A THEORETICAL RESULT

Consider the following abstracted and idealized version of the contrastive learning game. An encoder receives an input, and communicates features in that input via bits. A distinguisher has to identify the original input from a set of k (distractor) inputs, based on the communicated features. The encoder and distinguisher win if the distinguisher correctly identifies the original input. The encoder and distinguisher need to decide on a communication protocol before playing the game. Each bit corresponds to one feature. The encoder sends a 1 if a given feature is present and a 0 otherwise.

The question we're answering in this section is: what is the optimal feature occurrence (or bit entropy) for a feature when the encoder can choose b bits, and the distinguisher has to choose between k inputs.

Below we calculate that the optimal strategy is to use l independent features that are each present in exactly half of the images. The chance of the receiver picking out the right image depends on k.

For this calculation we will assume the encoder can choose $b = 2$ bits, i.e. can communicate two features x and y. Let f_x and f_y be the frequency of respectively feature x and feature y in the dataset. To answer the question we will calculate the values of f_x and f_y that maximize the chance of winning.

Let c_x be the random variable that represents: the correct input has feature x, and c_y the variable that represents: the correct input has feature y. We assume that these variables are independent. Let v be the random variable that represents the number of inputs in the set of k inputs that the distinguisher gets to see, that have both feature x and feature y.

$$\begin{aligned} P(\text{win}) = \quad & P(\text{win}|c_x, c_y) \cdot P(c_x, c_y) \\ & + P(\text{win}|c_x, c_{\neg y}) \cdot P(c_x, c_{\neg y}) \\ & + P(\text{win}|c_{\neg x}, c_y) \cdot P(c_{\neg x}, c_y) \\ & + P(\text{win}|c_{\neg x}, c_{\neg y}) \cdot P(c_{\neg x}, c_{\neg y}) \end{aligned}$$

Note that $P(c_x, c_y) = f_x \cdot f_y$. Below we calculate that

$$P(\text{win}|c_x, c_y) = \Sigma_{v=1}^{k} \frac{1}{v} \cdot (f_x f_y)^{v-1} \cdot (1 - f_x f_y)^{k-v} \cdot \binom{k-1}{v-1}.$$

To do so we introduce one more helper variable $\tilde{v}$ which represents the number of inputs in the set of k inputs that the distinguisher gets to see, that have both feature x and feature y, but excluding the correct input.

We now calculate

$$\begin{aligned} P(\text{win}|c_x, c_y) &= \Sigma_{v=1}^{k} P(\text{win}|c_x, c_y, V = v) \cdot P(\tilde{V} = v - 1|c_x, c_y) \\ &= \Sigma_{v=1}^{k} P(\text{win}|c_x, c_y, V = v) \cdot P(\tilde{V} = v - 1) \end{aligned}$$

Note that $P(\text{win}|c_x, c_y, V = v) = \frac{1}{v}$ and

$$P(\tilde{V} = v - 1) = (f_x f_y)^{v-1} \cdot (1 - f_x f_y)^{k-1-(v-1)} \cdot \binom{k-1}{v-1}$$

Hence

$$P(\text{win}|c_x, c_y) = \Sigma_{v=1}^{k} \frac{1}{v} \cdot (f_x f_y)^{v-1} \cdot (1 - f_x f_y)^{k-v} \cdot \binom{k-1}{v-1}.$$

Applying the Bionomial theorem gives us the following equality

$$\begin{aligned}
P(\text{win}|c_x, c_y)P(c_x, c_y) &= f_x f_y \cdot \Sigma_{v=1}^{k} \frac{1}{v} \cdot (f_x f_y)^{v-1} \cdot (1 - f_x f_y)^{k-v} \cdot \binom{k-1}{v-1} \\
&= \Sigma_{v=1}^{k} \frac{1}{v} \cdot (f_x f_y)^{v} \cdot (1 - f_x f_y)^{k-v} \cdot \binom{k-1}{v-1} \\
&= \frac{1}{k} \Sigma_{v=1}^{k} (f_x f_y)^{v} \cdot (1 - f_x f_y)^{k-v} \cdot \binom{k}{v} \\
&= \frac{1}{k} \left((f_x f_y + 1 - f_x f_y)^k - (1 - f_x f_y)^k \right) \\
&= \frac{1}{k} \left(1 - (1 - f_x f_y)^k \right) \\
&= \frac{1}{k} - \frac{1}{k} (1 - f_x f_y)^k
\end{aligned}$$

We can write similar equations for $c_{\neg x}$ and $c_{\neg y}$ and combining them results in

$$\begin{aligned}
P(\text{win}) = \frac{4}{k} - \frac{1}{k} \big(&(1 - f_x \cdot f_y)^k \\
&+ (1 - f_x \cdot (1 - f_y))^k \\
&+ (1 - (1 - f_x) \cdot f_y)^k \\
&+ (1 - (1 - f_x) \cdot (1 - f_y))^k \big).
\end{aligned}$$

More generally, for arbitrary number of bits b and feature frequencies $f_1, \ldots, f_b$ we find

$$P(\text{win}) = \frac{2^b}{k} - \frac{1}{k} \left((1 - f_1 \cdots f_b)^k + (1 - (1 - f_1) f_2 \cdots f_b)^k + \ldots + (1 - (1 - f_1) \cdots (1 - f_b))^k \right)$$

The derivative of $P(\text{win})$ with respect to f_1 is

$$\begin{aligned}
\frac{\partial P(\text{win})}{\partial f_1} = \quad & f_2 \cdots f_b (1 - f_1 \cdots f_b)^{k-1} \\
& - f_2 \cdots f_b (1 - (1 - f_1) f_2 \cdots f_b)^{k-1} \\
& + \ldots \\
& - (1 - f_2) \cdots (1 - f_b)(1 - (1 - f_1) \cdots (1 - f_b))^{k-1}
\end{aligned}$$

When $f_1 = 0.5$ the components with a factor of f_1 compensate for the ones with a factor of $(1 - f_1)$, and so the derivative is 0 for $f_1 = 0.5$. Deriving with respect to other feature values gives analogues results. That is, one optimal feature occurrence value for maximizing $P(\text{win})$ is 0.5.

B Training Methodology

In order to prevent overfitting and the representation of 'trivial features' (e.g. specific pixel values) in the representations, during training we use a stack of image augmentation layers independently applied prior to each image encoder. This involves a random rotation of up to 0.1 radians, a random contrast shift of up to 10%, a random translation of up to 10% along both axes, and a random zoom of up to 10% (all with a nearest-neighbour filling of blank pixels).

The models were optimized using Adam (Kingma & Ba, 2015) with a learning rate of 0.001. The batch size used for training was dependent on the number of distractors, and each epoch iterated through the entire training dataset. See Table 1 for the full breakdown of test accuracy values for trained models, i.e. the mean and standard deviations for the proportion of occasions where the distinguisher was correctly able to identifier x^* by using r.

All of the code was implemented with Tensorflow 2 (Abadi et al., 2015) and datasets were pulled from Tensorflow Datasets[4] (TFDS) (TF Devs, 2022). CIFAR-10 was split into the default TFDS

[4]The license for these datasets can be found at: `https://github.com/tensorflow/datasets/blob/master/LICENSE`

training and test sets (50,000 training images and 10,000 test images). Training and analysis were performed with an NVIDIA RTX 3090 GPU.

We trained 54 independent encoder-distinguisher pairs[5] for 10 epochs on CIFAR-10 and removed models that did not converge (as defined by not reaching an 80% drop in loss), resulting in 51 trained models (taken as the best performing checkpoint). Models were trained with varying combinations of representation lengths and number of distractors: $(|r|, k) \in \{64, 128, 256, 512\} \times \{3, 5, 10, 20\}$. We also trained models with representation lengths 8, 16 and 32, visualizations of which can be found in Figure 9, which we discarded because their bit entropies were too homogeneous to meaningfully study the effect of masking out low versus high entropy bits.

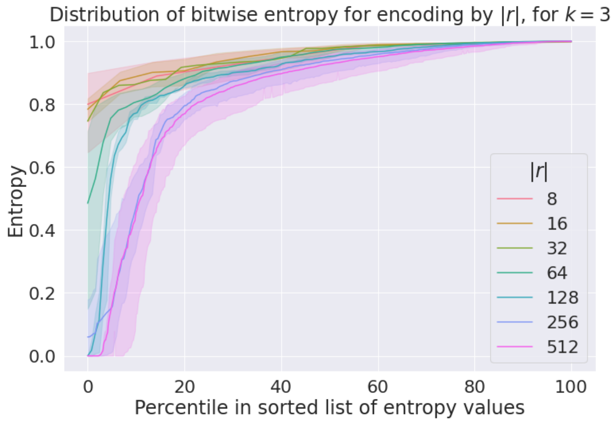

Figure 9: The x-axis represents entropy percentile of bits in the representation. The y-axis shows the entropy values of bits (measured on CIFAR-10). In other words, we take the list of bits and sort them by entropy, and then plot the sorted line as percentiles in order to compare the distributions of different lengths. The translucent regions show the error bars from various training runs. We can see that for lower $|r|$ values, the entropy distributions do not tend to go below 0.8.

C Experiments

The code for the experiments can be found at the following repository: **[URL removed to preserve anonymity]**

C.1 Overfitting Analysis

In Figure 10 we see that the test and training accuracies are very similar (with the test accuracy even being slightly higher) and so no overfitting has happened.

[5] A sweep of 3 runs for each pair of $(|r|, k)$ plus 6 initial separate runs.

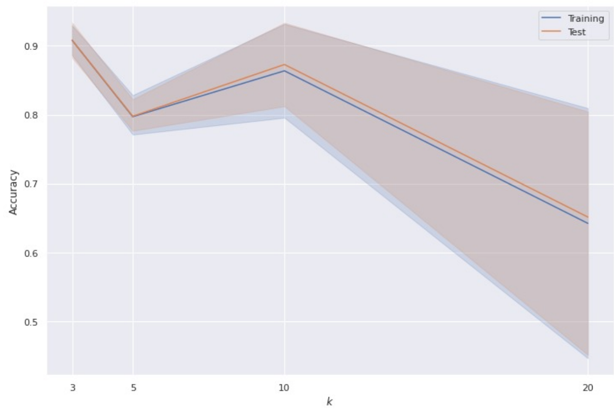

Figure 10: For different values of k the blue line shows the training accuracy and the orange line shows the test accuracy.

C.2 OOD Accuracy Change for all Masking Proportions and all values of k and $|r|$

Figure 11 shows the OOD accuracies for each dataset (using the data of all the values of k and all the analysed representation lengths). Figure 12 shows the accuracies for each dataset and each value of k. Figure 13 shows the accuracies for each dataset and each representation length $|r|$.

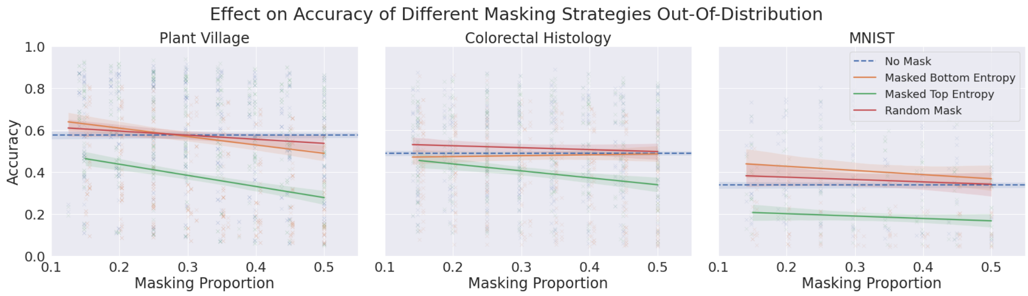

Figure 11: The y-axis represents the accuracy and the x-axis the masking proportion. Different masking strategies are represented by different colors.

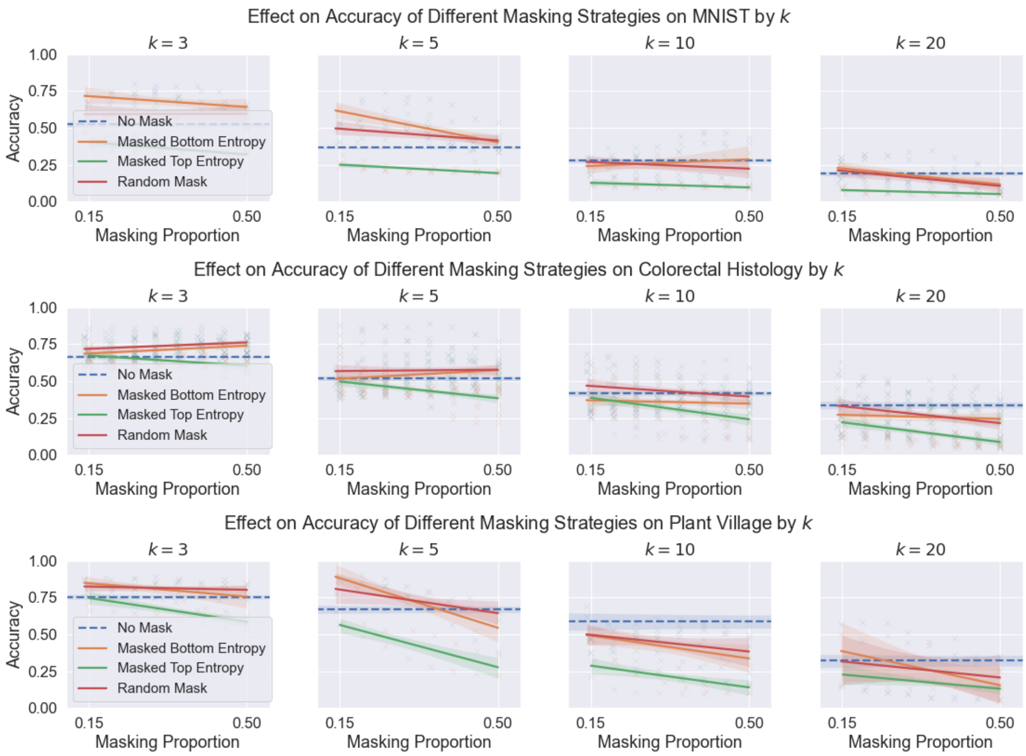

Figure 12: The y-axis shows the accuracy and the x-axis shows different masking proportions. Masking strategies are indicated by color.

C.3 OOD Mean Accuracy Change From Masks

The tables in this section are the same as Table 2 in Section 4.2, except separated by different values of k. Figure 14 is a visualisation of the data along with the 'distance out-of-distribution' for each k value.

Table 3: Mean change in accuracy (in percentage points) ($k = 3$, $p_{mask} \approx 0.25$)

Dataset Strategy	CIFAR-10	Colorectal Histology	MNIST	Plant Village
Masked Bottom Entropy	2.8 ± 1.4	2.8 ± 2.3	16.6 ± 7.5	8.1 ± 5.5
Masked Top Entropy	4.0 ± 1.4	-0.4 ± 4.3	-12.7 ± 5.2	-2.9 ± 1.5
Random Mask	4.0 ± 1.4	5.8 ± 3.0	6.5 ± 3.9	6.2 ± 2.5

Table 4: Mean change in accuracy (in percentage points) ($k = 5$, $p_{mask} \approx 0.25$)

Dataset Strategy	CIFAR-10	Colorectal Histology	MNIST	Plant Village
Masked Bottom Entropy	4.5 ± 3.5	0.7 ± 7.6	22.2 ± 8.0	13.8 ± 9.1
Masked Top Entropy	2.7 ± 6.3	-3.4 ± 8.5	-12.3 ± 4.6	-13.3 ± 15.0
Random Mask	5.8 ± 3.0	5.1 ± 4.9	9.5 ± 7.4	9.8 ± 4.9

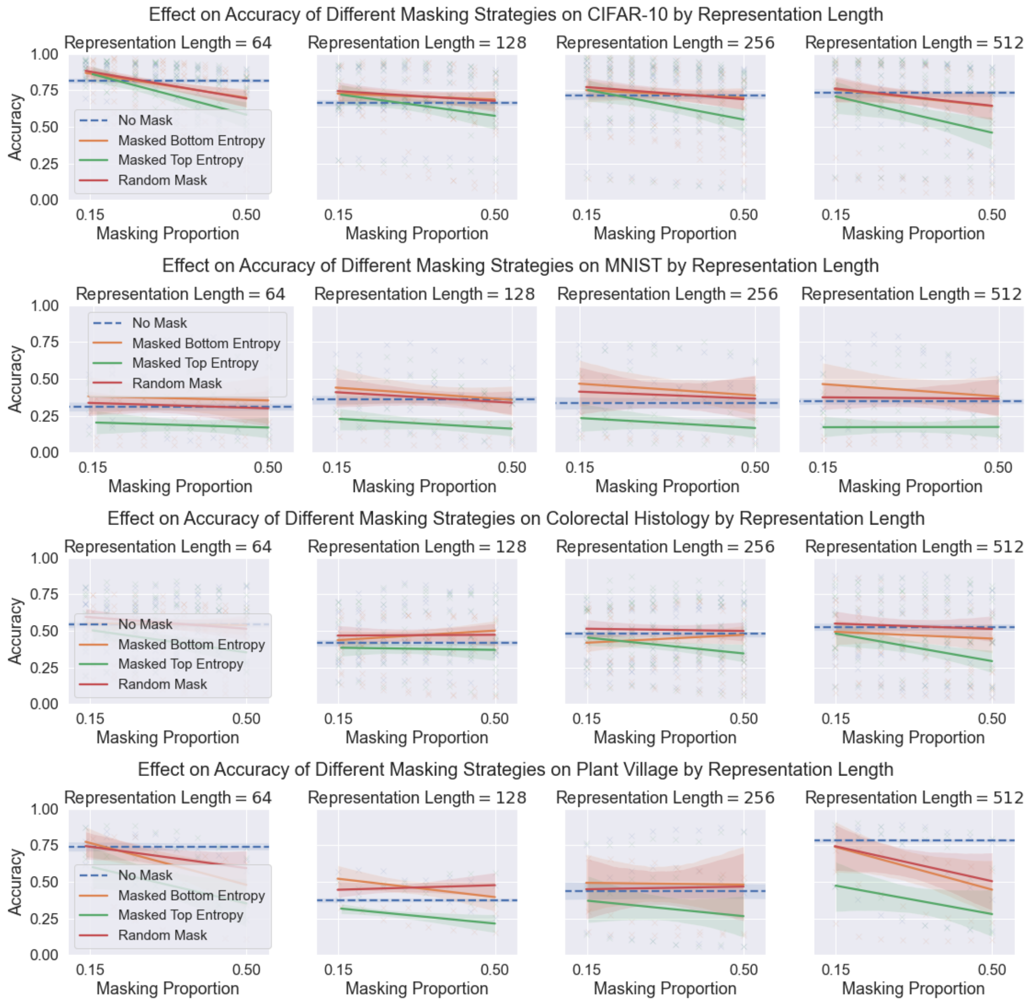

Figure 13: The y-axis shows the accuracy and the x-axis shows different masking proportions. Masking strategies are indicated by color.

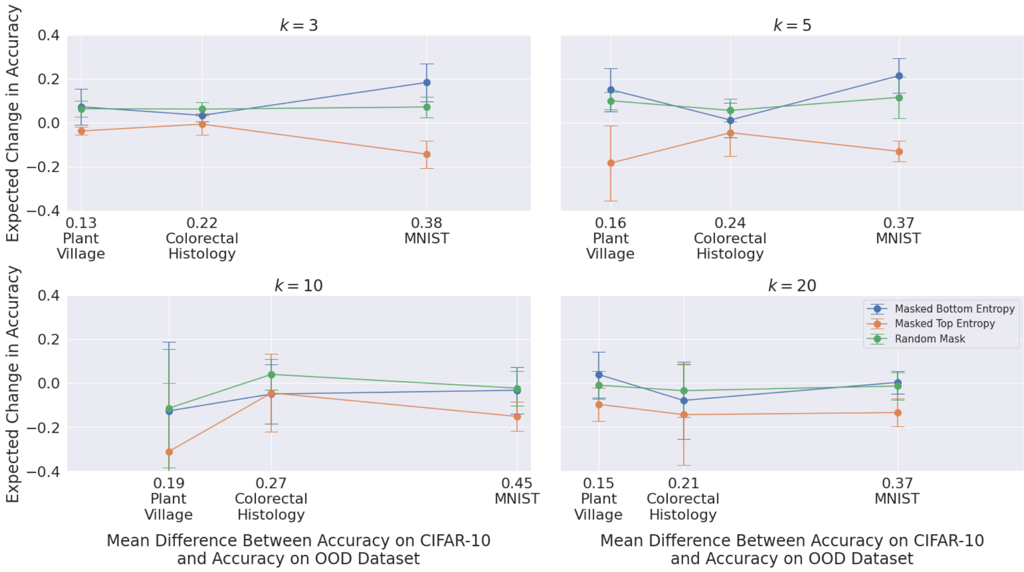

Figure 14: Mean change in accuracy when apply each masking strategy to each model varying by k on the OOD datasets. Standard deviations denoted with error bars.

Table 5: Mean change in accuracy (in percentage points) ($k = 10$, $p_{mask} \approx 0.25$)

Dataset Strategy	CIFAR-10	Colorectal Histology	MNIST	Plant Village
Masked Bottom Entropy	2.2 ± 5.8	-4.0 ± 12.2	-3.3 ± 9.2	-3.0 ± 24.0
Masked Top Entropy	-3.9 ± 19.0	-4.8 ± 17.3	-15.4 ± 7.4	-21.3 ± 26.5
Random Mask	3.5 ± 9.1	3.9 ± 5.8	-1.1 ± 5.5	-3.1 ± 20.7

Table 6: Mean change in accuracy (in percentage points) ($k = 20$, $p_{mask} \approx 0.25$)

Dataset Strategy	CIFAR-10	Colorectal Histology	MNIST	Plant Village
Masked Bottom Entropy	-0.6 ± 5.7	-7.3 ± 17.5	0.7 ± 4.9	-2.4 ± 17.2
Masked Top Entropy	-8.6 ± 20.7	-13.6 ± 22.2	-12.9 ± 6.4	-12.0 ± 12.4
Random Mask	-0.5 ± 12.0	-2.6 ± 10.3	-0.6 ± 6.1	-0.8 ± 8.2