

Advocacy Learning: Learning through Competition and Class-Conditional Representations

Ian Fox and Jenna Wiens

Department of Computer Science and Engineering, University of Michigan, Ann Arbor, USA
{ifox, wiensj}@umich.edu

Abstract

We introduce *advocacy learning*, a novel supervised training scheme for attention-based classification problems. Advocacy learning relies on a framework consisting of two connected networks: 1) N *Advocates* (one for each class), each of which outputs an argument in the form of an attention map over the input, and 2) a *Judge*, which predicts the class label based on these arguments. Each Advocate produces a class-conditional representation with the goal of convincing the Judge that the input example belongs to their class, even when the input belongs to a different class. Applied to several different classification tasks, we show that advocacy learning can lead to small improvements in classification accuracy over an identical supervised baseline. Though a series of follow-up experiments, we analyze when and how such class-conditional representations improve discriminative performance. Though somewhat counter-intuitive, a framework in which subnetworks are trained to competitively provide evidence in support of their class shows promise, in many cases performing on par with standard learning approaches. This provides a foundation for further exploration into competition and class-conditional representations in supervised learning.

1 Introduction

In recent years, researchers have proposed a large number of modifications to the standard supervised learning setting with the goal of improving performance [Parascandolo *et al.*, 2018; Vaswani *et al.*, 2017]. These modifications focus on training different parts of the network (*i.e.*, subnetworks) to *cooperate*. However, in several real-world settings, such as in economics and law, agents who *compete* are critical for identifying good solutions. While recent work in adversarial networks investigates the use of competition for training models, the final model evaluated (*i.e.*, the generator) is cooperative [Goodfellow *et al.*, 2014]. In contrast, we investigate a training scheme in which subnetworks compete during training *and* evaluation. In our model, subnetworks compete to provide evidence in the form of class-conditional attention

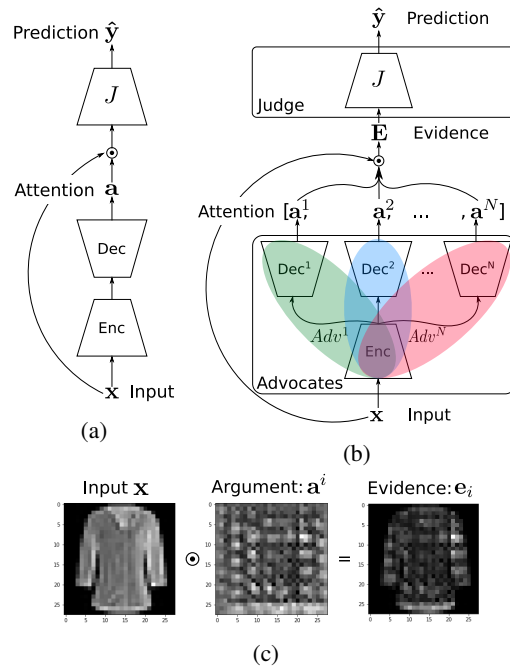


Figure 1: **a)** A simple single-attention framework. The encoder-decoder produces an attention map $\mathbf{a} \in \mathcal{R}^{n \times n}$, which is combined with the input $\mathbf{x} \in \mathcal{R}^{n \times n}$ via an element-wise product (indicated by \odot) to create the input to the decision module, or Judge J . **b)** Our advocacy learning framework. Each decoder Dec^i is trained separately to output a class-conditional attention map, or argument $\mathbf{a}^i \in \mathcal{R}^{n \times n}$, which is combined with the input via element-wise product (separately for each attention map) to create evidence $\mathbf{E} = [\mathbf{e}_0, \dots, \mathbf{e}_N]$, where $\mathbf{e}_i = \mathbf{a}^i \odot \mathbf{x}$ is evidence supporting class i . Each advocate is shown in a different color, the number of Advocates is equal to the number of classes. **c)** An example of an attention map \mathbf{a}^i used to generate evidence \mathbf{e}_i .

maps. Here, we use the term ‘attention map’ to refer to a filter that indicates parts of the input that are useful for accurate classification, similar to the idea of saliency [Itti *et al.*, 1998]. An example of a standard network architecture with attention is given in **Figure 1a**. We hypothesize that class-conditional attention maps could offer advantages over standard attention maps by emphasizing portions of the input indicative of their class.

Our proposed approach consists of two main subnetworks: a single Judge and multiple Advocates (see **Figure 1b**). Each Advocate produces an attention map that *advocates* for a particular class. A decision is reached by the Judge, which weighs the arguments produced by the Advocates. For this approach to work well, there must be a balance among the Advocates (so that each Advocate can influence the Judge), and the Judge must be able to effectively use the given evidence (so as not to be deceived by the advocates). We achieve this balance via *advocacy learning*, which trains the components jointly, but according to multiple objectives. These different objectives are key to striking the right balance between providing strong but factual evidence. We also explore a variant, honest advocacy learning, where the Advocates are not trained to deceptively compete with one another, but still provide class-conditional attention maps. In a series of experiments, we compare advocacy learning to several baselines in which the entire network is trained according to the same standard objective. Across several image datasets, we observe a small but consistent improvement in classification accuracy by using class-conditional attention maps.

2 Methods

We propose a novel approach to optimizing networks for supervised classification that encourages class-conditional representations of evidence in the form of attention maps. We hypothesize that, depending on how they are learned, class-conditional attention maps could offer advantages over standard attention maps by encouraging competition among components of the network. At a high level, the Judge learns to solve the classification problem given some evidence, while each Advocate supplies that evidence by arguing in support of their class. This setup disentangles evidence supporting each class, encouraging strong class-conditional representations. Advocacy learning consists of both a specific architecture (*i.e.*, Advocate and Judge subnetworks) and a specific method for training, both are described below.

2.1 Problem Setting

We consider the task of solving a multi-class classification problem in a supervised learning setting with element-wise attention. We assume access to a labeled training set consisting of labeled examples $\{\mathbf{x}, y\}$, where $\mathbf{x} \in \mathbb{R}^d$ (where d may be a product $d_1 \times d_2$, such as in an image) and $y \in \{1, \dots, N\}$, where N is the number of classes. We refer to the one-hot label distribution entailed by y as \mathbf{y} , so $\mathbf{y}[y] = 1$ and $\mathbf{y}[j] = 0$ for all $j \neq y$. We use square brackets for indexing into a vector. We focus on deep learning methods due to their suitability for representation learning. We indicate the parameters of deep models as θ and subscript them according to the particular subnetwork being referenced. The parameters of the Judge are θ_J and the parameters for each Advocate i are θ_i where $i \in \{1, \dots, N\}$. Our proposed approach aims to solve the multi-class classification problem through a novel training scheme, designed to produce class-specific evidence.

2.2 Network Architecture

As mentioned above, our proposed approach is composed of two sets of modules: multiple Advocates and a single Judge. A high-level overview of our architecture, which we call an Advocacy Net, is given in **Figure 1b**. Here, for context, we describe a generic framework, providing specific implementation details in a later section.

Advocate Subnetwork

This subnetwork consists of N Advocates Adv^i , where i corresponds to the index of the class the Advocate represents. Given the input \mathbf{x} , each advocate generates an argument in the form of an attention map $Adv^i(\mathbf{x}; \theta_i) \rightarrow \mathbf{a}^i \in [0, 1]^d$. The Advocate modules produce an attention map with dimensionality equal to the input using a convolutional encoder-decoder, as is standard for producing pixel-level output in images [Badrinarayanan *et al.*, 2017]. Note that for complex input, such as medical images, other fully convolutional architectures such as U-Nets may be more appropriate [Ronneberger *et al.*, 2015]. Based on these attention maps, each Advocate presents an argument \mathbf{e}_i (or evidence) to the Judge in the form of an element-wise product between attention maps and the input, $\mathbf{e}_i = \mathbf{a}^i \odot \mathbf{x}$.

Each Advocate is trained to emphasize aspects of the input indicative of the Advocate’s class. This differs from a supervised attention map, which focuses on aspects of the input indicative of the inputs underlying class. In our implementation, Advocates share some underlying evidence in the form of a shared encoder. This allows the Advocates to share useful representational abstractions.

Judge Subnetwork

The Judge J takes as input the combined evidence $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_N] \in \mathbb{R}^{N \times d}$, and outputs a probability distribution over classes $J(\mathbf{E}; \theta_J) \rightarrow \hat{\mathbf{y}}$. We make specific class predictions using $\arg \max(\hat{\mathbf{y}})$. The architecture of the Judge is flexible; the only limitation is that the input size must be proportional to the total number of classes. In our implementation, the Judge module is a convolutional network with fully connected output layers.

Though certain constraints on the architecture of the network are necessary, the interplay among the modules and how they are trained are key to the advocacy learning framework. Trained end-to-end with the objective of minimizing training loss, there would be no difference between the proposed architecture and a network with multiple attention channels. In the next section, we describe the key differences in how we train the Advocates vs. the Judge.

2.3 Training Algorithm

The complete advocacy learning algorithm is presented in **Algorithm 1**. We learn the parameters of the Judge subnetwork by minimizing the cross-entropy loss: $CE(\hat{\mathbf{y}}, y) = -\log \hat{\mathbf{y}}[y]$, as is standard for classification. The Advocates are trained according to a different objective. In particular, Advocate i is trained by minimizing the advocate cross-entropy loss: $CE^A(\hat{\mathbf{y}}, i) = -\log \hat{\mathbf{y}}[i]$. Under this objective, each Advocate is trained to represent samples from all classes as its own. We also consider a variant, called honest advocacy

Algorithm 1: Advocacy Learning Algorithm

Input : Labeled training data $\mathbf{D} = \{\mathbf{x}_k, y_k\}_{k=1}^S$ where S is the number of samples, $\mathbf{x}_k \in \mathcal{R}^d$, and $y_k \in \{1, \dots, N\}$

Output: Trained Network $A = (J, Adv^1, \dots, Adv^N)$

```

1 Initialize parameters for Judge  $\theta_J$  and Advocates  $\theta_1, \dots, \theta_N$ ;
2 while training do
    /* Showing single sample for simplicity */
3 Draw example  $(\mathbf{x}, y) \sim \mathbf{D}$ ;
4 for  $i \in \{1, \dots, N\}$  do
5      $\mathbf{a}^i \leftarrow Adv^i(\mathbf{x}; \theta_i)$ ;
6      $\mathbf{e}_i = \mathbf{a}^i \odot \mathbf{x}$ ;
7 end
8  $\mathbf{E} \leftarrow [\mathbf{e}_1, \dots, \mathbf{e}_N]$ ;
9  $\hat{\mathbf{y}} \leftarrow J(\mathbf{E}; \theta_J)$ ;
10  $L_J = -\log(\hat{\mathbf{y}}[y])$ ; // Cross-Entropy Loss
11  $\theta_J \leftarrow \theta_J - \eta \nabla_{\theta_J} L_J$ ;
12 for  $i \in \{1, \dots, N\}$  do
13     if not honest or  $i = y$  then
14         /* Honest Advocates update on true examples */
15          $L_{Adv^i} = -\log(\hat{\mathbf{y}}[i])$ ;
16          $\theta_i \leftarrow \theta_i - \eta \nabla_{\theta_i} L_{Adv^i}$ ;
17     end
18 end
19 return  $A = (J, Adv^1, \dots, Adv^N)$ 

```

learning, in which the Advocates are not trained to deceive, but aim to minimize:

$$CE^{HA}(\hat{\mathbf{y}}, i, y) = \begin{cases} -\log \hat{\mathbf{y}}[i], & \text{if } i = y \\ 0, & \text{otherwise} \end{cases}$$

We optimize the parameters of the Judge and Advocates by making a prediction and then interleaving steps of gradient descent, updating the Judge and each Advocate individually according to their respective loss functions. At each step, we freeze the parameters of all other subnetworks.

3 Baselines & Experimental Setup

We evaluate our proposed advocacy learning approach across a variety of tasks and compare against a series of different baselines. In this section, we explain each baseline and provide implementation details.

3.1 Model and Baselines

On all datasets, we compare (honest) advocacy learning against two baselines that incorporate attention:

- **Attention Net:** this baseline modifies the Advocacy Net architecture by removing all but one attention module. The one remaining module is trained using a standard end-to-end optimization approach minimizing cross entropy loss. This allows us to compare advocacy learning against a similar model using standard supervision.

- **Multi-Attention Net:** two differences exist between Attention Nets and Advocacy Nets: the optimization procedure and the architecture. To tease apart these two aspects, we include a comparison against a model with an identical architecture, but trained using a standard end-to-end loss.

3.2 Implementation Details

We implement our models using PyTorch [Adam *et al.*, 2017]. Our specific model architecture (number of layers, filters, *etc.*) is available via our public code release¹.

In our experiments, we optimize the network weights using Adam [Kingma and Ba, 2014] with a learning rate of 10^{-4} , and use Dropout and batch normalization to prevent overfitting. We split off 10% of our training data to use as a validation set for early stopping. We cease training when validation loss fails to improve over 10 epochs. Model performance is reported on the canonical test splits for each dataset. We regularize the attention maps by adding a penalty proportional to the L1-norm of the map to encourage sparsity consistent with common notions of attention. Parameters were initialized using the default PyTorch method.

4 Results and Discussion

We begin by examining the performance of advocacy learning across three image datasets, analyzing when and how advocacy learning impacts performance. We then present performance on a significant real-world medical dataset and modified datasets designed to highlight the effects of competition and deception in learning. We conclude by providing some intuition for why advocacy learning works.

4.1 Advocacy Learning on Multi-Class Balanced Image Data

We begin by examining the performance of our Advocacy Net variants and baselines on two publicly available image classification datasets: MNIST and Fashion-MNIST [Xiao *et al.*, 2017]. As described above, the parameters of the Advocate modules are not optimized to improve overall classification performance, and as a result their inclusion could lead to a reduction in performance (by deceiving the Judge). However, we find that this is not the case (Table 1). On these two datasets, advocacy learning does as well as or outperforms all baselines. The improvement is most pronounced in Fashion-MNIST, perhaps due to the denser images or the baseline performance allowing room for improvement. Moreover, this difference is not solely attributable to the class-conditional nature of the attention-maps; across both datasets the Advocacy Nets outperform the Honest Advocacy Nets (test accuracy 99.42 vs. 99.32 in MNIST and 91.62 vs. 90.81 in FMNIST). This suggests that deception, in addition to competition, can help produce useful attention maps.

We also examined performance on a more challenging image classification problem, CIFAR-10. When using a similar architecture and training procedure as in the above datasets, we found the Advocacy Net improved over other approaches

¹<https://github.com/igfox/advocacy-learning>

Model	Dataset	
	MNIST	FMNIST
Attn Net	99.16 ± 0.30 (0%)	89.71 ± 0.86 (0%)
Multi-Attn Net	99.33 ± 0.09 (20%)	90.11 ± 0.40 (4%)
Honest Adv. Net	99.32 ± 0.08 (19%)	90.81 ± 0.34 (11%)
Advocacy Net	99.42 ± 0.05 (31%)	91.62 ± 0.41 (19%)

Table 1: Accuracy ± standard deviation over 5 random seeds on the datasets among the various models. Values in parentheses show reduction in error rate relative to Attention Net. Note that the difference between the Advocacy Net and other approaches are consistent across the individual seeds.

(in particular, test accuracy 83.47 vs. 79.73 for the Multi-Attention Net), though both approaches performed poorly relative to state-of-the-art. Thus, we examined replacing the Judge network with a ResNet-110, using a fixed-epoch training scheme with learning rate decay (similar to a popular open source implementation²). This increased the performance of the Multi-Attention Net and drastically lowered the Advocacy Net performance (test accuracy 30.54 vs. 92.01), suggesting that under our current training scheme Advocacy Learning is unstable when the Judge is high-capacity relative to the Advocates. Notably, Honest Advocacy Learning continues to perform well, beating the Multi-Attention Net (test accuracy 92.68 vs. 92.01).

4.2 Impact of Class Conditional Attention

Results in **Table 1** demonstrate that class-conditional attention can improve upon supervised attention maps. Honest Advocacy Nets perform similarly to Multi-Attention Nets on MNIST and slightly better on FMNIST (99.32 vs. 99.33 and 90.81 vs. 90.11). These architectures differ only in that the attention maps in the Honest Advocacy Net are class-conditional. The competition introduced by the Advocacy Net further improves performance, outperforming the Multi-Attention Net on both datasets.

To further compare advocacy learning with the end-to-end supervised baselines, we plot the averaged difference between the confusion matrices of the Multi-Attention Nets and Advocacy Nets (**Figure 2**). Overall, we observe that Advocacy Nets result in improvements for a subset of class pairs (e.g., classes 4 and 9), but leave the majority of predictions unchanged. On MNIST (**Figure 2(a)**), we find a few examples where advocacy learning lowers performance. In particular, the Advocacy Net is more likely to misclassify 8s as 9s; though the reverse error rate (9s as 8s) does not increase. This is likely due to the asymmetric morphological relationship among the digits: using per-pixel attentions in $[0, 1]$, an 8 can be obscured to look like a 9, but the converse is less likely. The Advocacy Net appears to help emphasize curves in the input (reducing the instances with 9 classified as 4 or 7 classified as 9). On Fashion-MNIST (**Figure 2(b)**) we observe that certain class pairs (pullovers or coats vs. shirts) are markedly improved, while most others are unaffected. This suggests that advocacy learning improves performance by distinguish-

²<https://github.com/bearpaw/pytorch-classification>

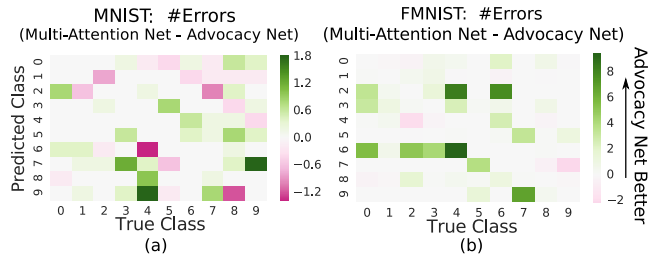


Figure 2: Averaged difference across five runs in confusion matrices between the multi-attention and Advocacy networks (≥ 0 means the Advocacy net performed better) on a) MNIST and b) FMNIST. We zero out the diagonal elements to focus on misclassification. A positive number means the Multi-Attention Net made more misclassifications than the Advocacy Net and vice-versa. We observe the Advocacy Net tends to improve performance across classes, but can make certain morphologically similar examples (i.e., 8 vs 9 in a) more difficult.

ing among classes with similar morphology; though this analysis is confounded by the fact that it tends to be those class pairs that have the greatest potential room for improvement.

Qualitative examples of attention maps from the Honest Advocates and Multi-Attention Net are given in **Figure 3**. We found Honest Advocacy Nets gave denser (and thus more interpretable) attention maps than Advocacy Nets. We observe that both Advocacy Nets and Multi-Attention Nets generate a variety of attention maps with checkering characteristic of deconvolutional layers [Odena *et al.*, 2016]. An interesting example of class-conditional behavior is shown by the advocate for class 1. This Advocate, representing the “pants” class, emphasizes the sides of the shirt, similar to pants legs.

4.3 Balancing the Advocates and Judge

Since the Advocates are encouraged to deceive the Judge, a natural question arises - How does the relative capacity of these components impact performance? To answer this question, we consider a variation of our Advocacy Net, in which we can vary the capacity of different pieces. We replace the convolutional layers in the Advocate encoder and Judge with some number of convolutional residual blocks, as in [He *et al.*, 2016]. By changing the number of blocks, we can increase/decrease the capacity of the Advocate or Judge. Architectural details of this modification can be found in our code. We varied the number of residual blocks from 1-5 and 1-3 in the Judge and Advocates respectively. We observed that the Advocacy Nets achieved the highest classification accuracy when Judge capacity was high and Advocate capacity was low (best accuracy 99.46%). We performed an identical architecture search with the Multi-Attention Net; there was no capacity setting that beat the best results attained by the Advocacy Net (best 99.34% vs. 99.46%). This suggests that it is important for the Judge to have more capacity than the Advocates; though our experiments on CIFAR suggest that a balance must be struck. While a high-capacity Judge is better able to use evidence provided by the Advocates, it may train slowly, and thus be more susceptible to deceptive Advocates.

To better understand the impact of deception in advocacy

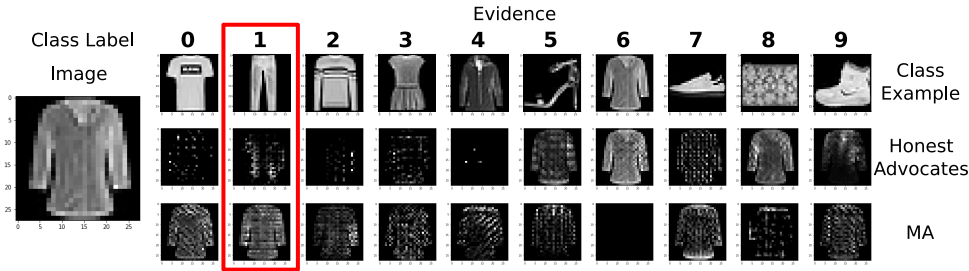


Figure 3: Evidence generated from a Fashion-MNIST example. The top row shows a sample from the class the column represents. The second row shows evidence from an Honest Advocate Net (the order corresponds to class). The bottom row shows evidence generated by the Multi-Attention Net (the ordering is arbitrary). The image is an example of class 6 (shirts). Of particular note is the argument generated by the Advocate for class 1 (pants).

Model	Dataset (Metric)			
	(AUROC)	MIMIC (AUPR)	Imbalanced	MNIST (Accuracy) Binary
Attention Net	83.29 ± 0.84 (0%)	45.79 ± 1.80 (0%)	98.68 ± 0.48 (0%)	99.23 ± 0.22 (0%)
Multi-Attention Net	82.80 ± 0.94 (-3%)	45.74 ± 1.71 (0%)	99.00 ± 0.13 (24%)	99.32 ± 0.14 (11%)
Honest Advocate Net	83.22 ± 0.93 (0%)	46.34 ± 1.73 (1%)	99.17 ± 0.06 (37%)	99.31 ± 0.13 (10%)
Advocate Net	77.73 ± 1.43 (-33%)	39.03 ± 4.58 (-12%)	99.17 ± 0.14 (37%)	98.72 ± 0.58 (-66%)

Table 2: Accuracy ± standard deviation over 5 random seeds on the modified MNIST datasets among the various models. The MIMIC results are reported in terms of AUPR. Values in parentheses show reduction in error rate relative to Attention Net.

learning, we took a fully trained network and examined the effect of freezing the Judge while continuing to update the Advocates for both advocate and Honest Advocacy Nets. We found that training without the Judge did not affect network performance in the Honest Advocate Net, but decreased performance in the advocate net by 85%. Thus, adaptations by the Judge play a crucial role in maintaining advocate net accuracy.

Up to this point, we have considered only Advocacy Nets in which all Advocates share an encoder. Such an architecture could encourage implicit sharing of information, possibly tempering the negative effects of deception. To test this hypothesis, we evaluated an Advocacy Net without a shared encoder on MNIST and FMNIST. On both datasets (averaged across 5 runs), the Advocacy Net without a shared encoder achieved lower performance both in absolute terms and relative to an Honest Advocacy Net without a shared encoder (98.29 vs. 99.05 for MNIST, 86.47 vs. 89.29 for FMNIST). This suggests the shared encoder is an important way for Advocates to balance performance.

4.4 Generalization to other Settings

The results presented so far all involve multi-class image datasets with balanced classes. To explore how these assumptions affect the performance of advocacy learning, we applied advocacy learning to a large electronic health record (EHR) dataset, MIMIC III [Johnson *et al.*, 2016]. This dataset, a publicly available repository of EHR data, has become an important benchmark in the machine learning for health community [Harutyunyan *et al.*, 2017], and is helping to drive advances in precision health [Desautels *et al.*, 2016;

Maslove *et al.*, 2017]. We used the clinical time-series subset of the database for mortality prediction, as in [Harutyunyan *et al.*, 2017]. We also considered variants of MNIST that break the multi-class and balanced assumptions. These additional experiments test the generalizability of our findings to i) different data types (time series as opposed to images), ii) imbalanced classes, and iii) binary labels. Our results are presented in **Table 2**.

We report our results on MIMIC in terms of the area under the precision recall curve (AUPR) and the receiver operating characteristic curve (AUROC), since the task is binary with considerable class imbalance in the test set (13.23% positive). Notably, in this task advocacy learning performs slightly worse than all baselines. However, honest advocacy learning provides a small benefit relative to the baselines in terms of AUPR.

This reversal of the results from **Table 1** is interesting, and helps illuminate cases where advocacy learning may or may not work. There are many differences between MIMIC and MNIST/FMNIST that could explain why advocacy learning applied to MIMIC fails. First, this task involves classifying patients based on a feature vector of vital sign measurements. Our attention mask allows for input values to be reduced to some proportion between 0 and 1. This type of attention can allow for pictures of shirts to look like pants, or an 8 to look like a 9, but it may not allow for vitals from sick patients to look like vitals from healthy patients (or vice-versa). Other major differences, besides the data type, include class imbalance and the number of classes. As these are qualities that can be imbued in other datasets, we further examine them by modifying MNIST to create new datasets we call Imbalanced

MNIST and Binary MNIST.

In Imbalanced MNIST, we subsampled the training set, introducing class imbalance. After subsampling, the least represented class, 0, had 600 training samples, and each successive class had 600 additional samples. The test set remained unchanged. We found that class imbalance in the training set lowered the performance of all models by 0.1-0.3% relative to those same models performance with balanced training data; the Advocacy Net was more strongly affected than the Honest Advocacy Net. However, both models resulted in similar accuracy (advocacy learning 99.17 ± 0.14 vs. honest advocacy learning 99.17 ± 0.06).

Binary MNIST contains only two classes: 4 and 9. The per-class number of examples in the training and test set were unchanged. We found that the switch to a binary formulation *reduced* absolute performance for the Advocacy Net by 0.47% relative to the performance of the full network evaluated only on 4s and 9s (99.19 vs. 98.72), whereas the Honest Advocacy Net *improved* binary performance by 0.32%. This decrease suggests that, in practice, the competition among many Advocates helps the Judge achieve good performance in the presence of deception.

In all datasets, the class-conditional attention provided by honest advocacy learning did not hurt, and in the presence of imbalanced data helped. This suggests the value of class-conditional representations, with or without deception.

5 Related Work

The fact that advocacy learning, which encourages deceitful subnetworks, works at all, let alone better than the baselines in some tasks, may be surprising. Several recent works, however, have found competition useful for learning [Goodfellow *et al.*, 2014; Silver *et al.*, 2017; Sabour *et al.*, 2017]. Specifically, adversarial relationships, or situations where subnetworks compete with one another, have recently garnered interest [Goodfellow *et al.*, 2014; Ghosh *et al.*, 2018]. For example, Ghosh *et al.* examine a multi-generator setting, where each generator is encouraged to capture distinct portions of the class distribution. This resembles how advocates captures relevant evidence for their corresponding class. However, while the relationship between an Advocate and the Judge is not entirely cooperative (the Advocate may argue for an untrue class), neither is it entirely adversarial (the Advocate may argue for a true class). While these systems used competition among networks during training, competition within a network has been used as well. A winner-take-all competitive framework was found to lead to superior semi-supervised image classification performance [Makhzani and Frey, 2015], and the dynamic routing used in Capsule Networks can be seen as type of competition [Sabour *et al.*, 2017].

Beyond this work in competition, advocacy learning is related to both i) input transformation and ii) debate agents. First, others have considered transformations of the input in the context of improving classification. Parascandolo *et al.* proposed the use of a mixture-of-experts model to learn inverse data transforms to improve performance. The competition among experts to provide quality transforms of the input resemble the competition among Advocates to convince the

Judge. Our work differs in that: 1) our Advocate updates are unsupervised, 2) the goal of the Advocates is not to improve performance, and 3) Advocates are assigned classes. Hong *et al.* examined the use of class-conditional attention for improving semantic segmentation. Their attention maps are similar to the argument maps generated by Advocacy Nets, but are trained with an end-to-end objective. Similarly, Capsule Nets [Sabour *et al.*, 2017] use class-specific capsules at the output layer to define a class-conditional parse of the input. This can be viewed as a bottom-up analog to the class-specific attention maps produced by Advocates, but it trained end-to-end unlike advocacy learning.

Second, work in AI debate sets up a similar task to our own, training agents to ‘debate’ in order to convince a Judge about the class associated with an input [Irving *et al.*, 2018]. The authors use Monte-Carlo tree search to simulate a debate with the goal of identifying a series of pixels to convince a pre-trained Judge classifier of an input example’s class. While conceptually similar to advocacy learning, our work differs in motivation and methodology. In addition to considering a different learning framework: neural networks, vs. Monte-Carlo tree search, there are two key differences in problem formulation: 1) our work does not use a pre-trained Judge, and 2) our work involves Advocates, which are assigned classes, not debaters, which choose classes.

6 Conclusion

We have presented a novel approach to attention-based supervised classification: advocacy learning. Our approach divides a network into two subnetworks: i) a set of Advocates trained to provide arguments supporting their corresponding class, and ii) a Judge that uses these arguments to predict the true class. Over a series of experiments on three publicly available datasets, we showed that class-conditional attention can improve performance relative to standard attention, and that in some circumstances competitive training can further improve performance.

The use of a deep network for the Judge has important implications for the interpretability of the derived attention maps. If the Judge is a high-capacity nonlinear network, then the evidence it may find convincing will, by default, be uninterpretable. However, the flexibility of the proposed architecture means that work on training interpretable networks or interpreting trained networks applies [Ribeiro *et al.*, 2016]. Future work could consider using these techniques to create interpretable Advocacy Nets. Extensions may also consider improving the balance between cooperation and competition by controlling the ratio of honest and deceptive updates, or the ratio of class-specific updates. Currently, the proposed architecture is limited by the one-to-one relationship between the number of classes and number of advocates, which makes training on datasets like ImageNet infeasible. Future work could examine methods to remove this linear relationship, such as training advocates that work across class hierarchies. While there are many avenues for future improvements, the experiments explored in this work suggest that competition and class-conditional representations can in some cases be used to improve the utility of attention in classification tasks.

References

- [Adam *et al.*, 2017] Paszke Adam, Chintala Soumith, Chanan Gregory, Yang Edward, DeVito Zachary, Lin Zeming, Desmaison Alban, Antiga Luca, and Lerer Adam. Automatic Differentiation in PyTorch. In *Autodiff Workshop in the Proceedings of Neural Information Processing Systems*, 2017.
- [Badrinarayanan *et al.*, 2017] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- [Desautels *et al.*, 2016] Thomas Desautels, Jacob Calvert, Jana Hoffman, Melissa Jay, Yaniv Kerem, Lisa Shieh, David Shimabukuro, Uli Chettipally, Mitchell D Feldman, Chris Barton, David J Wales, and Ritankar Das. Prediction of Sepsis in the Intensive Care Unit With Minimal Electronic Health Record Data: A Machine Learning Approach. *JMIR Medical Informatics*, 4(3), September 2016.
- [Ghosh *et al.*, 2018] Arnab Ghosh, Viveka Kulharia, Vinay P. Nambodiri, Philip H. S. Torr, and Puneet K. Dokania. Multi-Agent Diverse Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8513–8521, 2018.
- [Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [Harutyunyan *et al.*, 2017] Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, and Aram Galstyan. Multitask Learning and Benchmarking with Clinical Time Series Data. *arXiv:1703.07771 [cs, stat]*, March 2017. arXiv: 1703.07771.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [Irving *et al.*, 2018] Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv:1805.00899 [cs, stat]*, May 2018. arXiv: 1805.00899.
- [Itti *et al.*, 1998] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, November 1998.
- [Johnson *et al.*, 2016] Alistair EW Johnson, Tom J. Pollard, Lu Shen, H. Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.
- [Kingma and Ba, 2014] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference for Learning Representations*, 2014.
- [Makhzani and Frey, 2015] Alireza Makhzani and Brendan J. Frey. Winner-take-all autoencoders. In *Advances in Neural Information Processing Systems*, pages 2791–2799, 2015.
- [Maslove *et al.*, 2017] David M. Maslove, Francois Lamontagne, John C. Marshall, and Daren K. Heyland. A path to precision in the ICU. *Critical Care*, 21(1):79, April 2017.
- [Odena *et al.*, 2016] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.
- [Parascandolo *et al.*, 2018] Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning Independent Causal Mechanisms. In *International Conference on Machine Learning*, pages 4036–4044, July 2018.
- [Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 1135–1144, New York, NY, USA, 2016. ACM. event-place: San Francisco, California, USA.
- [Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [Sabour *et al.*, 2017] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3856–3866, 2017.
- [Silver *et al.*, 2017] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, October 2017.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747 [cs, stat]*, August 2017. arXiv: 1708.07747.