# XAI-Guided Interventional Retraining for Domain Generalization

## 2020-04-02

Suppose $m$ is a model trained in one domain. Here is a simple method to obtain, from $m$, another model $m'$ that hopefully generalizes better to other domains.

Let $\{h_u\}_{u=1}^F$ be all the feature units, and $\{z_c\}_{c=1}^C$ be the logit units. The weight between $h_u$ and $z_c$ in $m$ is $w_{uc}$.

- For each training example $(\mathbf{x}_i, y_i)$, feed it to the model $m$ and run an XAI method to obtain an heatmap $e_i$, which is what $m$ considers as the core evidence in $\mathbf{x}_i$ for the class $y_i$. Assume $e_i$ is normalized to the interval $[0, 1]$.

- Create a "purified" input $\mathbf{x}'_i$ by combining $\mathbf{x}_i$ and $e_i$. The simplest way is to do pointwise multiplication. We can also convert $e_i$ into a binary mask using a threshold before the pointwise multiplication.

- Feed $\mathbf{x}'_i$ to $m$ and compute the activation $h'_{ui}$ of each feature unit $h_u$. Let $a_{ui} = h'_{ui} w_{uy_i}$. It is the contribution of feature $h_u$ for classifying the purified example $\mathbf{x}'_i$ into the class $y_i$.

  Assume $e_i$ correctly highlights the core evidence in $\mathbf{x}_i$ for class $y_i$. Then the features $h_u$ that do not focus on the core evidence would have low activation $h'_{ui}$ on the "purified" example $\mathbf{x}'_i$, and hence $a_{ui}$ would be low. Also, $a_{ui}$ would be small when $w_{uy_i}$ is small. As such, $a_{ui}$ would be high only if: (1) the feature $h_u$ focus on the core evidence for $y_i$ and it is important for the class $y_i$ according to the model $m$.

  If a feature $h_u$ is important for $y_i$ according to $m$ (i.e., with high $w_{uy_i}$), and it does not focus on the core evidence, it would lead to poor domain generalizability (DG). For better DG, we want to reduce the impact of such features. This can be achieved if we reduce the impact of all features with low $a_{ui}$. Note that by doing this we are also reducing the impact features with low $w_{uy_i}$, which should not bring about much impacts because the weights are low already.

- For each class $c$, define
$$A_{uc} = \frac{\sum_{i:y_i=c} a_{ui}}{\sum_{i:y_i=c} 1}$$

  Note that $\sum_{i:y_i=c} 1$ is the number of examples in the class $c$ in the training set. So, $A_{uc}$ is the average of the $a_{ui}$'s for the training examples in the class $c$.

  Let say that the feature units $h_u$ with $A_{uc} \geq \delta$ are *important* to the class $c$ and the others are *unimportant* to $c$. Here, $\delta$ is a hyperparameter.

- To obtain $m'$,
  - Set the weightsfor the features unimportant to the class $c$ to 0, and keep them frozen during the following retraining process. The weights in the backbone are also frozen.
  - Re-initialize the weights for the important features, and retrain them.

This method feels more “heavy-handed” than previous methods. It is expected to lead to a decrease of the iid test error, and hopefully also lead to an increase of the ood test error.