Neural Network Decision-Making Criteria Consistency Analysis via Inputs Sensitivity

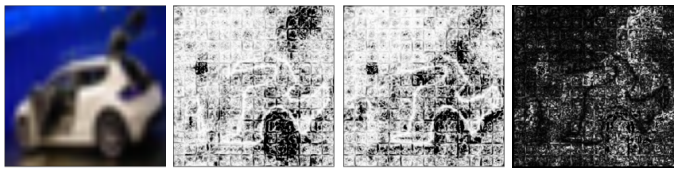
Eric Xing[†], Liangliang Liu[‖], Xin Xing[‡], Yunni Qu[¶], Nathan Jacobs[‡] and Gongbo Liang[§*]
[†]Western Kentucky University, Bowling Green, KY, USA
[‖]Henan Agricultural University, Zhengzhou, Henan, China
[¶]University of Toronto, Toronto, ON, Canada
[‡]University of Kentucky, Lexington, KY, USA
[§]Texas A&M University–San Antonio, San Antonio, TX, USA
[*]Corresponding authors: gongbo.liang@tamusa.edu

***Abstract*—Neural networks (NNs) have demonstrated exciting results on various tasks within the last decade. For example, the performance on image classification tasks has been improved dramatically. However, the performance evaluations are often based on a black-box performance, such as accuracy, while insightful analysis of the black-box, such as the prediction formation mechanism, is often missing. Empirically, a NN usually produces a stable overall performance on the same task across multiple training trials when treating it as a black-box. However, when unveiling the black-box, the performance is usually volatile. The decision-making criteria learned by the training trials are often significantly different, which is problematic in many ways. We believe achieving consistent criteria between different training trials is equally important to achieving high performance, if not more. This work, firstly, evaluates the decision-making criteria of NNs via inputs sensitivity using feature-attribution explanation methods in combination with computational analysis and clustering analysis. Through intensive experimentation, we find that decision-making criteria are easily distinguishable between training trials of the same architecture and task, suggesting the criteria learned between training trials are significantly inconsistent. To mitigate this inconsistency, we propose three general training schemes. Our demonstration result shows that the proposed methods effectively reduce the inconsistency of the decision-making criteria learned by different training trials while maintaining the overall performance.**

I. Introduction

Recently, the high performance of neural networks (NNs) has led to the rapid adoption of NN models in many domain-specific imaging analysis tasks [1]–[5]. Convolutional neural networks (CNNs) have been the major driving force of this development for the last decade [6]–[10]. In the last few years, the shift toward transformer-based architectures [11] led to the development of Vision Transformer (ViTs) [12], which have become a popular alternative approach to CNNs and produced numerous exciting results in the literature [13]–[16].

As a data-driven approach, neural networks learn features directly from the training data and make predictions based on the learned features. Due to the high performance of NNs, such features are often thought of as more robust or precise when compared to conventional hand-crafted features. Researchers are even seeking to find hidden patterns from such features that can be used to understand a given task [17]–[19]. However, the learned features are not guaranteed to be consistent when training the same NN architecture for multiple trials using the same training data and hyperparameters (Figure 1). Such a behavior could be potentially problematic in different ways.



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

Fig. 1: Feature visualization for two ViT models trained using the same architecture, training data, and hyperparameters. (a) Input image. (b)-(c) Integrated Gradients for Model I and II, show the two models using different features for decision-making making. Darker color indicates more important features. (d) Difference between the features used by the two models.

For instance, in scientific fields, a trustworthy result must be reproducible. Unfortunately, the inconsistent features learned by different trials imply specific training is not reproducible, which greatly limits the adoption of NNs in many scientific fields and raises concerns about trustworthiness. In addition, the inconsistent features provide the models with inconsistent decision-making criteria. This may not be acceptable for many real-world tasks, such as medical imaging analysis, where domain experts are expected to make decisions based on the same standards. This inconsistency also raises the concern of correctness and validity. It is crucial for many application domains, such as medical imaging analysis or autonomous driving, that performance needs to be maximized, and that decision-making criteria are sound and meaningful. A high-performance model based on irrelevant or uncomprehensive features would eventually lead to catastrophic consequences and cost human lives [20], [21]. Thus, we believe analysis beyond the overall performance, such as decision-making criteria consistency, is fundamental when applying neural networks to real-world tasks.

In this work, we propose to represent the decision-making criteria through input sensitivity analysis, which finds the most

influential areas in an input image in the prediction formation of the model. Then, computational analysis and clustering analysis are used to evaluate decision-making criteria consistency. To better understand the inconsistency between the decision-making criteria, we train and evaluate 98 models with four popular neural network architectures and six training strategies. We find that the inconsistency between the decision-making criteria does not have a clear relationship with the performance difference between the models. As expected, the inconsistency may be highly related to the degree of randomness during the training. One interesting finding is that ImageNet pre-training may be a simple solution to mitigate the inconsistency of decision-making criteria between training trials that is less well known. Furthermore, we propose three more general training schemes to reduce inconsistency of decision-making criteria: Naïve Averaged Training, Averaged Training with Clustering, and a Bucketing method.

We consider our contribution as below:

- Evaluating the common factors that may relate to the uncertainty of neural network decision-making criteria through input sensitivity analysis.
- Demonstrating the degree of randomness in training is highly related to the uncertainty of the learned decision-making criteria.
- Discovering ImageNet pre-training as a simple solution to mitigate the inconsistency of the learned decision-making criteria that is less well-known.
- Proposing three general training schemes that effectively reduce the inconsistency of the learned decision-making criteria while maintaining the model overall performance.

II. Background

A. Neural Network Learning

A typical neural network used in the imaging domain is often represented by h_Θ, where Θ is a list of tunable parameters or weights ($\Theta = \{\theta_1, \theta_2, ..., \theta_n\}$). In an ideal situation, the feature space of a given task is embedded in Θ after the parameters are fully optimized. Loss functions, such as the cross-entropy loss for classification, are often applied to guide neural network learning by providing feedback during the training. The loss is minimized by adjusting Θ to push the probability of target label to 1, which constrains the learning process to overall performance. However, because of this, the learning of decision-making criteria or features is less constrained.

B. Weights Initialization and ImageNet Pre-Training

The goal of neural network training is to converge to an optimal set of weights for the model that outputs the desired result on a given task. The weights are iteratively updated during the training from an initial set. Undesired initial weights may lead to vanishing or exploding gradients issues [22], [23]. The most widely used method is to initialize the weights randomly, close to zero, and under the normal distribution.

Initializing weights based on a pre-trained model is another popular approach. Such a method is also called transfer learning, a machine learning technique in which a model trained on one task can be repurposed to improve generalization in another setting [23], [24]. Transfer learning is widely used in the medical imaging domains, especially when the training dataset is small. Multiple strategies may be used for pre-training, such as supervised [25] or self-supervised pre-training [26]–[28]. ImageNet pre-training is one of the most widely used ones among all pre-training strategies [29]–[31].

III. Decision Criteria Consistency Evaluation

To understand the uncertainty of the learned decision-making criteria of neural networks, we selected four popular neural network architectures namely, AlexNet [6], ResNet-18 [8], ResNet-50 [8], and ViT [12]. We trained the networks using three pre-training methods: training from scratch (i.e., no pre-training) and ImageNet pre-trained weights with fixed or tunable feature extractors. We also considered both random seeding and fixed seeding during training. In total, twenty-four unique combinations of architecture, pre-training method, and seeding method were considered. Then, we ran three training trials for each combination using the same training data and hyperparameters. Finally, we compared the consistency of the learned decision-making criteria of the three training trials using computational-based and clustering-based methods.

A. Decision-Making Criteria Visualization

Effectively visualizing the learned decision-making criteria of a neural network is non-trivial. We visualize the decision-making criteria based on input sensitivity analysis, which aims to find the most influential areas from the input image. Two input sensitivity analysis methods are selected, namely Integrated Gradients (IG) [32] and GradientShap [33], [34].

Integrated Gradients is an axiomatic model interpretability algorithm that assigns an importance score to each input feature by approximating the integral gradients of a model's output with respect to the input. The method can be applied to any differentiable model and can be used to understand feature importance by extracting rules from the model. IG is a widely used state-of-the-art method for input sensitivity analysis and feature-attribution explanation.

SHAP (SHapley Additive exPlanations) [33] is a game-theoretic approach to explaining the output of any machine learning model. It connects optimal credit allocation with local explanations using the Shapley values from game theory and their related extensions. The method assigns each feature an importance value for a particular prediction based on each feature's share in the prediction. SHAP is another widely used method for input sensitivity analysis and feature-attribution explanation. GradientShap is an implementation of SHAP by `Captum.ai` [34].

B. Similarity Comparison

We evaluate the decision-making criteria consistency by comparing the similarity of the learned decision-making criteria between the training trials. After different training trails

are completed, we test the models of the training trials using the same test set and extract the IG and GS for each of the testing data. Then, computational-based and clustering-based comparisons are performed on each of the pairs of models.

1) Computational-Based Methods: We use the cosine similarity (CS) and the Canonical Correlation Analysis (CCA) [35] as the computational evaluation metrics.

Cosine similarity is widely used to measure the similarity between two non-zero vectors of an inner product space. The value is bounded in [0, 1]. Normally, the CS value 1 is considered "identical", and 0 is considered as "not similar at all". We apply CS on the corresponding IG or GS of a test sample and report the mean CS across all the test samples.

Canonical Correlation Analysis is a tool for quickly comparing two representations in a way that is both invariant to affine transform and fast to compute. The original paper applies CCA to measure the similarity between layers of neural network models. We use CCA to compare the learned feature spaces of two training trials represented by either IG or GS. We store the IG or GS of one training trial in a $N \times M$ matrix, where N is the number of samples and M is the number of features. Then, a single CCA value is calculated between two metrics that is bounded in [0, 1], with a higher value indicating a higher degree of similarity.

2) Clustering-Based Method: We apply hierarchical clustering on a set of IG or GS of the corresponding training trials to investigate whether the learned decision-making criteria of two training trials are separable.

Given N test samples, we extract the IG/GS of them from the two training trials, $2N$ features are generated after this step. Then, we compute the pairwise correlation of the $2N$ IG/GS and apply hierarchical clustering. We present the result using clustering heatmaps (e.g., Figure 2). The rows and columns in a heatmap are the same. Each row or column shows the correlation of the given sample with the test samples, including itself. The color bars on the left and top indicate the training trial where each sample is coming from. The color-coding in the heatmap represents the pairwise correlation of each testing sample. Since we are only interested in finding patterns that indicate whether the clusters are coupled with the training trials, the absolute value of the correlation is not critical.

Ideally, if two learned decision-making criteria are similar, we should see up to N small clusters containing two samples (Figure 2 right). Since we use the same samples to test the training trials, if the learned decision-making criteria are similar, the IGs/GSs of the same sample should be closer to each other and be clustered together. Otherwise, we may see larger clusters that contain samples majority from only one training trial (Figure 2 left).

C. Dataset

The Kather 5000 [36] dataset is used in this study that contains 5000 histological images of 150×150 pixels. Each image belongs to exactly one of eight tissue categories: tumor epithelium, simple stroma, complex stroma, immune cells, debris, normal mucosal glands, adipose tissue, or background (no tissue). The dataset was randomly partitioned into training and testing sets with a $4:1$ ratio.

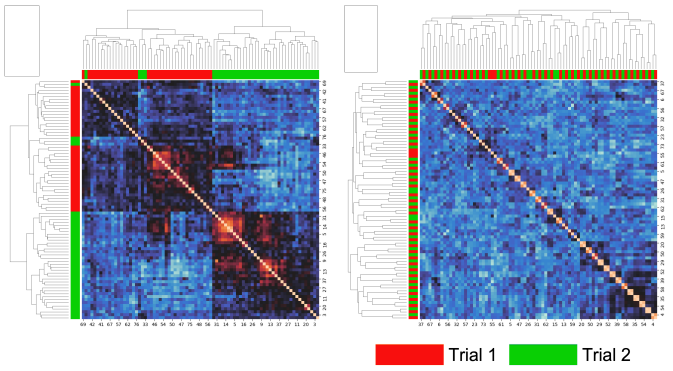


Fig. 2: The clustering evaluation of two ViT models. Left: ViT NoPreTrain-R. Right: ViT w/ Naïve Averaged Training.

D. Neural Network Models

Four network architectures are used in this study: AlexNet, ResNet-18, ResNet-50, and ViT. We keep all the layers, except the last, for each of the network architectures. We change the output dimension of the last layer to eight. For the transfer learning models, the pre-trained weights of AlexNet, ResNet-18, and ResNet50 are loaded directly from PyTorch [37]. We use the `timm` [38] weights for ViT transfer learning.

E. Result

Table I shows the average performance of the twenty-four unique combinations. The prefix **NoPreTrain** indicates the models are trained from scratch; **Fixed** and **Finetuned** show the models have fixed or tunable pre-trained feature extractors, respectively. Postfixes **-R** and **-NR** denote random seeding and fixed seeding, respectively. Each model is trained three times using the same training data and hyperpameters. The mean accuracy and the difference in performance between the three training trials are reported in the table.

The table reveals that all the models perform reasonably well, with the mean accuracy between 88.20% and 97.55%. The pre-trained ViT achieves the highest accuracy with fine-tuned feature extractor and random seeding. In general, fine-tuned models often have the highest accuracy. There is no clear winner between the NoPreTrain models and Fixed models. The table also shows that the performance between the three training trials of each model is generally very similar, with 17 out of 24 models being $\leq 1\%$ difference. The differences of all the models are $\leq 1.2\%$, except the pre-trained ResNet-50 model with the fixed feature extractor and random seeding. This result indicates neural networks usually have stable overall performance when evaluating the network as a whole.

Table II reveals the consistency of the decision-making criteria by reporting the average similarity between each pair of training trials. The result shows the pre-trained models with fixed feature extractors are more than likely to produce similar learned decision-making criteria between training trials. The

TABLE I: Mean Performance Difference Between Three Trials of Each Training Method

Models	Mean Performance (Difference)					
	NoPreTrain-R	**NoPreTrain-NR**	**Fixed-R**	**Fixed-NR**	**Finetuned-R**	**Finetuned-NR**
AlexNet	0.9330 (0.005)	0.9352 (0.005)	0.8892 (0.001)	0.8820 (0.011)	0.9591 (0.011)	0.9512 (0.003)
ResNet-18	0.8917 (0.012)	0.8970 (0.002)	0.8960 (0.010)	0.9100 (0.011)	0.9550 (0.000)	0.9604 (0.002)
ResNet-50	0.9148 (0.001)	0.9242 (0.012)	0.8724 (0.098)	0.9213 (0.001)	0.9182 (0.000)	0.9211 (0.000)
ViT	0.9277 (0.008)	0.9309 (0.002)	0.9490 (0.000)	0.9475 (0.003)	0.9755 (0.003)	0.9705 (0.003)

TABLE II: Decision-Making Criteria Similarity Analysis for Different Training Methods

Model	Feature	Canonical Correlation Analysis (Larger is Better)					
		NoPreTrain-R	**NoPreTrain-NR**	**Fixed-R**	**Fixed-NR**	**Finetuned-R**	**Finetuned-NR**
AlexNet	IG	0.1512	0.4071	0.6039	0.9662	0.4118	0.5853
	GS	0.1518	0.4024	0.5866	0.9610	0.4013	0.5718
ResNet-18	IG	0.1227	0.1329	0.5624	1.0000	0.2053	0.2813
	GS	0.1230	0.1315	0.5581	1.0000	0.2016	0.2799
ResNet-50	IG	0.1621	0.1573	0.3713	1.0000	0.6453	1.0000
	GS	0.1632	0.1565	0.3378	1.0000	0.6254	1.0000
ViT	IG	0.1132	0.1187	0.9612	0.9543	0.4286	0.4181
	GS	0.1120	0.1203	0.8087	0.8097	0.4232	0.4022
		Cosine Similarity (Larger is Better)					
AlexNet	IG	0.0587	0.4027	0.5922	0.9666	0.3620	0.5679
	GS	0.0572	0.3987	0.5718	0.9613	0.3487	0.5528
ResNet-18	IG	0.0060	0.0372	0.5752	1.0000	0.1732	0.2631
	GS	0.0048	0.0320	0.5382	1.0000	0.1671	0.2527
ResNet-50	IG	0.0068	0.0110	0.3472	1.0000	0.6839	1.0000
	GS	0.0048	0.0065	0.2943	1.0000	0.6278	1.0000
ViT	IG	0.0700	0.0640	0.9638	0.9523	0.3241	0.33203
	GS	0.0667	0.0710	0.7909	0.7822	0.3139	0.3097

models trained from scratch are the least likely to produce similar learned decision-making criteria. In addition, the learned decision-making criteria of models trained with fixed seeding are more stable than those trained with random seeding. There is no clear winner on architectures, which may lead to more stable decision-making criteria learning.

One interesting discovery is that the inconsistency in the learned decision-making criteria may not be coupled with the overall performance difference. For instance, AlexNet NoPreTrain-R has a mean accuracy of 93.30% with a 0.5% difference between the three training trials, and AlexNet Finetuned-R has a mean accuracy of 95.91% with a 1.1% difference between the training trials. Intuitively speaking, the latter model may have a more inconsistent learned decision-making criteria since it has a greater overall performance difference between training trials. However, Table II shows the learned decision-making criteria of the latter model are significantly more consistent than the former.

IV. Improving Decision-Making Criteria Stability

A. Reducing Randomness and ImageNet Pre-Training

As expected, we observe that the randomness of neural networks is one of the critical factors leading to unstable decision-making criteria learning. For instance, random seeding is one major contributor to the randomness of a neural network model that would negatively impact the learned decision-making criteria. Figure 3 (a) and (b) show the heatmaps of AlexNet models that were trained from scratch with random and fixed seedlings, respectively. Two clusters largely depending on the training trials are observed in the random seeding model (Figure 3 (a)), indicating the learned decision-making criteria of the random seeding model are inconsistent. However, the situation is improved when using fixed seeding. Similar results also are seen in many other examples.

In addition, we also find that ImageNet pre-training may be useful for reducing the inconsistency in the decision-making criteria learning. For instance, Figure 3 (c) shows when using ImageNet pre-trained weights, the decision-making criteria learning of AlexNet is further improved. Similar effects are also observed in other examples. To our best knowledge, this is the first work that was discovered ImageNet pre-training might be a simple solution to stabilize the decision-making criteria learning. However, a noticeable domain gap exists between ImageNet and specific domains, such as medical imaging, raising concerns about applying ImageNet pre-training in specific domains [26], [28]. Additionally, ImageNet pre-trained weights may not be readily available for many model architectures. People may need to pre-train their own ImageNet model, which is time-consuming and computationally expensive. Such drawbacks limit the usage of applying ImageNet pre-trained weights in mitigating the inconsistency in learned decision-making criteria.

B. Averaged Training

We propose using Averaged Training (AT), an ensemble-based approach, to learn stable decision-making criteria. The proposed method is not limited to any specific architecture, nor is a specific pre-training dataset required.

1) Naïve Averaged Training:

Intuition: In order to produce stable decision-making criteria, it is most intuitively reasonable to steer the model to an "average" decision-making criteria. If a large number

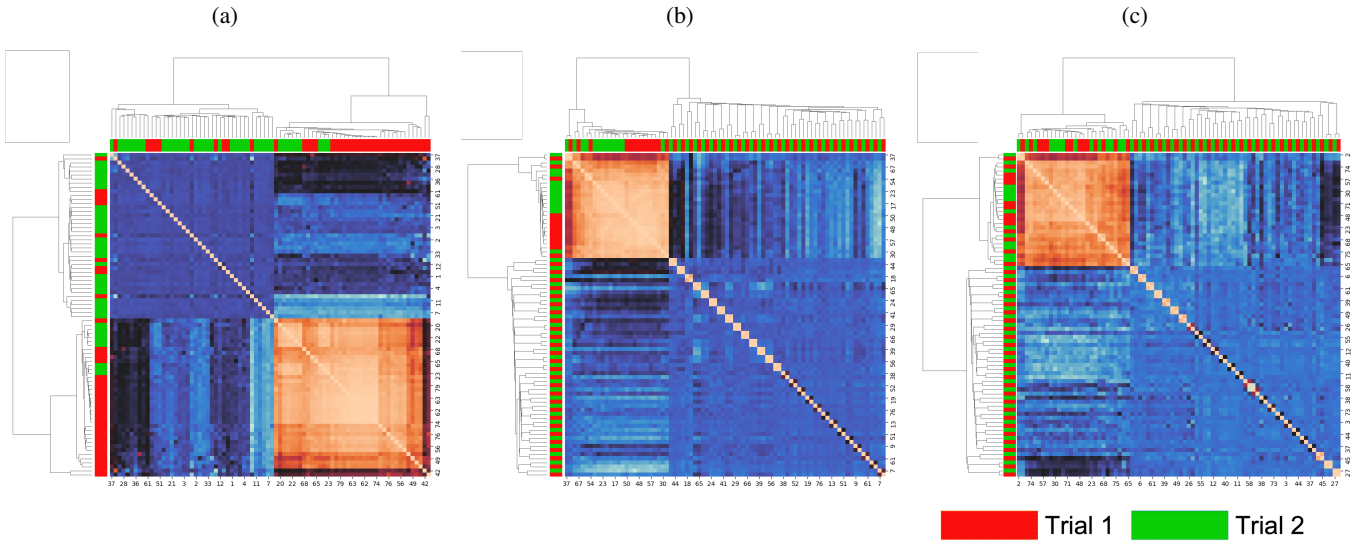


Fig. 3: The clustering evaluation of three AlexNet models: (a) NoPreTrain-R, (b) NoPreTrain-NR, (c) Finetuned-NR.

TABLE III: Feature Space Similarity of Vision Transformer (ViT) with Averaged Training

# of Sparingly Trained Models	Similarity	
	IG (CCA/CS)	GS (CCA/CS)
1*	0.1143 / 0.0644	0.1178 / 0.0705
10	0.1131 / 0.0651	0.1235 / 0.0770
30	0.2498 / 0.1902	0.2630 / 0.1752
50	0.4942 / 0.1720	0.4868 / 0.1434

* Same as ViT NoPreTrain-R.

of models are trained so that their decision-making criteria are mostly decided, averaging the parameters of these models will produce model parameters that utilize an average of the decision-making criteria of the averaged models.

Procedure: The method starts with separately training a number (n) of models from scratch. The initial training stops when each model begins to converge. A new model is then initialized using the averaged weights of the n pre-trained models and trained to complete. Under such a training schema, models generally start learning with an "average" decision-making criteria, which may reduce inconsistency.

Results: Table III shows the decision-making criteria consistency of ViT models using different n averaged over three training trials. Note that when $n = 1$, the model is the same as ViT NoPreTrain-R. The table reveals by increasing the number of sparingly trained models for averaging, the reduction in inconsistency increases. Figure 2 shows the clustering heatmap of ViT NoPreTrain-R and the ViT with AT training when $n = 50$. The decision-making criteria consistency is significantly improved by the proposed Naïve AT method. However, the drawback of the Naïve AT is the model performance may be degraded when the number of initial models is large (e.g., $n > 30$).

2) Averaged Training with Clustering:

Intuition: While steering a model to an average decision-making criteria increases decision-making criteria stability, this may intuitively degrade the model performance, as the model must learn to fit on decision-making criteria with a greater number of elements over the same number of training iterations. We propose to address this potential limitation by using clustering to steer the model to the most common decision-making criteria.

Procedure: We train a large number of models (m) until initial convergence and generate GS data for all models. We maintain a disjoint-set union data structure over all of the models. We union disjoint sets S_i and S_j where $i, j \leq m$ if there exist models A and B such that $A \in S_i$ and $B \in S_j$, and the cosine similarity between the GS data of A and B is above a threshold t. We check this condition on all $i, j \leq m$. After this procedure is complete, we find the disjoint set with the highest cardinality and generate the initial seeding based on this set of models. To do this, we iterate through each parameter in the model and find the average value of this parameter across all models in the set, producing an initial seeding. This procedure is summarized by Algorithm 1.

Results: Figure 4 shows the decision-making criteria consistency and accuracy of the Averaged Training with Clustering when $t = [0.35, 0.70]$ for $m = 40$ averaged over 3 training trials. We use solid lines to indicate our results and the dashed lines to indicate the baseline results (i.e., ViT NoPreTrain-R). The cosine similarity (CS) and Canonical Correlation Analysis (CCA) of IG are shown in the figure for different thresholds. The figure reveals that our method easily improves the learned decision-making criteria by 3X or more while maintaining similar accuracy. The GS-based consistency evaluation shows a similar result.

3) Bucketing Approach:

Intuition: To reduce the larger computational time associated with generating GS results for the Averaged Training with Clustering, we can still utilize a voting scheme without GS data. We do this by using voting to initialize the last

Algorithm 1 Averaged Training with Clustering

Require: Similarity Threshold t, Number of Models m, Models M, Model Parameters P, and GS data D, such that M_i is the ith model, P_i is Θ for M_i, and D_i is the GS data for M_i, $1 \leq i \leq m$, Cosine Similarity function CS, such that $CS(a, b)$ gives the similarity of GS data a and b, where $a, b \in D$

$S \leftarrow \{\{1\}, \{2\}, ..., \{m\}\}$ ▷ Each model is in its own set
for $i \leftarrow 1$ to m **do**
  for $j \leftarrow i$ to m **do**
    if $CS(D_i, D_j) \geq t$ **then** ▷ Found similar models
      $A \leftarrow$ set in S containing i
      $B \leftarrow$ set in S containing j
      S.remove(A) ▷ Remove old sets
      S.remove(B)
      S.add($A \cup B$) ▷ Insert union
    end if
  end for
end for
$l \in S$ such that $|l|$ is the greatest ▷ Largest set of models
$I \leftarrow \{0, 0, ..., 0\}$, such that $|I| = |P_1|$
for $i \in l$ **do**
  $Current \leftarrow P_i$
  for $j \leftarrow 1$ to $|I|$ **do**
    $I_j = I_j + Current_j/|l|$ ▷ Add averaged value for this parameter and model
  end for
end for
return I ▷ Return initial model seeding

convolutional (Covn) layer(s) with the most common values as determined by sparingly trained models.

Procedure: We start with training a large number of models (k) until initial convergence. Then, we iterate through the last l convolutional (Covn) layers to find the initial seeding. Specifically, for each layer, we first determine the mean and standard deviation (STD) of the values in the Covn kernels. Then, we iterate through all of the positions in the Covn kernels and create six buckets for each position based on the distance to the mean, namely: $\leq$ 2 STD below mean, 1 to 2 STD below mean, 0 to 1 STD below mean, 0 to 1 STD above mean, 1 to 2 STD above mean, and $\geq$ 2 STD above mean. After that, we place the models in the buckets. The bucket with the most models is used to determine a value for the kernel position in the initial seeding. The value is the midpoint of the interval of values it captures (or mean - 2.5 STD and mean + 2.5 STD for the leftmost and rightmost buckets, respectively).

Results: Our result on AlexNet averaged over 3 training trials shows the bucketing approach can significantly reduce the inconsistency of decision-making criteria (CS of IG = 0.1811, GS = 0.1793, much greater than the CS of IG = 0.0587, GS = 0.0572 associated with NoPreTrain-R AlexNet) learning while maintaining or improving the network's overall performance.

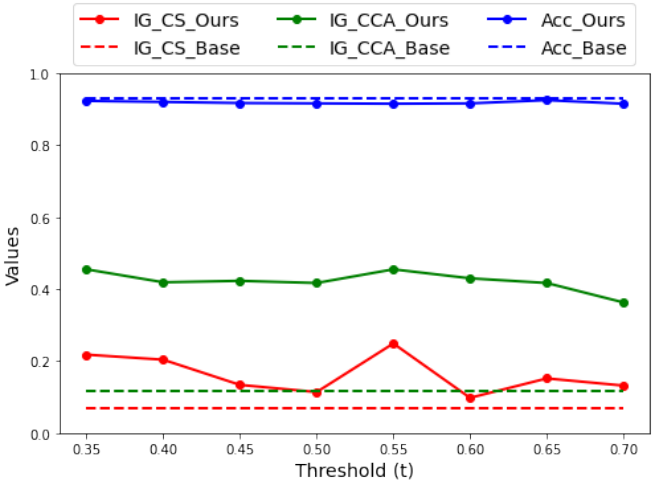


Fig. 4: Decision-making criteria consistency and accuracy of Averaged Training with Clustering with different thresholds.

V. Conclusion

Neural networks have shown promising learning power in diverse fields since the 2010s. As a result, academia and industry continue to seek to apply neural networks for more and more real-world tasks. However, fundamental challenges still need to be addressed to ensure the safety of using neural networks in our daily lives. We believe the uncertainty of the learned decision-making criteria is one of such challenges that prevent adopting neural networks in various domains, such as medical imaging analysis. Human experts are expected to make decisions based on the same standards consistently and predictably in the medical domain. However, the inconsistent decision-making criteria learned by multiple NN training trials provide different standards for the same task that are unacceptable and might lead to numerous additional issues.

Our experimentation indicates that the inconsistency of decision-making criteria is highly linked to the degree of randomness during network training. However, reducing randomness is often not practical. ImageNet pre-training may be a simple solution to improve consistency. Unfortunately, ImageNet pre-trained weights are not always readily available. In addition, the obvious domain gap between ImageNet and a specific imaging domain also raises concerns about adopting ImageNet pre-training in the domains. We propose using the Averaged Training, Averaged Training with Clustering, and Bucketing schema to provide stable decision-making criteria learning. We demonstrate the proposed methods significantly improve the inconsistency of the learned decision-making criteria while maintaining similar accuracy. We hope our work could be a strong baseline for further research.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.

[2] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[3] Y. Zhang, G. Liang, T. Salem, and N. Jacobs, "Defense-pointnet: Protecting pointnet against adversarial attacks," in *IEEE International Conference on Big Data*, 2019.

[4] Y. Su *et al.*, "A deep learning view of the census of galaxy clusters in illustristng," *Monthly Notices of the Royal Astronomical Society*, vol. 498, no. 4, pp. 5620–5628, 2020.

[5] G. Liang, X. Wang, Y. Zhang, and N. Jacobs, "Weakly-supervised self-training for breast cancer localization," in *Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 2020.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.

[9] L. Liu, J. Chang, Y. Wang, P. Zhang, G. Liang, and H. Zhang, "Llrhnet: Multiple lesions segmentation using local-long rang features," *Frontiers in Neuroinformatics*, p. 26, 2022.

[10] L. Liu *et al.*, "Decomposition-based correlation learning for multi-modal mri-based classification of neuropsychiatric disorders," *Frontiers in Neuroscience*, vol. 16, 2022.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need." [Online]. Available: https://arxiv.org/abs/1706.03762

[12] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *International Conference on Learning Representations*, 2021.

[13] R. P. Mihail, G. Liang, and N. Jacobs, "Automatic hand skeletal shape estimation from radiographs," *IEEE transactions on nanobioscience*, vol. 18, no. 3, pp. 296–305, 2019.

[14] X. Xing *et al.*, "Dynamic image for 3d mri image alzheimer's disease classification," in *European Conference on Computer Vision*. Springer, 2020, pp. 355–364.

[15] Q. Ying *et al.*, "Multi-modal data analysis for alzheimer's disease diagnosis: An ensemble model using imagery and genetic features," in *43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society*, 2021.

[16] X. Xing, G. Liang, Y. Zhang, S. Khanal, and N. Jacobs, "Advit: Vision transformer on multi-modality pet images for alzheimer disease diagnosis," in *IEEE International Symposium on Biomedical Imaging*, 2022.

[17] Z. A. Shboul, M. Alam, L. Vidyaratne, L. Pei, M. I. Elbakary, and K. M. Iftekharuddin, "Feature-guided deep radiomics for glioblastoma patient survival prediction," *Frontiers in neuroscience*, vol. 13, p. 966, 2019.

[18] "Integrating deep and radiomics features in cancer bioimaging."

[19] K. Kobayashi, M. Miyake, M. Takahashi, and R. Hamamoto, "Observing deep radiomics for the classification of glioma grades," *Scientific Reports*, vol. 11, no. 1, pp. 1–13, 2021.

[20] X. Jiang, M. Osl, J. Kim, and L. Ohno-Machado, "Calibrating predictive model estimates to support personalized medicine," *Journal of the American Medical Informatics Association*, vol. 19, no. 2, pp. 263–274, 2012.

[21] G. Liang, Y. Zhang, X. Wang, and N. Jacobs, "Improved trainable calibration method for neural networks on medical imaging classification," in *British Machine Vision Conference (BMVC)*, 2020.

[22] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[23] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[24] E. S. Olivas, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques: Algorithms, Methods, and Techniques*. IGI Global, 2009.

[25] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, *Transfusion: Understanding Transfer Learning for Medical Imaging*, 2019.

[26] H.-Y. Zhou, S. Yu, C. Bian, Y. Hu, K. Ma, and Y. Zheng, "Comparing to learn: Surpassing imagenet pretraining on radiographs by comparing image representations," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2020.

[27] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Advances in Neural Information Processing Systems*, 2020.

[28] G. Liang *et al.*, "Contrastive cross-modal pre-training: A general strategy for small sample medical imaging," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–1, 2021.

[29] Y. Zhang *et al.*, "2d convolutional neural networks for 3d digital breast tomosynthesis classification," in *IEEE International Conference on Bioinformatics and Biomedicine*, 2019.

[30] K. Mendel, H. Li, D. Sheth, and M. Giger, "Transfer learning from convolutional neural networks for computer-aided diagnosis: a comparison of digital breast tomosynthesis and full-field digital mammography," *Academic radiology*, vol. 26, no. 6, pp. 735–743, 2019.

[31] G. Liang *et al.*, "Joint 2d-3d breast cancer classification," in *IEEE International Conference on Bioinformatics and Biomedicine*, 2019.

[32] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[33] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017.

[34] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan, and O. Reblitz-Richardson, "Captum: A unified and generic model interpretability library for pytorch," 2020.

[35] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *Advances in Neural Information Processing Systems*, 2017.

[36] J. N. Kather *et al.*, "Multi-class texture analysis in colorectal cancer histology," *Scientific reports*, vol. 6, p. 27988, 2016.

[37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[38] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019.