# 1 Supplementary Materials

## 1.1 Ethics Statement

The model proposed in this work, as well as the dataset introduced, involves an educational setting and thus introduces significant ethical concerns. One of the main concerns of the use of LLMs in educational settings is the potential for biases present in LLMs that negatively impact students of a specific ethnic, cultural, or racial background. These concerns are in part due to the issues of standard LLMs lacking a knowledge base to draw from, and use to inform how it should behave with specific individuals. This is a natural result of the nature of LLMs and their method of training on massive datasets, without explicitly accounting for the potential biases that exist in these datasets.

This potential concern is mitigated in this work because of the specific educational setting, in detecting phishing emails, which are designed by the original cybersecurity experts to be applicable to a wide range of end users. Additionally, our method is inspired by accounting for individual differences in students and tailoring their educational experience to their personal level of experience, potential cognitive biases, and the behavior they demonstrate during training.

However, the application of this approach outside of the setting used in this work should take care in ensuring that the method of calculating the similarity of educational examples shown to students not be biased. While this is an inherent concern in the use of LLMs in education, our proposed approach of using more individualized metrics of similarity can hopefully reduce the likelihood of LLM biases negatively impacting student education. This is because our proposed model is based on individual past experiences and biases when calculating subjective similarity.

## 1.2 IBL Model and Pseudocode

### 1.2.1 Activation

IBL models work by storing instances $i$ in memory $\mathcal{M}$, composed of utility outcomes $u_i$ and options $k$ composed of features $j$ in the set of features $\mathcal{F}$ of environmental decision alternatives. These options are observed in an order represented by the time step $t$, and the time step that an instance occurred in is given $\mathcal{T}(i)$. IBL models predict the value of options in decision-making tasks by selecting the action that maximizes the value function. In calculating this activation, the similarity between instances in memory and the current instance is represented by summing over all attributes the value $S_{ij}$, which is the similarity of attribute $j$ of instance $i$ to the current state. This gives the activation equation as:

$$A_i(t) = \ln\left(\sum_{t' \in \mathcal{T}_i(t)} (t - t')^{-d}\right) + \mu \sum_{j \in \mathcal{F}} \omega_j (S_{ij} - 1) + \sigma\xi \tag{1}$$

The parameters that are set either by modelers or set to default values are the decay parameter $d$; the mismatch penalty $\mu$; the attribute weight of each $j$ feature $\omega_j$; and the noise parameter $\sigma$. The default values for these parameters are $(d, \mu, \omega_j, \sigma) = (0.5, 1, 1, 0.25)$. The value $\xi$ is drawn from a normal distribution $\mathcal{N}(-1, 1)$ and multiplied by the noise parameter $\sigma$ to add random noise to the activation.

### 1.2.2 Probability of Retrieval

The probability of retrieval represents the probability that a single instance in memory will be retrieved when estimating the value associated with an option. To calculate this probability of retrieval, IBL models apply a weighted soft-max function onto the memory instance activation values $A_i(t)$ giving the equation:

$$P_i(t) = \frac{\exp A_i(t)/\tau}{\sum_{i' \in \mathcal{M}_k} \exp A_{i'}(t)/\tau} \tag{2}$$

The parameter that is either set by modelers or set to its default value is the temperature parameter $\tau$, which controls the uniformity of the probability distribution defined by this soft-max equation. The default value for this parameter is $\tau = \sigma\sqrt{2}$.

## 1.3 Blended Value

The blended value determines the ultimate action selected by the model and is calculated of an option $k$ at time step $t$ according to the utility outcomes $u_i$ weighted by the probability of retrieval of that instance $P_i$ and summing over all instances in memory $\mathcal{M}_k$ to give the equation:

$$V_k(t) = \sum_{i \in \mathcal{M}_k} P_i(t) u_i \tag{3}$$

These blended values are used to determine the action $a_{t+1}$ selected by the model at the next time

step.

$$a_{t+1} = \max_{k \in K} V_k(t) \tag{4}$$

In standard IBL models, this action can be used in simulations to allow the model to gain experience in a given task. In model tracing, which is used in the method proposed in this work, the memory of instances is made up of the past observations and decisions of the participant, with the action representing a prediction of their future behavior.

### 1.4 Similarity Measure

The definition of the similarity measure $S_{ij}$ is highly influential in the behavior of the IBL model, as it determines which instances from memory are drawn from to predict utility. In simple binary choice tasks without attributes. The similarity metric can be defined as the equality function $S_{ij} = 1$ if $i == j$ else 0. In more complex domains such as the phishing email identification task used in this work, one approach is to use the embeddings of emails to compare the similarity of instances, and rely on the cosine similarity metric to compute the similarity of instances in memory (**?**). The model presented in this work relies on an initial baseline similarity metric, the standard cosine similarity, to then build more individual specific metrics of similarity.

This means that, for the following IBL model pseudocode, the similarity $S_{ij}$ between the instances that correspond to emails with the embeddings $x_i$ and $x_j$ is calculated within the internal IBL cognitive model as:

$$S_{ij} = \frac{x_i^T x_j}{||x_i||||x_j||} \tag{5}$$

It is important to note that this cosine similarity is not the final resulting similarity measure of the IBIS method we propose here. Instead, this value is used by the IBL model to predict the behavior of individual participants as they engage in the task. This IBL model is used to predict an individual participants value of annotating a document $x$ with category $c$ as the value $V_c(c|x)$.

#### 1.4.1 IBL Model Pseudocode

**Input:** default utility $u_0$, a memory dictionary $\mathcal{M} = \{\}$, global counter $t = 1$, step limit $L$, a flag $delayed$ to indicate whether feedback is delayed.
**repeat**
  Initialize a counter (i.e., step) $l = 0$ and observe state $s_l$
  **while** $s_l$ *is not terminal and* $l < L$ **do**
    **Execution Loop**
      **Exploration Loop** $k \in K$ **do**
        Compute activation values $A_i(t)$ of instances $(k_i, T(i))$ by Eq: (1)
        Compute retrieval probabilities $P_i(t)$ by Eq: (2)
        Compute blended values $V_k(t)$ corresponding to $k$ by Eq: (4)
      **end**
      Choose an action $a$ corresponding to option $k_l \in \arg\max_{k \in K} V_k(t)$
    **end**
    Take action $a$, move to state $s_{l+1}$, observe $s_{l+1}$, and receive outcome $u_{l+1}$
    Store $t$ into instance corresponding to selecting $k_l$ and achieving outcome $u_{l+1}$ in $\mathcal{M}$
    If $delayed$ is true, update outcomes using a *credit assignment* mechanism
    $l \leftarrow l + 1$ and $t \leftarrow t + 1$
  **end**
**until** *task stopping condition*

**Algorithm 1:** Pseudo Code of Instance-Based Learning Process

### 1.5 Similarity Function Equations

#### 1.5.1 Cosine Similarity

Embedding cosine similarity is calculated for the currently observed embedding $x$ and the embedding held in IBL model memory $x'$ as:

$$\text{CS}(x, x') = \frac{x^T x'}{||x||||x'||} \tag{6}$$

This equation as the basis of the IBIS model similarity measure. After forming predictions of individual participant behavior, the ultimate similarity measure determined by the IBIS model is calculated using the weighted soft-max in section 1.3.5.

#### 1.5.2 Semantic Similarity

The semantic similarity function is calculated using the same equation as the cosine similarity, but using a vector of human-crafted semantic attributes that were assigned to each of the emails. So for an email with the hand-crafted vector of attributes $x = [x_0, x_1, x_2, x_3, x_4, x_5]$ the similarity to anothjer email $x'$ is given by:

$$\text{SS}(x, x') = \frac{x^T x'}{||x||||x'||} \tag{7}$$

#### 1.5.3 Weighted Cosine Similarity

The weighted cosine similarity functions in the same way as the cosine equation, with the addition of the weight $w$ that is applied to both the observed email embedding $x$ and the embedding held in memory $x'$.

$$\text{CS}_w(x, x', W) = \frac{(Wx)^T(Wx')}{||Wx||||Wx'||} \tag{8}$$

This weight is trained to maximize the similarity between the embedding similarities and data from the human subjective similarity measure.

#### 1.5.4 Pruned Cosine Similarity

The weighted cosine similarity functions in the same way as the cosine equation, with the addition of the mask $M$ that is applied to both the observed email embedding $x$ and the embedding held in memory $x'$. This mask is similar to the weight in the weighted cosine similarity equation, but it only uses 1 or 0 values.

$$\text{CS}_w(x, x', M) = \frac{(Mx)^T(Mx')}{||Mx||||Mx'||} \tag{9}$$

Another difference between this method and the weighted cosine similarity is that it is learned iteratively by masking additional embedding values to increase the similarity between embedding measures and the human subjective similarity metric.

#### 1.5.5 Ensemble Similarity

The ensemble similarity metric is a weighted sum of all previous similarity metrics using a weight vector $w = [w_1, ..., w_n]$ that is formed to minimize the difference between ensemble similarity and the average human subjective similarity.

#### 1.5.6 IBIS Similarity

The proposed IBIS similarity measure works by first building the IBL model using email embeddings that calculate similarity according to the cosine similarity measure. Then this model traces the behavior of an individual and predicts the values associated with their categorization for a document $V_k(c|x)$, this allows for the comparison of similarity to another document $x'$ using the same value $V_k(c|x')$ and the soft-max equation:

$$IBIS(x, x') = \frac{V_k(c|x)V_k(c|x')}{\sum_{c' \in C} V_k(c'|x) \sum_{c' \in C} V_k(c'|x')} \tag{10}$$

This value can be calculated for each individual participant only taking into account their own decision making, but applied onto documents that were not observed by that participant.

### 1.6 IBIS Algorithm Psudocode

**Input:** default utility $u_0$, a memory dictionary $\mathcal{M} = \{\}$, global counter $t = 1$, step limit $L$. Dataset of stimuli $D$
**repeat**
  Initialize a counter (i.e., step) $l = 0$ and observe state $s_l$
  **while** $s_l$ *is not terminal and* $l < L$ **do**
    **Execution Loop**
      **Exploration Loop** $k \in K$ **do**
        Compute $A_i(t)$ by Eq: (1)
        Compute $P_i(t)$ by Eq: (2)
        Compute $V_k(t)$ by Eq: (4)
      **end**
      Update similarity by Eq: (10) using each data point in $D$
      Predict student action $a$ by $k_l \in \arg\max_{k \in K} V_k(t)$
    **end**
    Observe student action $a$, observe $s_{l+1}$, and student feedback outcome $u_{l+1}$
    Store $t$ instance in $\mathcal{M}$
  **end**
**until** *task stopping condition*

**Algorithm 2:** Pseudo Code of Instance-Based Learning Cosine Similarity Update