$$
\begin{aligned}
\boldsymbol{H}_{L-F}^{w} &= \text{Encoder}_{w}(\boldsymbol{H}_{0}^{w}) \\
\boldsymbol{H}_{L-F}^{v} &= \text{Encoder}_{v}(\boldsymbol{H}_{0}^{v}) \\
\boldsymbol{H}_{L}^{w} &= \text{Encoder}_{s}(\boldsymbol{H}_{L-F}^{w}) \\
\boldsymbol{H}_{L}^{v} &= \text{Encoder}_{s}(\boldsymbol{H}_{L-F}^{v}) \\
\boldsymbol{H}_{l}^{w} &= \left[\boldsymbol{w}_{l}^{\texttt{[T\_CLS]}}, \boldsymbol{w}_{l}^{1}, ..., \boldsymbol{w}_{l}^{M}, \boldsymbol{w}_{l}^{\texttt{[T\_SEP]}}\right] \\
\boldsymbol{H}_{l}^{v} &= \left[\boldsymbol{v}_{l}^{\texttt{[I\_CLS]}}, \boldsymbol{v}_{l}^{1}, ..., \boldsymbol{v}_{l}^{N}\right]
\end{aligned}
\tag{1}
$$

- with $l \in \{1, ..., L-F, ..., L\}$
- we define $\boldsymbol{H}_{L}^{w}$ as the final output of the student model for the caption, and $\boldsymbol{H}_{L}^{v}$ as the final output of the student model for the image, with $\boldsymbol{H}_{L}^{w} \in \mathbb{R}^{(M+2)\times D}$ and $\boldsymbol{H}_{L}^{v} \in \mathbb{R}^{(N+1)\times D}$

Image-Text Matching with Feature Fusion

Up to this point, we only adapted the Image-Text Contrast (ITC) from VLMo to our model. Notably, we did not employ Masked Language Modeling (MLM) because we leverage Knowledge Distillation to learn the features of the teacher model, which are the representations the model returns. We also avoided using Image-Text Matching (ITM) due to the nature of the CLS token in the shared Transformer block of our adaptation, which represents image and text independently (as in TODO: cite SHRe), rather than an image-text pair (image and text together). This configuration prevents us from directly passing our representation to a classification head for ITM, since a combination of image and text is necessary to predict a match.

In VLMo, as described in Equation 2, the final output demonstrates that text tokens and image patches are concatenated, allowing text tokens to attend to image patches through Self-Attention (TODO: Cite transformer paper), and vice versa. Therefore, the global image-text representation, $\boldsymbol{w}_{L}^{\texttt{[T\_CLS]}}$, contains information from both the image and text. Consequently, a classification head can utilize this representation to infer if an image-text pair matches.

$$
\boldsymbol{H}_{L}^{wv} = \left[\boldsymbol{w}_{L}^{\texttt{[T\_CLS]}}, \boldsymbol{w}_{L}^{1}, ..., \boldsymbol{w}_{L}^{M}, \boldsymbol{w}_{L}^{\texttt{[T\_SEP]}}, \boldsymbol{v}_{L}^{\texttt{[I\_CLS]}}, \boldsymbol{v}_{L}^{1}, ..., \boldsymbol{v}_{L}^{N}\right] \tag{2}
$$

In our approach however, $\boldsymbol{w}_{L}^{\texttt{[T\_CLS]}}$ and $\boldsymbol{v}_{L}^{\texttt{[I\_CLS]}}$ contain only the information of the image and text, respectively (see Equation 3).

$$
\boldsymbol{H}_{L}^{w} = \left[\boldsymbol{w}_{L}^{\texttt{[T\_CLS]}}, \boldsymbol{w}_{L}^{1}, ..., \boldsymbol{w}_{L}^{M}, \boldsymbol{w}_{L}^{\texttt{[T\_SEP]}}\right], \boldsymbol{H}_{L}^{v} = \left[\boldsymbol{v}_{L}^{\texttt{[I\_CLS]}}, \boldsymbol{v}_{L}^{1}, ..., \boldsymbol{v}_{L}^{N}\right] \tag{3}
$$

Although it is possible to compute the cosine similarity between the global text representation $\boldsymbol{w}_{L}^{\texttt{[T\_CLS]}}$ and the image representation $\boldsymbol{v}_{L}^{\texttt{[I\_CLS]}}$, this is already performed by the contrastive loss (explained in TODO: cite cotrastive loss section). Instead, we propose an approach we call feature fusion. Inspired by the method of concatenating image and text timesteps (as implemented in VLMo, BEiT-3, and FLAVA) to generate a representation of an image-text pair, we concatenate the global representations of the image and text to form an image-text representation. This combined representation is then passed to a classification head to predict whether the image-text pair matches.

$$
\begin{aligned}
\boldsymbol{u} &= \boldsymbol{w}_{L}^{\texttt{[I\_CLS]}} \parallel \boldsymbol{v}_{L}^{\texttt{[T\_CLS]}} \\
\boldsymbol{p} &= [p_0, p_1] = \text{Head}_{\text{ITM}}(\boldsymbol{u})
\end{aligned}
\tag{4}
$$

Here, $\boldsymbol{w}_{L}^{\texttt{[T\_CLS]}}$ and $\boldsymbol{v}_{L}^{\texttt{[I\_CLS]}}$ together form the input $\boldsymbol{u}$ to the image-text matching head Head_{ITM}, where $\boldsymbol{u} \in \mathbb{R}^{2\times D}$. The output $\boldsymbol{p} \in \mathbb{R}^{2}$ contains the logits indicating whether the image and text

match (p_1) or not (p_0), they are normalized using softmax to obtain probabilities before computing the loss.

Because the contrastive loss is applied on the [I_CLS]/[T_CLS] token (depending which modality was the input) of both the first and last MLP layer of the shared Transformer block, we also introduce ITM for both layers. This means that we have two classification heads, Interm-Head$_{\text{ITM}}$ and Head$_{\text{ITM}}$, one for the first and one for the last layer, respectively.

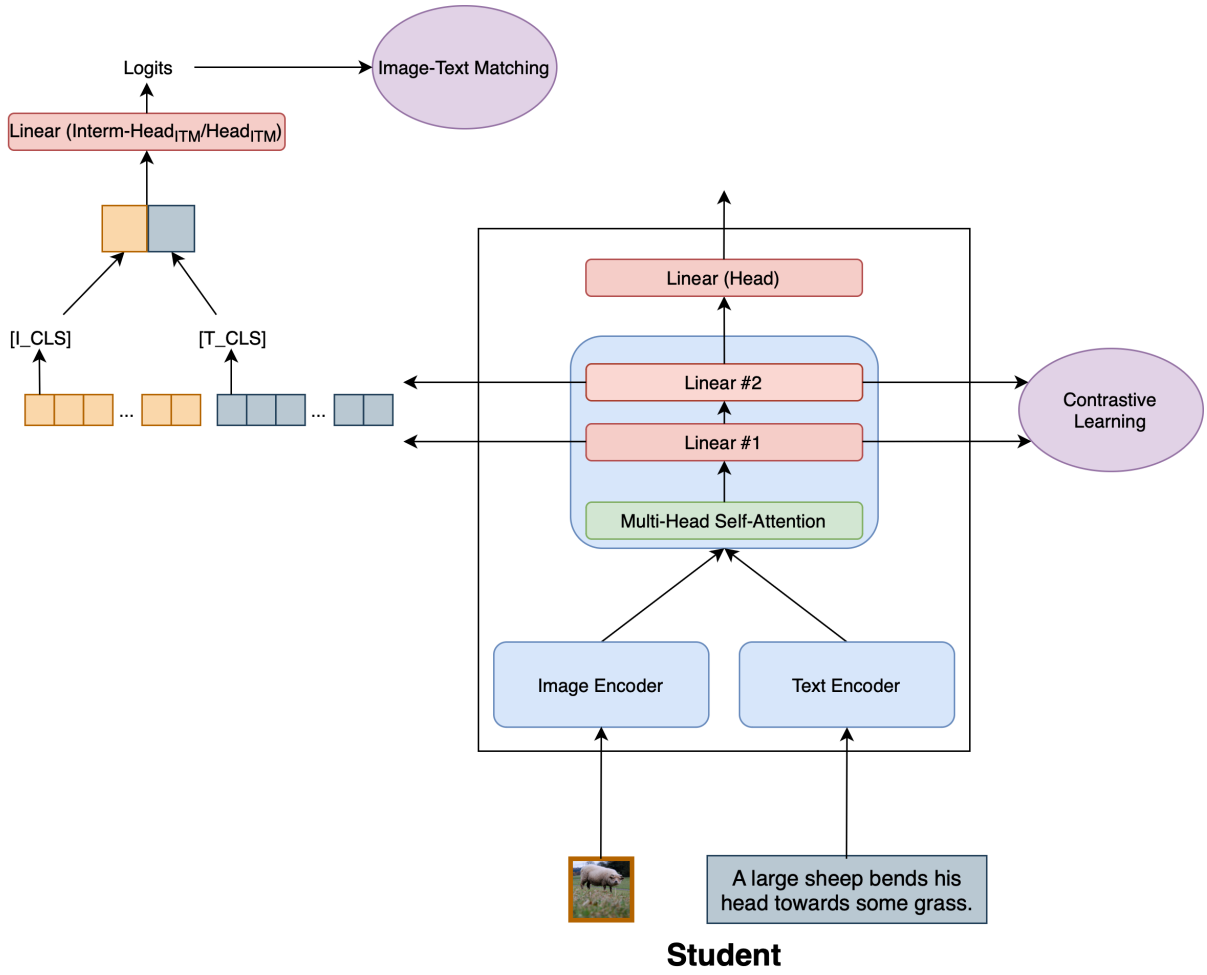

Figure 1: Image-Text Matching (ITM) with Feature Fusion in Sx3HRe. The global representations of the image and text are concatenated to form a representation of the image-text pair, which is then passed to a classification head to predict whether the image-text pair matches. The figure shows one classification head to keep the figure concise. The head is either Interm-Head$_{\text{ITM}}$, for outputs of the first linear layer (Linear #1), or Head$_{\text{ITM}}$, for outputs of the second linear layer (Linear #2). The Knowledge Distillation loss (MSE), and the teacher, are omitted for simplicity. The image-text example is taken from the COCO train set (TODO: cite coco).

Before applying binary cross-entropy (Equation 5) to calculate the ITM loss $\mathcal{L}_{\text{ITM}}$ (Equation 6), $\boldsymbol{p}$ is softmax-normalized. The labels are defined such that $y_0 = 1$ and $y_1 = 0$ if the image-text pair does not match, and $y_0 = 0$ with $y_1 = 1$ if they do. Since ITM is applied to two layers simultaneously, the total ITM loss is the mean of the losses of both layers.

$$\text{CE}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = -\sum_{i=0}^{1} y_i \log(\hat{y}_i), y_i \in \{0, 1\} \tag{5}$$

$$\mathcal{L}_{\text{ITM}} = \frac{1}{2} * (\text{CE}(\boldsymbol{p}, \boldsymbol{y}) + \text{CE}(\boldsymbol{p}, \boldsymbol{y})) \tag{6}$$

In this initial experiment, Head_{ITM} and $\text{Interm-Head}_{\text{ITM}}$ are single fully connected layers. The resulting loss for the student model is similar to that of VLMo and ALBEF:

$$\mathcal{L}_{\text{VLMo}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{ITC}} + \mathcal{L}_{\text{ITM}} \tag{7}$$

The difference lies in the MLM loss, which is replaced by the Knowledge Distillation loss $\mathcal{L}_{\text{KD}}$:

$$\mathcal{L}_{\text{Sx3HRe}} = \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{ITC}} + \mathcal{L}_{\text{ITM}} \tag{8}$$

Image-Text Matching requires both positive and negative examples. An intuitive approach to generating negative examples is to sample a random text for a given image from the current batch, and vice versa. However, this method might make the task too easy, as negative example will be completely unrelated to the image in most cases.

To address this issue, we follow VLMo and ALBEF and employ hard-negative mining, which involves sampling difficult negative examples that are hard to distinguish from positive examples. ALBEF and VLMo utilize the cosine similarities of the current batch from contrastive learning to achieve this. A negative image-text pair for a given image is generated by sampling a text/caption using a multinomial distribution, where the cosine similarities serve as probabilities, and the actual text/caption of the image has a probability of 0. This means that captions with a high cosine similarity to the image are more likely to be sampled as negative examples. The same process is applied to generate negative examples for captions.

Given a batch size N, N positive examples (the actual image-text pairs), and $2N$ negative examples are generated:

- Fuse representations of actual image-text pairs to obtain N positive examples.
- Fuse representations of each image in the batch with its hard-negative caption to get N negative examples.
- Fuse representations of each caption in the batch with its hard-negative image to produce another N negative examples.

In all cases, the joined representation $\boldsymbol{u}$ (as described in Equation 4) begins with the image representation, followed by the text representation.

The remainder of the training setup remains unchanged, the classification heads adds just 6,148 parameters (3,074 each) to the model, which is negligible compared to the total number of trainable parameters, which now stands at 132 million.

The results (Figure 2) show a continuous training accuracy of 66.67%, which remains constant throughout the first epoch with no deviation from this value. This suggests that our approach is not effective. We suspect this outcome arises because the classification head fails to learn anything meaningful from the fused representations, leading it to predict the most frequent class in the batch to somehow minimize the loss.

In this setting, the most frequent class corresponds to the scenario where the image-text pair does not match, as each batch contains $2N$ negative examples and only N positive examples. Consequently, 66.67% of the fused representations in a batch are negative examples. If the classification head predicts that all examples are negative, it will achieve an accuracy of exactly 66.67%.

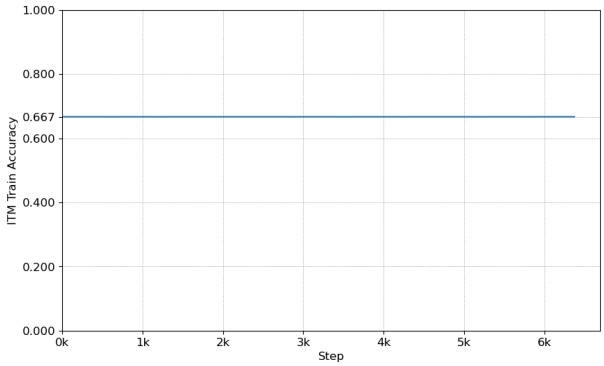

Figure 2: Training accuracy of the student model for Image-Text Matching (ITM) in the first epoch. The accuracy does not change from the initial value of 66.67%, and always predicts that image-text pairs do not match.

As this result gives no reason to continue the training, we stop it after the first epoch.

We hypothesize that a single linear layer may be insufficient for learning the matching between image and text representations. Our setting differs from that in VLMo, where the CLS token's representation already encapsulates information from both the image and text. In our case, we fuse separate embeddings, and the resulting representation by itself lacks semantic meaning. Consequently, a simple classification head is unable to extract any meaningful information from this representation.

To address this, we replace the simple classification heads with feed-forward networks (FFN) of the same architecture as those used in Transformer blocks, i.e. two linear layers. The first layer expands the dimensionality of the fused representation from a 1536-dimensional vector to a 6144-dimensional vector. The second layer reduces the dimensionality to 2, corresponding to the number of classes. This modification adds 6 million parameters to the model, which we deem acceptable if the results show a significant improvement.

As shown in Figure 3, the ITM heads now appear to learn from the fused representations. Unfortunately, this fact is neither reflected in the zero-shot performance on ImageNet-1K, nor in the retrieval performance on MSCOCO and Flickr30K, illustrated in Table 1.

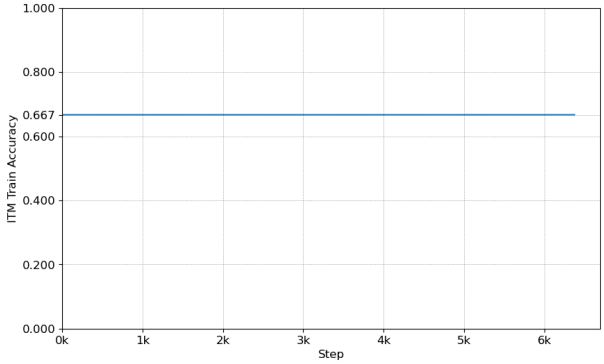

Figure 3: Training accuracy of the student model for Image-Text Matching (ITM) in the first epoch. The accuracy does not change from the initial value of 66.67%, and always predicts that image-text pairs do not match.

Model	ImageNet-1k Accuracy	Retrieval MSCOCO	Retrieval Flickr30K	#Params
CLIP (TODO: cite)	**72.6**	66.73	**90.1**	428M
Sx3HRe$_{\text{MSE Head}}$	26.1	**66.3**	42.4	**132M**
Sx3HRe$_{\text{MSE Head + ITM}}$	25.5	65.7	41.6	138M

Table 1: Comparison of Sx3HRe with other approaches when adding ITM. The results show that our fusion-based approach does not improve the performance of the model. ImageNet-1k accuracy is based on CLIPs zero-shot transfer, while retrieval performance is the mean of R@1, R@5, and R@10 on the respective datasets.