in the regressor, they are drawn randomly with each training batch, preventing the networks from decorrelating only a subspace of the original feature space.

1. **Models are trained in sequence instead of parallel:** the first network is trained without any decorrelation loss. After a model is trained, its extracted features on all training samples are saved. While the next model is being trained, these features are loaded with the corresponding batch samples and then used for decorrelation. As such, rather than dynamically decorrelating multiple networks at once, which requires simultaneous training of all networks, we simply use the features extracted by the previously trained networks as constant values to decorrelate against. For decorrelating against multiple models, we average the modified correlation loss against all the previously trained models. Thus, the entire decorrelation loss for model $k$ in an ensemble is:

$$L_{\text{cor}}(Z_1, Z_{k-1} \cdots Z_0) = \frac{1}{k} \sum_{i=0}^{k-1} L_R(Z_i, Z_i). \quad \text{(Equation 4)}$$

**Final decorrelation scheme**

Figure 5 illustrates the sequential training of the decorrelated ensemble. The total loss for model $k$ on batch $(X_0, Y_0)$ with extracted features from $X_0$ using $k - 1$ previous models as $(Z_{k-1} \cdots Z_0)$ is:

$$L_{\text{total}} = L_{\text{ce}}(f_k(X_0), Y_0) + \lambda L_{\text{cor}}(Z_k, Z_{k-1} \cdots Z_0), \quad \text{(Equation 5)}$$

where $L_{\text{ce}}$ is the cross-entropy loss and $\lambda$ is a weighting hyperparameter. The implementation is summarized in the Algorithm 1. While stochastic gradient descent is shown here, any optimizer can be used for the gradient step.

**Fourier partitioning scheme**

With the Fourier partitioning scheme, the first ensemble model has no modification. The other models were trained normally, but inputs were filtered during both training and inference (Figure 6). This approach is inspired by Yin et al., who showed that neural networks can achieve high classification accuracy in many computer vision tasks with only a portion of an input’s frequency data and that models often overfit to discriminative features in specific frequency bands:

$$\tilde{y}_{ijk} = f_k(h_k \oplus x_i).$$

In practice, filter convolution was done by point-wise multiplication in the Fourier domain, computed using the fast Fourier transform.

**Adversarial attacks**

Adversarial attacks are formulated by maximizing the loss objective $L$ of the model $f$ by modifying $x$ (with the paired label $y$) within a set of valid perturbations $\Delta$ (i.e., perturbations that do not change the underlying semantic content of the input):

$$\max_{\delta} J(x + \delta, y) \quad \text{(Equation 6)}$$

subject to $\delta \in \Delta(x)$. 

Patterns 5, 101116, December 13, 2024
Two algorithms were used to craft adversarial attacks: PGD and SAP. PGD is widely used as a strong, iteratively optimized attack, where $\Delta$ is defined as a small, $\epsilon$ ball:  
$$x_i' = \text{Clip}_\epsilon(x_i' - \alpha \text{sgn} (\nabla_x L(f(x_i'), y_i))) $$ (Equation 7)

$x_i'$ is the sample at the $i$th iteration, $y$ is the corresponding label, $\alpha$ is a step size, and the clipping operation clips all values to be within the $\epsilon$ ball of radius $\epsilon$ around $x$, as well as any implicit bounds on the domain of $X$.

The attack algorithm targets a full ensemble, $f = f_{\text{ens}}$, by applying the cross-entropy loss as $L$ to ensemble inference from Equation 1, which yields:

$$L(f_{\text{ens}}(x_i'), y_i)) = -\log \frac{1}{K} \sum_{k=1}^{K} f_k(x_i') |y_i|.$$ (Equation 8)

Thus, computing the gradient $\nabla_x$ of this upper bound is equivalent to taking the gradient with respect to each individual model loss and averaging these results. Rather than compute forward and backward passes for all $K$ models at each iteration, we find it effective to simply alternate the target model with each iteration:

$$x_i' = \text{Clip}_\epsilon(\epsilon' - \alpha \text{sgn} (\nabla_x L(f_k(x_i'), y_i)))$$

where $k = i\%K$. (Equation 9)

SAP is a variation of PGD designed to craft smooth attacks for ECG signals. The details of SAP can be found in Han et al.\textsuperscript{39} To summarize, perturbation $\theta$ is iteratively optimized in a fashion similar to PGD, but it is also convolved with a sequence of $M$ Gaussian kernels at every step, each of which is parameterized by its width $s$ and standard deviation $\alpha$:

$$\theta_i = \text{Clip}_\epsilon(\theta_i' + \alpha \text{sgn}(\nabla_\theta L(f_k(x_i', \theta_i)), y_i)))$$

$$x'(\theta) = x + \frac{1}{M} \sum_{m=1}^{M} \theta @ K(s_m, \sigma_m).$$ (Equation 10)

As with the above PGD modification, the target model is alternated with each iteration to craft an ensemble attack. The convolution with Gaussian kernels smooths high-frequency perturbations, removing unrealistic square-wave artifacts. PGD and SAP attacks were optimized over 20 steps in total. For both attacks, $\alpha$ was scaled as $\epsilon/10$. All adversarial attacks were crafted from the validation to target one of the models in an ensemble. For experiments with PhysioNet 2017 data, the convolution kernels are identical to those used in Han et al.\textsuperscript{39} $s = (5, 7, 11, 15, 19)$, $\sigma = (1, 3, 5, 7, 10)$. For the CPSC 2018 experiments, we used $s = (9, 11, 15, 19, 21)$, $\sigma = (5, 7, 10, 13, 17)$.

**Experiment details**

The two ECG datasets used in this work are from the PhysioNet 2017 and CPSC 2018 challenges. The PhysioNet dataset contains 30- to 60-s-duration single-channel ECG recordings sampled at 300 Hz. All samples were zero-padded to be 60 s. Class labels were normal, A. Fib., other rhythm, and noise. Scaling of the signal magnitudes was identical to those in Han et al.\textsuperscript{39} The CPSC 2018 data contained 6- to 60-s ECG recordings at 500 Hz with nine class labels: normal, A. Fib., other rhythms, atrioventricular block, left bundle branch block, right bundle branch block, premature atrial contractions, premature ventricular contraction, ST-segment depression, and ST-segment elevated. All samples were truncated or zero-padded to be 48 s. For simplicity, the minority of samples labeled with multiple diagnoses were not used. In addition, all channels were normalized from $-1$ to $1$.

For all experiments, each ensemble consisted of $K = 3$ classification networks, each with the architecture used in Goodfellow et al.\textsuperscript{6} for experiments using the PhysioNet 2017 data. For experiments using the CPSC 2018 data, this architecture was simply modified to have 12 input channels and 9 output classes. Each network was trained for 80 epochs (batch size of 64) using the Adam optimizer with a learning rate of $10^{-3}$. Pytorch 1.8.1 was used with two NVIDIA Titan RTX GPUs and a 90/10 training/ validation split. For all ensembles using decorrelation training, hyperparameters $r = 32$ and $\lambda = 0.2$ were used (the uncompressed latent dimension $D$ was 64).

For ensembles that underwent additional ensemble adversarial training, each model in the ensemble was sequentially trained using adversarial samples. In practice, this is identical to regular training, except each sample batch is perturbed using PGD (Equation 7) prior to the forward training step. The perturbations target the model being trained and use $\epsilon = 10$ and 0.025 for PhysioNet 2017 and CPSC 2018, respectively. Each model was trained for an additional 2 h, translating to 6 extra training hours total for each adversarially trained ensemble.

DVERGE training is similar to adversarial ensemble training in that both use PGD to perturb a sample $x_i$ within a small $\epsilon$ bound. Differences are that in DVERGE, (1) this optimization is used to maximize similarity between the distilled features of $x_i$ and some other randomly drawn sample $x$ at some randomly selected feature layer $l$ of the network and (2) the samples perturbed using network $l$ are used to train other networks $i \neq j$. Thus, we use the feature distillation objective and training procedure from Yang et al.\textsuperscript{51}:

$$x_i'(x, x_i) = \text{argmin}_{x_i'} \|f_i'(x) - f_i'(x_i')\|^2$$

subject to $\epsilon \geq \|x - x_i\|_\infty$. (Equation 10)

We found that deploying this original implementation decreased natural accuracy on PhysioNet 2017 data, where there is a low number (4) of discrete classes. DVERGE creates new samples by distilling the non-robust features of one random sample onto another.\textsuperscript{51} However, if these two samples belong to
the same class (which is more likely to occur when there are few classes or class imbalances), the features distilled from one sample may already be similar to the other, negating any benefit. Consequently, we added the following constraint: $(x, x_\delta)$ batches are drawn such that $y \neq y_\delta$.

Values for $\varepsilon$ were identical to those used in adversarial ensemble training. The DVERGE training also ran for the same total training time (6 additional hours) as adversarial ensemble training. For each batch, feature layer $l$ was randomly, uniformly selected from the post batch-norm layers of the networks.

**Algorithm 1.** Training step for model $f_k$ (with parameters $\theta_k$) using decorrelation. $\lambda$ and $r$ are hyperparameters.

```plaintext
Draw $X_0, Y_0, [Z_{k-1}, \ldots, Z_0]$  ▶ Draw training batch and corresponding features from prior models

$Z_{k,b} \leftarrow f_k'(X_b)$
$N, D \leftarrow \text{shape}(Z_{k,b})$
$\hat{Y}_b \leftarrow f_k(X_b)$
$L \leftarrow L_{ce}(\hat{Y}_b, Y_b)$
$i \leftarrow 0

\text{while } i \leq k - 1 \text{ do}
Z_1, Z_2 \leftarrow Z_{k,b}, Z_{i,b}$
if $t \sim \text{Uniform}[0, 1] < 0.5$ then $Z_1, Z_2 \leftarrow Z_2, Z_1$
end if
$R \sim N(0, 1 / \sqrt{D}) \in \mathbb{R}^{D \times 1 \times r}$
$Z_1 \leftarrow [Z_1, 1]^T$
$L' \leftarrow L + \frac{\lambda}{2} L_{ce}(Z_1, Z_2)$  ▶ Apply decorrelation loss from Equation 3
$i \leftarrow i + 1
$end while$
$\theta_k \leftarrow \theta_k - \eta \nabla \ell \theta_k$
```

Figure 5. Decorrelation training process illustration

The current model $f_k$ is trained end to end (feature extractor and regressor) using both cross-entropy and our correlation loss. The correlation loss references previous models’ extracted sample features as opposed to training multiple models in parallel.
RESOURCE AVAILABILITY

Lead contact
Requests for further information and resources should be directed to and will be fulfilled by the lead contact, Ge Wang (wangg6@rpi.edu).

Materials availability
This study did not generate new materials.

Data and code availability
Data used in this paper are from the 2017 PhysioNet Cardiology Challenge4 and the 2018 CPSC.5 All original code has been archived at Zenodo at https://doi.org/10.56789/.14052936 and is publicly available as of the date of publication.56 An active repository containing our source code is also on GitHub: https://github.com/WANG-AXIS/DNA_ECG.

ACKNOWLEDGMENTS

This work was partially supported by US National Institutes of Health (NIH) grants R01EB026646, R01CA233888, R01CA237267, R01HL151561, R21CA264772, and R01EB031102 and a National Science Foundation Graduate Research Fellowship supporting C.W. The authors thank all members of the lab for their support.

AUTHOR CONTRIBUTIONS

C.W. and G.W. jointly conceived the idea for this study. C.W. designed code for executing all experiments and drafted the paper. G.W. was heavily involved in supervising the project, interpreting results, and editing the paper.

DECLARATION OF INTERESTS

The authors declare no competing interests.
7. Drenkow, N., Sani, N., Slighter, I., and Unberath, M. (2022). A systematic review of robustness in deep learning for computer vision: Mind the gap?. Preprint at arXiv. https://arxiv.org/abs/2112.00639.

8. Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. (2020). Measuring robustness to natural distribution shifts in image classification. Adv. Neural Inf. Process. Syst. 33, 18583–18599.

9. Wang, G., Qiao, J., Bi, J., Jia, Q.-S., and Zhou, M. (2020). An adaptive deep belief network with sparse restricted boltzmann machines. IEEE Transact. Neural Networks Learn. Syst. 31, 4217–4228. https://doi.org/10.1109/TNNLS.2019.292864.

10. Wang, G., Zhao, Y., Liu, C., and Qiao, J. (2024). Data-driven robust adaptive control with deep learning for wastewater treatment process. IEEE Trans. Ind. Inf. 20, 149–157. https://doi.org/10.1109/TII.2023.3257296.

11. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. Preprint at arXiv. http://arxiv.org/abs/1312.6199.

12. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2019). Towards deep learning models resistant to adversarial attacks. Preprint at arXiv. http://arxiv.org/abs/1706.06083.

13. Ren, K., Zheng, T., Qin, Z., and Liu, X. (2020). Adversarial attacks and defenses in deep learning. Engineering 6, 348–360. https://doi.org/10.1016/j.eng.2019.12.012.

14. Song, Y., Shu, R., Kushman, N., and Ermon, S. (2018). Constructing unrestricted adversarial examples with generative models. Preprint at arXiv. http://arxiv.org/abs/1805.07894.

15. Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., and Song, D. (2019). Generating adversarial examples with adversarial networks. Preprint at arXiv. http://arxiv.org/abs/1801.02610.

16. Liu, X., and Hsieh, C.-J. (2019). Rob-Gen: Generator, discriminator, and adversarial attacker. Preprint at arXiv. http://arxiv.org/abs/1907.10454.

17. Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). DeepFool: a simple and accurate method to fool deep neural networks. Preprint at arXiv. http://arxiv.org/abs/1511.04599.

18. Chaubey, A., Agrawal, N., Barmwal, K., Gullani, K.K., and Mehta, P. (2020). Universal adversarial perturbations: A survey. Preprint at arXiv. http://arxiv.org/abs/2005.08087.

19. Akhtar, N., and Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access 6, 14410–14430. Conference Name: IEEE Access. https://doi.org/10.1109/ACCESS.2018.2807385.

20. Goodfellow, I.J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. Preprint at arXiv. http://arxiv.org/abs/1412.6572.

21. Papernot, N., McDaniel, P., and Goodfellow, I. (2016a). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. Preprint at arXiv. http://arxiv.org/abs/1605.07277.

22. Gilmer, J., Metz, L., Faghri, F., Schoenholz, S.S., Raghu, M., Wattenberg, M., and Goodfellow, I. (2018). Adversarial spheres. Preprint at arXiv. http://arxiv.org/abs/1801.02774.

23. Dube, S. (2018). High dimensional spaces, deep learning and adversarial examples. Preprint at arXiv. http://arxiv.org/abs/1801.06364.

24. Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Favzi, A., De, S., Stanforth, R., and Kohli, P. (2019). Adversarial robustness through local linearization. Preprint at arXiv. http://arxiv.org/abs/1907.02610.

25. Hein, M., and Andriushchenko, M. (2017). Formal guarantees on the robustness of a classifier against adversarial manipulation. Adv. Neural Inf. Process. Syst. 30.

26. Roth, K., Kilcher, Y., and Hofmann, T. (2020-10-23). Adversarial training is a form of data-dependent operator norm regularization. Preprint at arXiv. http://arxiv.org/abs/1906.01527.

27. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. Preprint at arXiv. http://arxiv.org/abs/1905.02175.
4697–4708. Curran Associates, Inc. https://doi.org/10.5555/3495724.3496118

47. Graves, A. (2011). Practical Variational Inference for Neural Networks. In NIPS.

48. Neal, R. (1992). Bayesian learning via stochastic dynamics. In NIPS.

49. Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In NIPS.

50. Wiedeman, C., and Wang, G. (2022). Disrupting adversarial transferability in deep neural networks. Patterns 3, 100472. https://doi.org/10.1016/j.patter.2022.100472.

51. Yang, H., Zhang, J., Dong, H., Inkawhich, N., Gardner, A., Touchet, A., Wilkes, W., Berry, H., and Li, H. (2020). DIVERGE: Diversifying vulnerabilities for enhanced robust generation of ensembles. Preprint at arXiv. http://arxiv.org/abs/2009.14720.

52. Gal, Y., and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In International Conference of Machine Learning, 48. https://doi.org/10.5555/3045390.3045502.

53. Kendall, A., Badrinarayanan, V., and Cipolla, R. (2017). Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In Proceedings of the British Machine Vision Conference (BMVA Press), pp. 57.1–57.12.

54. Mobiny, A., Yuan, P., Moulik, S.K., Garg, N., Wu, C.C., and Van Nguyen, H. (2021). Dropconnect is effective in modeling uncertainty of bayesian deep networks. Sci. Rep. 11, 5458. https://doi.org/10.1038/s41598-021-84854-x.

55. Jensen, J.L.W.V. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. Acta Math. 30, 175–193. https://doi.org/10.1007/BF02418571.

56. Wiedeman, C., and Wang, G. (2024). Dna_ecg: First release. https://doi.org/10.5281/zenodo.14052935.