1 Architectural Differences

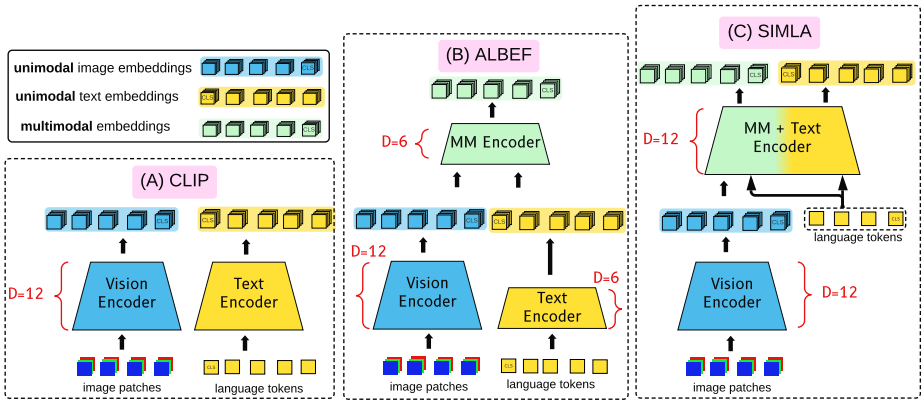

Fig. 1. Architectural differences between CLIP (left panel), ALBEF (middle panel) and the proposed model (right panel).

1.1 CLIP vs. SIMLA

CLIP[43] has a symmetric dual-encoder architecture which is designed for global alignment between unimodal text and image representations. Each encoder is a 12-layer transformer encoder dedicated to a single modality. SIMLA has a single-stream architecture designed for alignment at multiple levels. The primary architectural differences between CLIP and SIMLA are:

1. SIMLA includes a multimodal encoder with cross-attention that enables alignment between patch-level image regions and the caption.
2. SIMLA adds additional training tasks, taking advantage of the multimodal encoder to align on multiple levels.
3. SIMLA's text encoder is dual-purpose: it is used as both a multimodal encoder and text encoder by sharing weights.

1.2 ALBEF vs. SIMLA

ALBEF[24] can be seen as an asymmetric variant of CLIP, with a transformer-based multimodal encoder atop the unimodal text and image encoders for stronger fusion. Furthermore, ALBEF aligns the unimodal text and image representations before fusion within the multimodal encoder. The primary architectural differences between ALBEF [24] and SIMLA are:

1. SIMLA's multimodal encoder can fuse raw, unaligned language tokens with image patch embeddings from the visual encoder. In contrast, ALBEF's multimodal encoder requires already aligned vision/language *features* as input for fusion.
2. SIMLA reuses the multimodal encoder as a text encoder by sharing weights.
3. SIMLA's multimodal encoder is capable of using both image patches and language tokens as queries in the attention layers due to the cross-modality reconstruction task. ALBEF's multimodal encoder can only use language tokens as queries in the attention layers.
4. SIMLA has twice the depth of multimodal fusion (12 layers vs 6 layers) with the same number of parameters.

1.3 General Similarities and Differences

ALBEF, CLIP, and SIMLA all have the same number of transformer encoder layers (24), though they are distributed differently. Specifically, all of CLIP's layers are dedicated to unimodal representation learning (image / text encoders), with no fusion layers. ALBEF incorporates a multimodal encoder (fusion layers), but reduces the amount of layers dedicated to unimodal representation learning (text / image encoders). SIMLA incorporates a multimodal encoder (fusion layers), but avoids the need to reduce the number of layers dedicated to unimodal representation learning through weight sharing between the text encoder and multimodal encoder. We take advantage of the observation [33,35,52] that pretrained language models have substantial capability for reuse and novel tasks, and reuse the text encoder as a multimodal encoder by adding cross-attention layers to the language model.

2 More Fine-Grained Alignment Examples

In Figure 2, we present examples of the image encoder's ability to ground image regions to language. The concept head atop the image encoder's `[CLS]` token is a linear classifier that predicts the presence or absence of tokens in the caption, based only on the image content. We apply Grad-CAM [45] to show what image regions the image encoder is looking at when it predicts the presence of a token. As visible in the Grad-CAM visualizations of Figure 2, the image encoder itself is capable of rudimentary natural language grounding.

3 Pseudolabel Extraction

The pseudolabel supervision loss is designed to train the image encoder's representation to explicitly encode the presence of crossmodal concepts. While "concept" is a broad term, we use it in a narrower sense: to denote object-level semantic regions of images or text. A subset of language tokens can clearly be used to denote objects (e.g. 'cow', 'chair', 'horse'). Other language tokens have

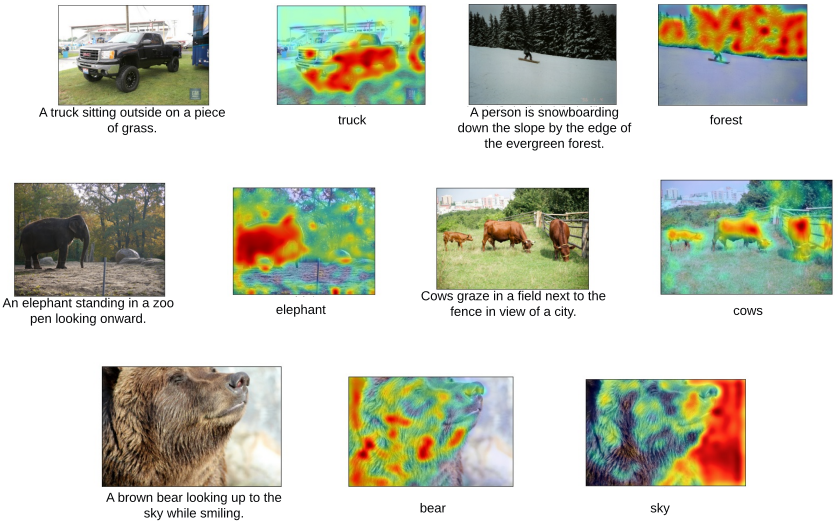

Fig. 2. Grad-CAM of the image encoder through the concept prediction head.

no obvious visual counterpart (e.g. 'forever', 'famine'). Based on this intuition, we use the language tokens present in a caption as labels for the associated image. However, only a subset of these labels will correspond to cross-modality concepts which can be represented both visually and textually. To select the subset of language tokens in a caption corresponding to cross-modal concepts, we use the attention weights of the last layer of the multimodal encoder. Let $\{\texttt{[cls]}, t_1, ..., t_N\}$ be the input language tokens, and let $(\{\vec{v}_{\text{cls}}, \vec{v}_1, ..., \vec{v}_N\})$ be the sequence of image patch embeddings produced by the image encoder for an image-text pair. We perform a forward pass through the multimodal encoder E_{mm} using the language tokens as the queries[1] and the image patches as the keys and values. Using the standard formulation of cross-attention [33,53] in Equation (1),

$$\text{Cross-Attention}(Q_t, K_i, V_i) = \text{softmax}\left(\frac{Q_t K_i^T}{\sqrt{d_k}}\right) V_i \tag{1}$$

where Q_t is query embedding sequence of the language tokens, V_i is the value embedding sequence of the image patches, and K_i is the key embedding sequence of the image patches, we compute a series of multimodal embeddings $\{\vec{m}_{\text{cls}}, \vec{m}_1, ..., \vec{m}_N\}$ having the same length as the sequence of language input tokens $\{\texttt{[cls]}, t_1, ..., t_N\}$. Next, we apply *self-attention*

$$\text{Self-Attention}(Q_{mm}, K_{mm}, V_{mm}) = \text{softmax}\left(\frac{Q_{mm} K_{mm}^T}{\sqrt{d_k}}\right) V_{mm} \tag{2}$$

[1] Using image patches as queries resulted in lower quality pseudolabels.

on the sequence of multimodal embeddings $\{\vec{m}_{\text{cls}}, \vec{m}_1, ..., \vec{m}_N\}$, which produces an attention matrix $\mathcal{A}_{self}$ of dimensions $N \times N$, where N is the length of the language sequence. It is then straightforward to choose the top k most attended positions using the 0-th row of $\mathcal{A}_{self}$, which corresponds to $\vec{m}_{\text{cls}}$, the multimodal representation of the image-text pair. The tokens in the most attended positions are then taken to be the natural language concepts most relevant to the content of the image, and are used as pseudolabels. In practice, we found that $k = 4$ yielded the best results.

4 How much does unimodal pretraining matter?

We experiment with training from scratch instead of initializing from pretrained weights of BERT and DeiT in Table 1. Initializing from pretrained core models is efficient: training from scratch slows down pretraining. This effect will likely diminish as the number of training pairs increases.

Table 1. Training with pretrained core models is more efficient.

		Flickr 0-shot		RefCOCO+	
Weight initialization	Pairs	TR@1	IR@1	TestA	TestB
From pretrained BERT/DeiT	591k	61.0	45.9	36.8	30.4
From Scratch	591k	18.8	13.9	18.4	14.4

5 Fashion Image Retrieval

We compare a fine-tuned version of SIMLA against the state of the art KaleidoBERT [68] on image-text retrieval in the fashion domain using the FashionGen [44] dataset. We use original test split and follow FashionBert's [13] procedure to create the gallery for evaluation. Specifically, we sample 1000 product IDs, and use the frontal pose for each product as the image. For the text, we use both the product name and the product description. We use the same fine-tuning settings as for Flickr.

Table 2. Image-text retrieval on FashionGen[44].

Model	TR@1	TR@5	TR@10	IR@1	IR@5	IR@10
KaleidoBERT [68]	33.8	60.6	68.6	28.0	60.1	68.4
SIMLA	**48.9**	**80.2**	**89.6**	**51.3**	**82.6**	**89.9**
Δ Change	↑ 14.9	↑ 19.6	↑ 21.3	↑ 23.1	↑ 22.5	↑ 21.5