MASKCLIP: MASKING IMPROVES TRANSFER FOR VISUAL MODELS TRAINED WITH NATURAL LANGUAGE SUPERVISION

Floris Weers
University of Twente
Apple Inc.
research@weers.dev

Supervisors:

Tom Gunter
Apple Inc.
tom_gunter@apple.com

Nicola Strisciuglio
University of Twente
n.strisciuglio@utwente.nl

Committee member:

Marieke Huisman
University of Twente
m.huisman@utwente.nl

ABSTRACT

The machine learning community has always aimed at training models that can be applied to a variety of tasks, without the need to design and train a model for each task individually. Foundation models are trained on large datasets with a training task independent of any specific target task and they have shown to perform very well on unseen tasks. The contrastive learning scheme has been proven to be a good pre-training task for a foundation model, in which the model learns to encode images and their matching texts close in an embedding space. Due to the design of the contrastive loss, it can be applied to an unseen task in a zero-shot setting by comparing pairs of similar or dissimilar image/text samples. In this case, we compute the similarity between encoded potential text labels and an encoded image, where the most similar text-image score tells us what is visible in the image. Large-scale vision and language models using contrastive learning have shown great whole-image classification performance. However, attention to detail and the capability to match text to a region of the image (visual grounding) is lacking. The masked patch prediction task is another pre-training task, in which the input is masked by a high ratio and the model needs to predict the original raw values of the masked areas. This task has shown to result in very good visual grounding.

We propose MaskCLIP, an architecture that incorporates a per-sample masking task into an existing technique for learning visual models with natural language supervision via a contrastive loss. We demonstrate that combining the two tasks in a multi-task setting, where we use per-patch similarity scores to improve the masking strategy, substantially improves the quality of the learned image representations for downstream structured and visual Q&A tasks. Our results indicate that the combination of both a contrastive and masked loss contributes to improved effectiveness of transfer learning to downstream tasks beyond whole-image classification in a data-efficient manner when trained on large and relatively noisy paired image-and-text datasets crawled from the web.

1 INTRODUCTION

While neural networks have existed for a long time (Schmidhuber, 2015), they were trained and used on very specific tasks. In many cases, the target dataset would also be used during training, and while the trained model showed to be useful, the same model could not be directly used for other tasks. We refer to this as in-domain training and application.

Recently, self-supervised models were shown to be useful for multiple down-stream tasks and they became the basis for foundation models (Bommasani et al., 2021). These models are first pre-trained on a general dataset that is not chosen for any downstream task in particular. The objective of the pre-training is for a model to be a general feature extractor. The pre-training task ensures that the foundation model learns abstract relations in the pre-training dataset that can be applied to several down-stream tasks. These tasks are referred to as down-stream tasks, as the model was not trained specifically for them. Sometimes, the down-stream task data is also different from the training data, e.g. when the down-stream tasks focuses on highly-accurate medical images, but the pre-training data only contains noisy web-scraped images without medical images. We refer to these tasks as out-of-domain tasks.

Because of the requirement to be a general feature extractor, it is challenging to design and train a good foundation model. The trained model needs to be good in many different aspects, with different downstream tasks requiring different properties. A task can be hard because of two main reasons.

Intrinsic The task is difficult on its own. E.g. because images are blurry, low resolution or the objects that need to be detected are occluded.

Little data The task is difficult because it is rare and not much training data is available. This is the case for many medical classification tasks where data is scarce.

Foundation models can help in both types of hard tasks, by pre-training on a large dataset to gain general knowledge. These models are then fine-tuned using the data that is available for the specific downstream task. We refer to fine-tuning on a different task as transfer learning (Mensink et al., 2021).

Next to fine-tuning on a downstream task, some foundation models can also be applied to down-stream tasks without any fine-tuning. In this case, we exploit the pre-training knowledge and the designed architecture to *zero-shot* predict answers for down-stream tasks. Contrastive learning is a pre-training task that has this zero-shot capability if applied in the way e.g. CLIP (Radford et al., 2021) or ALIGN (Jia et al., 2021) do. In their architecture, separate text and image encoders are used to encode the text and image domains as is shown in Figure 1. The model then learns to maximize the similarity between matching image-text pairs, while minimizing the similarity between non-matching pairs within the mini-batch. Any down-stream task that requires the model to e.g. classify what objects are visible in an image, can be immediately evaluated by computing the similarity between different object names and a single image. A high similarity means the object is visible in the image, while a low similarity means the object is not part of the image.

Initially, foundation models were introduced in the field of natural language processing (NLP). Devlin et al. (2019); Radford et al. (2018); Hendrycks et al. (2021) have shown that self-supervision that directly learns from raw data performs very well. In a self-supervised task, no explicit labeling is used which allows the model to use a large pre-training dataset, as was e.g. shown by BERT (Devlin et al., 2019). In computer vision, BiT (Kolesnikov et al., 2020) and ViT (Dosovitskiy et al., 2021) proved that scaling both the used dataset and the model capacity produces better image-representation models when training with labeled data, and Baevski et al. (2022); Caron et al. (2021b); He et al. (2021); Xie et al. (2021); El-Nouby et al. (2021); Bao et al. (2021) show that scale combined with self-supervision may be sufficient, bringing the worlds of vision and language representation learning closer together. In 2021, a line of work (Radford et al., 2021; Jia et al., 2021) emerged investigating large-scale weak supervision in the form of cross-modality alignment, showing state-of-the art performance, without any finetuning (*zero-shot*), on a wide variety of downstream classification tasks.

These works use the contrastive learning scheme as a pre-training task and make use of large web-crawled datasets of images and their alt-text (Schuhmann et al., 2021; 2022). This contrastive learning is considered a high-level task, as it is applied on whole images and whole texts and requires

the model to learn similarities and dissimilarities between whole-image and whole-text pairs. Due in part to the noisy and multi-domain nature of the large data used for pre-training, and in part to the high-level nature of the used contrastive loss, visual models trained in this manner exhibit good transfer to a variety of downstream whole-image classification tasks, but are typically very data inefficient due to the need to train on many different samples to ensure it learns good global representations. These models are also not as useful when transferring to Visual Question Answering (VQA) (Dou et al., 2021) and object-detection problems (Li et al., 2021c). In VQA, an image and a text question are provided as input, with a text answer as ground-truth output. The questions often refer to specific objects, or relations within the image (Johnson et al., 2017; Goyal et al., 2017; Antol et al., 2015), which require more attention to detail. Similarly, object-detection requires the detection of objects in an image. Often, these objects are occluded or not clearly visible (Chen et al., 2015; Plummer et al., 2015), which again requires a very detailed representation of the image. Models such as CLIP (Radford et al., 2021) lack visual grounding, the ability to match a query with a specific object in the scene (Zhong et al., 2021; Shen et al., 2021) due this high-level whole-image whole-text contrastive pre-training task.

Recent efforts to learn visual models through a self-supervised masked patch prediction task (He et al., 2021; Xie et al., 2021; El-Nouby et al., 2021) demonstrate excellent data efficiency and transfer to a wide variety of downstream tasks. MAE (He et al., 2021) in particular, showed very good performance in a data-efficient way, as not many samples are required to get good performance. In short, images are split into patches like in ViT (Dosovitskiy et al., 2021), masked at a high ratio (75%), and fed into a transformer (Vaswani et al., 2017) encoder. A decoder is then used to reconstruct the original image as a pre-training task. This reconstruction in the decoder induces structural information and details of the image to be embedded in the learned representations by the encoders, which also helps with any down-stream task that requires attention to detail, like VQA. We refer to this reconstruction task as the generative task. As the model need to reconstruct detailed patches, this is considered a low-level task that focuses on local attention.

In this work, we investigate techniques to address the shortcomings in CLIP, and in doing so produce a more well-rounded image encoding foundation model that can perform well in both whole-image classification tasks and tasks that require attention to detail. More specifically, we aim to answer the following questions.

RQ1 How can we improve data efficiency and visual grounding for visual models trained on natural language supervision via a contrastive loss, by incorporating a masked patch prediction problem in the style of BERT/MAE?

RQ2 How do we combine the high-level contrastive task with the low-level generative task, while making sure both tasks keep their beneficial properties?

RQ3 How do we guide the generative task to learn more details in important areas, without any additional data pre-processing or annotation?

We train a single multi-task model that uses both a contrastive task and a masked patch prediction task using both text and image modalities, called MaskCLIP. We demonstrate that models trained in this fashion have a better visual grounding ability, shown by an improved transfer to downstream structured tasks and visual Q&A tasks. In our comparisons with CLIP, we see improvements across the board, with an increase of 0.8% on zero-shot ImageNet, over 2% on zero-shot VALSE, more than 6% on the Visual Task Adaptation Benchmark (VTAB (Zhai et al., 2020), a group of image classification tasks) and over 5% when fine-tuning on the CLEVR (Johnson et al., 2017) VQA task. This work contributes the following.

- MaskCLIP, a multi-task vision and language model that uses both a generative and contrastive loss for zero-shot improvements and impressive down-stream performance.
- We show that modifying the final normalization layer in the text and image encoders, when used in combination with the generative task, allows the trained encoders to demonstrate meaningful patch-or-token-wise similarity scores when compared to a CLIP-style baseline, despite using only a whole-image vs whole-text contrastive loss.
- Similarity-masking, a new masking strategy that uses the improved per-patch similarity scores. It increases the masking probability for patches that are similar to the encoding from the other modality. This way, the model is guided to learning these 'similar' areas in

more detail. We demonstrate that whilst some downstream tasks are hurt by more guided masking, zero-shot performance benefits from the similarity masking technique.

1.1 Related work

For years, self-supervised training in combination with transformers (Vaswani et al., 2017) has been dominating NLP (Bommasani et al., 2021; Wang et al., 2019), with recent changes still improving in this same direction (Aghajanyan et al., 2022; Brown et al., 2020). Two popular pre-training tasks are masked language modeling (MLM) and causal language modeling (CLM). BERT (Devlin et al., 2019) proposed to use MLM (based on Taylor (1953)), where part of the input is masked and predicted by a stack of transformer blocks. BART (Lewis et al., 2019) improves upon BERT by using an encoder-decoder structure. In contrast, in CLM, the model needs to predict the next (or previous) tokens based on provided tokens (Radford et al., 2019; 2018; Howard & Ruder, 2018).

Two lines of work have shown great promise through extending this line of reasoning towards training visual models.

Augmentation Based Contrastive Learning: Using multiple augmentation-based views of the same image as positive pairs in a contrastive loss, while treating other images in the batch as negative pairs, is shown to be a successful pre-training method (Zbontar et al., 2021; Caron et al., 2021a; Chen et al., 2020a;b). Similar to this direction, several works also remove the negative pairs entirely and only focus on positive pairing (Chen & He, 2020; Grill et al., 2020). In all these works, the choice of data-augmentation seems to be very important. DINO (Caron et al., 2021b) shows good visual grounding, most likely because of the heavy data-augmentations (Atito et al., 2021).

Masked Patch Prediction: Masked image modeling (MIM) can be considered closest to the masking task that is applied in the NLP domain (Devlin et al., 2019). An image is divided into patches and a large part of these patches are randomly masked (He et al., 2021; El-Nouby et al., 2021; Xie et al., 2021). The model will predict the raw pixel values (or tokens) of the masked patches. Because of this pre-training task, these models are very robust against occlusions and they make good use of local attention.

BiT (Kolesnikov et al., 2020) and ViT (Dosovitskiy et al., 2021) were one of the first to show the importance of scale for good transfer of pre-trained representations. These works prove that both a large pre-training dataset and increased model capacity improve vision representation learning.

CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) apply the idea of self-supervised contrastive learning to noisy paired image-text data scraped from the web. Both the training dataset size as the model capacity are scaled up, building upon the proved scaling opportunities shown in Kolesnikov et al. (2020); Dosovitskiy et al. (2021). They consider the matching image and text as positive pairs, while all other image and text pairs in the batch as negative pairs. By using a large batch size, no memory bank (Chen et al., 2020b;c; He et al., 2020) or other more sophisticated technique is required for the contrastive loss (Chen et al., 2021), which speeds up computation and reduces complexity. Both modalities have separate encoders, where the contrastive loss is responsible for aligning the two embedding spaces. See Figure 1 for an overview of the architecture. These models show impressive zero-shot performance on whole-image classification tasks, due to the high amount and the high variety of pre-training data. Luo et al. (2021); Ge et al. (2022); Dong et al. (2022); Liu et al. (2022) approach video-text problems in a similar manner, using weak cross-modality supervision.

SimVLM (Wang et al., 2021) also trains on large-scale noisy image-text data, but uses generative modeling. They concatenate both image and text data as input for a transformer-based encoder, which creates an embedding. This embedding is used, in combination with part of a description or a question, by the decoder that will finish the text. Similar is UNITER (Chen et al., 2020d), which also embeds both image and text, but is using two separate encoders and which applies generative modeling in both image and text modality. These models show great promise in terms of data-efficiency, yet have a harder time performing in the large data/model-capacity regime.

Recently, MAE (He et al., 2021) proved that masked patch prediction, applied in the vision domain, is very data-efficient. In MAE, an encoder will only use the visible patches as input, while

the decoder needs to predict the raw values of the original masked patches. Figure 2 shows the architecture. While other works have tried to adapt this to the vision and language domain (Tan & Bansal, 2019; Dou et al., 2021; Arici et al., 2021; Yu et al., 2020; Li et al., 2021a; Lu et al., 2019; Su et al., 2020; Chen et al., 2020d; Huang et al., 2021; Kim et al., 2021), they either relied on Convolutional Neural Networks (CNN's) for region of interest (RoI) embeddings, used RoI's that were detected by an existing model, or only predicted the masked patches' object classes instead of the raw patch values. Both (Kim et al., 2021; Dou et al., 2021) tried patch prediction, but found it hurt their performance. Compared to successful masking pre-training tasks (He et al., 2021; El-Nouby et al., 2021), their proposed masking rate was significantly lower and masking was based on RoI, instead of randomly chosen same-sized patches.

SplitMask (El-Nouby et al., 2021) applies both a masked image modeling loss and a contrastive loss where the inverse mask is considered the other positive element. SplitMask is applied in the vision domain and uses a token from the decoder for the contrastive loss.

Within related works using encoder and/or decoder structures, two main types exist.

Single stream Both the vision and language modalities are concatenated and used as input for a single encoder/decoder. This allows for very fine-grained cross-domain relations, as the lowest level input can already be related to each other (Su et al., 2020; Alberti et al., 2019; Li et al., 2019).

Dual stream Each domain is first encoded using separate encoders. The two embeddings are then linked by a separate cross-attention transformer. This allows for high level relations between the domains (Tan & Bansal, 2019; Lu et al., 2019).

SemVLP (Li et al., 2021a) is an exception that uses both a single and dual stream, where all encoders share weights.

Multi-task methods for pre-training visual encoders are part of a highly active area of research. Florence (Yuan et al., 2021) makes use of a Swin transformer (Liu et al., 2021) as the image encoder, and a diverse array of tasks and datasets at pre-training time (including object detection, CLIP-style contrastive modeling, VQA, and supervised classification). ALBEF (Li et al., 2021b) builds on a CLIP-style baseline architecture, incorporating a single-stream encoder that consumes inputs from both modalities, targeting a masked language modeling loss for the text modality alongside an image-text matching loss for within-batch hard-negatives mined according to contrastive similarity. FLAVA (Singh et al., 2022) also makes use of both single as well as joint modality encoders, incorporating masked-image modeling and additional single-modality datasets (not just image-text pairs, but e.g. just images and just texts), allowing it to generalize to longer text inputs and visual data which is unlikely to be found in a caption. X-VLM (Zeng et al., 2022) utilizes image data which includes object-labelled bounding boxes, which allows the authors to extend the image-text matching and contrastive losses to cover within-image regions as well as whole-image captions. They too include a masked language modelling objective, achieving impressive performance on many zero-shot downstream tasks. It should be noted that this approach may not scale well to large pre-training datasets, due to the need for sufficient object bounding box data. SupMAE (Liang et al., 2022) adds a supervised labeling task to the MAE architecture, demonstrating that this results in better data

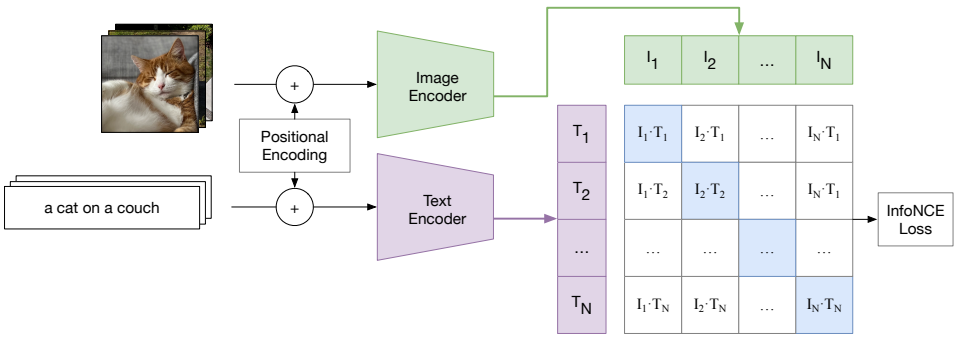


Figure 1: CLIP: dual-encoder architecture using a contrastive loss.

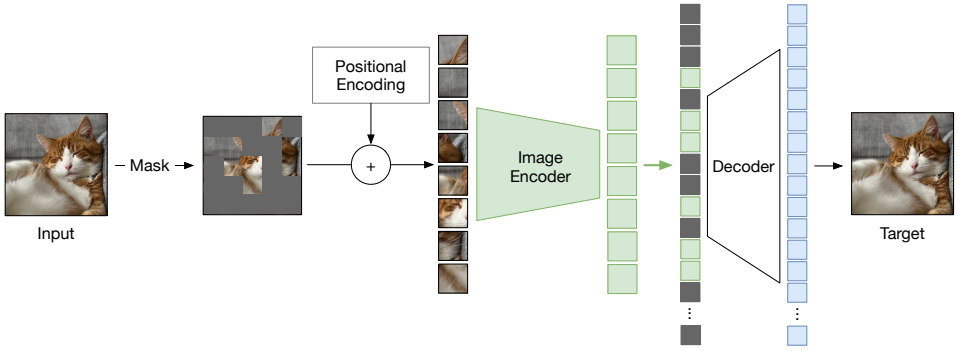


Figure 2: MAE: auto-encoder architecture on the image modality.

efficiency. MVP (Wei et al., 2022), meanwhile, finetunes a pre-trained CLIP backbone using MIM, and demonstrates that starting with a pre-trained model improves downstream visual recognition performance. CoCA (Yu et al., 2022) combines the cross-modality contrastive task from Radford et al. (2021) with image captioning in the style of Desai & Johnson (2021), and by pre-training at very large scale show that the resulting model is more performant than prior art across a very broad array of downstream visual understanding tasks.

We propose MaskCLIP, a model that uses both a contrastive and generative task, with a dual-stream architecture to support single-modality down-stream tasks (see Figure 3). Similar to large-scale models like CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), we use web-scraped image and text pairs for contrastive learning. Next to that, we apply MIM and MLM. Following MAE (He et al., 2021) and BART (Lewis et al., 2019) we use an encoder-decoder structure to predict masked patches. Our work is similar to ALBEF, but we apply the masked patch prediction to both modalities, without momentum distillation. The closest related works are CoCA and FLAVA, but without single-modality masked patch prediction tasks and with the use of high masking ratios.

2 Approach

Figure 3 shows the model architecture, where green components correspond to the image modality encoding stream, while the purple components correspond to the text modality encoding stream. The dashed lines are used in the contrastive task, while the solid lines show the generative task. We use a multi-task setting, with a structure close to CLIP (Radford et al., 2021) for the contrastive task, but with a generative task that predicts masked patches.

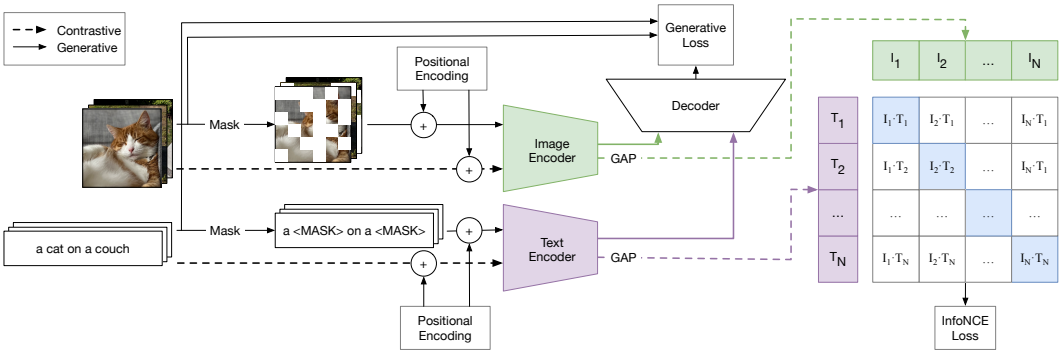


Figure 3: MaskCLIP: combining a generative and contrastive task to improve down-stream performance.

2.1 MODEL ARCHITECTURE

Image Encoder: Following ViT (Dosovitskiy et al., 2021), we divide the input image into equally-sized, non-overlapping patches. After applying a linear projection and adding a 2-D position encoding, we optionally apply masking with a high ratio (e.g. 75%). Only non-masked (visible) patches are used as input for the ViT encoder to reduce memory usage, following MAE (He et al., 2021). Initial empirical experimentation proved that using a learnable or fixed position encoding does not influence results, we therefore use a fixed 2D position encoding. The encoder is based on transformers from Vaswani et al. (2017), in which each layer consists of a self-attention module, skip connection and multilayer perception. We use a pre-layernorm for training stability (Xiong et al., 2020) and follow initialization from Radford et al. (2019).

Text Encoder: Following BERT (Devlin et al., 2019), text is tokenized and a 1-D position encoding is added. We then optionally apply masking with a high masking ratio (e.g. 75%) on the tokens. Masked tokens are replaced by a shared learnable mask token, as no memory usage be easily gained because of the dynamic input length. Following related work, we use a learnable 1D position encoding. We follow the same transformer-based implementation as in the image encoder, using pre-layernorm and initialization as in Radford et al. (2019).

Cross-modality Decoder: The decoder receives per-element encoded image and text representations from the encoders for the visible elements. For the image modality, masked patches are added as a shared trainable mask token. Positional encodings are once again added to all elements. Next to that, we add a per-modality trainable encoding to allow the decoder to easily distinguish the two modalities. The decoder is built out of transformer layers which share their implementation details with those used in both the image and text encoders. The output of the final decoder layer is mapped to raw pixel values for the masked image patches, and matrix-multiplied against the text encoder's vocabulary embedding array to produce a distribution over token IDs per text element.

2.2 PRE-TRAINING TASKS

We use two pre-training tasks, as defined below.

Contrastive task: For this task we do not apply any masking to either the input image or the text elements. We normalize the per-element output of the image and text encoders by dividing by the elementwise maximum of the L2 norm over the hidden dimensions, more on this in Section 4.3. We global-average-pool (GAP) the output of each encoder to produce a single encoding vector per modality, and then apply a linear projection layer with no bias term in order to map this to a shared `class` space. Following Radford et al. (2021), we maximize the cosine similarity of the image `class` token and the text `class` token of N real pairs in the mini-batch, while minimizing the cosine similarity of the $N^2 - N$ class token pairs (based on InfoNCE, Oord et al. (2019)). A symmetric Cross Entropy Loss is computed over these similarity values.

Generative task: In this case, we mask a large fraction of the input elements to each of the encoders. The per-elementoutput of the encoders is then concatenated, before positional encodings are once again added, alongside a modality indicator encoding. The output of the decoder is then mapped to predict the original pixel-values for the masked image patches and the original token IDs for the masked text. The decoded image patches are compared against ground-truth using mean squared error. The decoded text tokens are used in cross-entropy loss with the original token. Both losses are only applied on the masked patches. We refer to the combination of these two losses (predicted image patches and predicted text tokens) as the generative loss.

For each task we run both the encoders and the decoder forward propagation steps, using the same batch. Although a single forward pass though the model is computationally appealing, early empirical experimentation found that masking the inputs to compute the contrastive loss severely degrade performance. We therefore train in a multi-task setting, where gradients are accumulated in a single update.

2.2.1 GLOBAL AVERAGE POOLING

The choice of pooling strategy, used to produce a single representation vector out of per-patch-or-token encoder outputs, has been shown to have minimal effect on performance for transformers

trained on supervised classification problems. In Scaling Vision Transformers (Zhai et al., 2021), they showed that global-average-pooling (GAP) and using a separate `class` token show very nearly equivalent performance. Multiheaded-attention-pooling (MAP), where a final attention layer that can attend to all per-patch-or-token outputs is used to produce a single `class` token, showed to be a slight victor out of the three. For our contrastive task, in which an image may contain multiple regions of interest, it is plausible that GAP may be the best choice out of three. This is because GAP encourages the output to consider the contribution of each output patch-or-token equally, resulting in a bias towards a compressed representation that contains information about the whole image. If the text query is 'dog' and a dog exists in the image, we hope to retrieve the image even if another entity is the focus.

A useful pair of encoders with good visual grounding should allow us to make use of patch-and-token-wise similarities to probe for region of interests (RoI) in both the text and image input. The default layer-norm applied just before the pooling strategy makes this tricky, however, as it projects all elements onto the unit sphere. This means that we must rely on the dimensionality of the embedding space to be high enough that irrelevant patches are orthogonal to many possible encoded queries. To alleviate this pressure, we switch to a max-norm formulation, which allows the model to make element contributions arbitrarily small whilst still demonstrating identical training stability to layer-norm. To make this concrete, where layer-norm would apply:

$$\hat{e} = \frac{e - \mathrm{E}_h[e]}{\sqrt{\mathrm{Var}_h[e] + \epsilon}} * \gamma + \beta \tag{1}$$

per element e (transformer output patch-or-token) in a sequence of length S, with the expectation taken over the hidden dimension h and with trainable γ and β parameters, max-norm instead calculates:

$$\hat{e} = \frac{e}{\max\limits_{0 \leq s < S} \|e_s\|_h^2}. \tag{2}$$

This allows the contributions made by individual patches to be made arbitrarily small, which in turn allows element-wise similarity scores to be low through either orthogonality and/or negligible scale. In other words, the model can learn to ignore certain patches if they do not add value to the final `class`.

2.3 Training

In this section we first define the training objectives, before then discussing a strategy for using the elementwise similarities to guide the generative task's masking strategy.

2.3.1 Objectives

Our goal is to improve the visual grounding for CLIP style models, with the constraint that the resulting architecture and training recipe should admit large scale datasets and model sizes. To do so, we retain the pairwise InfoNCE loss, as in Radford et al. (2021). This loss scales well to large and noisy aligned text-image datasets, as well as large model sizes (Pham et al., 2022). In particular, we define the image-to-text loss as:

$$L_{i2t} = -\frac{1}{N} \sum_{i}^{N} \log \frac{\exp(x_i^T y_i / \sigma)}{\sum_{j=1}^{N} \exp(x_i^T y_j / \sigma)} \tag{3}$$

Where N is the global batch size, x_i is the is the normalized embedding of the image for the i-th pair and y_j is the normalized embedding of the text in the j-th pair in the batch. σ is a trainable temperature parameter. Note that often, initial training is done using a *local contrastive loss*, using N as the local batch size. This lowers the difficulty at the start. The text-to-image loss is symmetric, and the total contrastive loss is therefore:

$$L_c = \frac{1}{2}\left(L_{t2i} + L_{i2t}\right). \tag{4}$$

In addition to L_c, we introduce a generative masked element prediction task scored at the output of the decoder, which enables us to make use of within-example as well as within modality information, and furthermore provides an inductive bias towards position-wise consistency throughout the model. This loss relies only on the contents of the image-text pairs, and comprises mean-squared error for the masked image patches:

$$L_{\text{gen\_i}} = \frac{1}{N}\sum_{i}^{N}\left(p_i - P_i\right)^2, \tag{5}$$

and cross-entropy over the vocabulary for the masked text tokens

$$L_{\text{gen\_t}} = -\frac{1}{N}\sum_{i}^{N}\log\frac{\exp(t_{n,T_n})}{\sum_{c=1}^{C}\exp(t_{n,c})}. \tag{6}$$

The total loss is then

$$L = L_c + w_i * L_{\text{gen\_i}} + w_t * L_{\text{gen\_t}}, \tag{7}$$

where w_i and w_t are scalars used to control the relative weight of the generative losses.

2.3.2 Similarity masking

One difficulty with combining a "bottom-up" and "top-down" task in this fashion is that, whilst we hope there is sufficient overlap that the combination outperforms each task individually, the two tasks have different goals. The generative task aims to encode maximum redundancy per image-text pair, whilst the contrastive task looks to only encode what is relevant to the other modality. In practice this means that the generative task may use significant model capacity to represent information unlikely to be relevant (variation in textures, e.g.), which in turn penalizes the zero-shot classification performance. This effect is particularly pronounced at smaller model sizes, where the two tasks compete for capacity.

We therefore propose an alternative masking strategy in order to improve task overlap, visualized in Figure 4, and hereafter referred to as 'similarity masking'. In order to compute the per-element masking probabilities using this technique, we first calculate the similarity scores between the elements in the modality to be masked (e.g. patches for the image modality), and the GAP of the other modality (e.g. the `class` encoding for the text modality). As the pre-training data is noisy, we try to apply similarity masking predominantly to image-text pairs that are well aligned. This is done by

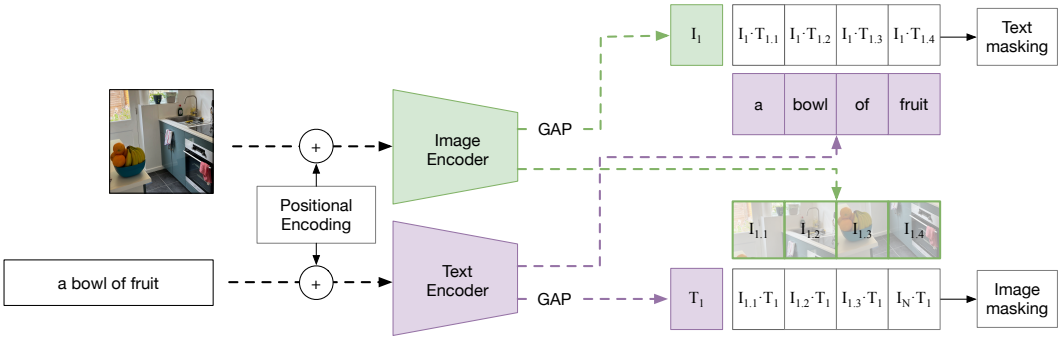


Figure 4: Compute patch-wise similarity scores, used to guide the masking towards more similar patches/tokens.

computing the whole image-text similarity score and applying a scaling function to normalize this value so that $i2t_{sim} \in [0, 1)$. Finally, we assign $i2t_{sim}$ masking probability mass to the element-wise similarity scores, normalized over an example, and $1 - i2t_{sim}$ mass to a uniform distribution (in effect to the default random masking strategy). By doing this, we encourage the generative task to assign model capacity to the same regions of the image and text that are well aligned when they are well aligned, and otherwise rely on the more robust random masking strategy to capture detail where the alignment is more uncertain.

3 EVALUATION

We evaluate MaskCLIP in three ways. Zero-shot evaluation while training, to compare initial performance and to early exit when any architecture change breaks the training dynamics. Further quantitative evaluation is done using down-stream tasks to probe the image and text encoders and to show visual grounding and local attention improvements. Finally, qualitative evaluation is performed to visually inspect and understand the model.

By using these three categories, and not running the full evaluation for every trained step, we reduce turn-around time while still maintaining accurate performance insights.

We will first go over the details of our training setup, before we explain the quantitative evaluation tasks. Finally, we introduce the used qualitative evaluation tasks.

3.1 EXPERIMENTAL SETUP

Initial experiments were performed using an image-encoder based on the architecture described as ViT-S in Xiao et al. (2021), in combination with a 12-layer text-encoder with 4 attention heads and 256 embedding dimensions. This relatively compact model allowed us to iterate quickly, and showed results in line with what we present for larger model sizes. We trained for 90k steps (3 epochs), using a global batch size of $16,384$ (512 image-text pairs per GPU). Following Radford et al. (2021), we use an AdamW optimizer (Kingma & Ba, 2017; Loshchilov & Hutter, 2019) and decay the learning rate using a cosine schedule (Loshchilov & Hutter, 2017). We use a noisy web-crawled dataset of over 500 million image and text pairs that is composed of LAION-400M (Schuhmann et al., 2021), CC3M (Sharma et al., 2018), CC12M (Changpinyo et al., 2021) and internal datasets.

Further research was done using ViT-B for the image encoder together with a 12-layer text encoder with 8 attention heads and 512 embedding dimensions. We train for 180k steps (6 epochs) and we use a global batch size of 16384 (256 per gpu). This training is performed on 64 A100 GPUs, for 6 days.

To increase training convergence speed, we only start with global contrastive loss after 10000 steps. We found the need to multiply the generative text and generative image loss by 0.05 and 0.1 respectively when using global contrastive loss. Both modalities are always masked by 75% simultaneously. For MaskCLIP trained with similarity masking, we initially use random masking and switch after 1 epoch. A more detailed overview of the hyperparameters can be found in Appendix C.

Baselines: To provide a fair baseline we train a CLIP model with the same configurations, pre-training data and pre-training duration. Unless otherwise noted, we refer to this as 'CLIP' in our tables of results.

3.2 QUANTITATIVE EVALUATION

Quantitative evaluation is done on a CLIP baseline, our proposed MaskCLIP and various ablations to compare results objectively.

Zero-shot evaluation: While training, we compute zero-shot retrieval performance every 3k steps on the Flickr30K (Plummer et al., 2015) and COCO (Chen et al., 2015) benchmarks. In Flickr30K and COCO, a single image contains multiple descriptions/labels that describe an object shown. The task is to retrieve one of the correct descriptions from an image (image-to-text), or to retrieve the correct image based on the description (text-to-image). By utilizing the dual-stream architecture for the encoders and pre-training with a contrastive objective, encoders can be directly used to perform the retrieval tasks using the pair-wise similarities. We use ImageNet (Deng et al., 2009) to evaluate

image classification performance. We use a single label per image and consider it an image-to-text retrieval problem. Similar to Radford et al. (2021), we perform image classification by encoding the target classes using the text encoder and computing the similarity for a given image and the set of class embeddings. The target class with the highest similarity score is considered the predicted label. We use the prompt "a photo of" (Radford et al., 2021) for all classification tasks. Both the object retrieval tasks and the image classification tasks allow us to quickly verify that new changes do not break training dynamics in new experiments.

Next to this, we evaluate the VALSE benchmark (Parcalabescu et al., 2022). This is a zero-shot VQA task that we approach as a retrieval task. In VALSE, each image has a correct description and a fake description (a foil). We compute the similarity between the image and its true description, and between the image and its foil. The image-text pair with the highest similarity score is considered the prediction. As the true description and its foil often only differ by a very small number of tokens, improved local attention benefits this task. See Appendix B.1 for more details about the different VALSE tasks.

Due to our argued improved local attention, we also evaluate on COCOAmodal (Sun et al., 2022), which focuses on classifying occluded objects within the COCO dataset. Finally, we evaluate using Winoground (Thrush et al.) as this is a hard and more recent benchmark.

Down-stream performance is only evaluated when the model is fully done training.

Linear-probing: We train a linear head on top of the frozen image encoder. Linear-probing is done on the Visual Task Adaptation Benchmark (VTAB) (Zhai et al., 2020), which contains a variety of tasks, categorized in three groups. Natural tasks contain image classification tasks with images that are captured using standard cameras. These tasks are a good indicator of whole-image classification performance. Specialized tasks contain images of the world that are captured through specialist equipment, like remote sensing and medical. The final group contains structured tasks, which require object counting or 3D depth prediction. Most of these images are synthetically created and their domain differs from benchmarks like ImageNet Deng et al. (2009). Especially the structured tasks are a good way to evaluate local attention, as it requires more attention to detail than a whole-image classification task. See Appendix B.2 for more details on the different VTAB datasets.

Next to this, we also perform linear-probing on CLEVR (Johnson et al., 2017). This benchmark forces the model to understand the image and text in a detailed way, and requires local information to be part of both the image and text embeddings. Since questions in the CLEVR benchmark often refer to relations between objects in an image, more local attention should benefit the performance. We evaluate this benchmark by concatenating both the `class` tokens of the two encoders.

Finally, to compare performance with other works, we also show the linear-probing result on ImageNet.

VQA fine-tuning: Next to linear probing on VQA, we also evaluate VQA benchmarks by finetuning more than just a linear layer. We freeze the encoders at all times and only fine-tune a decoder. We use a layer-wise learning rate decay (Clark et al., 2020), following He et al. (2021); Bao et al. (2021) during finetuning. We include different versions of finetuning, to ensure fair comparisons with other works. Using frozen pre-trained encoders, we will fine-tune using different decoder setups. We fine-tune a decoder from *scratch* that contains the same parameters and modality-specific tokens as is used in MaskCLIP during pre-training. We also fine-tune another *simple* decoder that contains 8 transformer layers without any position encoding or modality tokens added. Next to that, we make use of MaskCLIP's pre-trained decoder and either use a linear-head on top of the pre-trained decoder, or directly generate text tokens as answers. In this generative setup, we mask a fixed number of tokens where the answer would be and fine-tune on predicting the raw answer tokens (see Figure 5). The model also needs to predict the end-of-sequence (EOS) token, that defines the end of the answer. Since the encoders and decoders are not causal, we also experiment with running multiple passes through the decoder, replacing the masked tokens one by one with the prediction until the model predicts an EOS token. This way, multi-token answers should be easier to predict completely correct.

To show that the VQA task is not too easy, we also fine-tune without any encoders and input is immediately passed to the decoder. We perform VQA finetuning on both CLEVR Johnson et al. (2017) and VQAv2 Goyal et al. (2017).

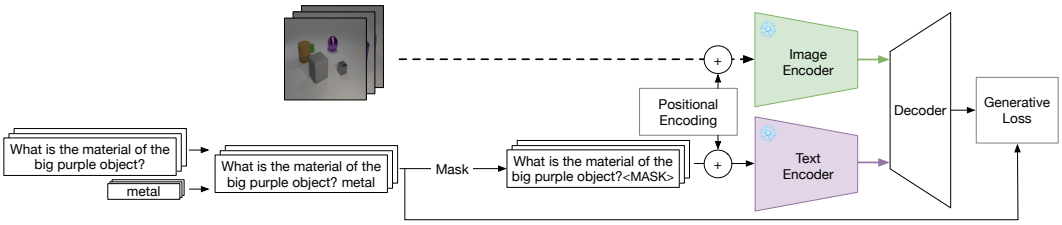


Figure 5: VQA finetuning: using the generative setup to generate answer token ids.

3.3 Qualitative evaluation

Qualitative evaluation will be done to inspect attention, similarity and generated images/sentences. Examples of similarity maps give us an idea of the usefulness of the similarity maps for our masking strategy. They also give an indication whether the model has better patch-wise similarity scores and improved visual grounding. Next to this, visualizing generated images and sentences help us understand whether the generative task is learning successfully, or has a hard time reconstructing the masked patches/tokens.

Similarity masking: Our new masking strategy is based on per-patch/token similarity (see section 2.3.2) and by visualizing the individual steps that we perform to compute the masks, we show that masking is more related to the interesting areas. We will show both the similarity-score map, as the probability map that is used to generate the final mask.

Similarity maps: As we have changed the normalization used before the pooling in the encoders to improve patch-level similarity scores, visually inspecting these gives us insights on whether this was done successfully. Next to this, better similarity maps support the argued improved visual grounding. We will show similarity maps of both the image modality as the text modality.

GradCAM: As we improve visual grounding, we should be able to get more localized GradCAM (Selvaraju et al., 2020) visualizations for specific queries. GradCAM is applied to the final linear layer of the last transformer block of the image encoder and shows how gradients flow through the model for a specific query, thus highlighting important regions in the image for computing the result.

Generative results: To show the model's capability to predict the original patch pixels/sentence tokens, we will feed in various images and objects and mask parts of both modalities. We will show that the model can reproduce sensible image and text results. We will also visualize the output per decoder-layer to see how the multiple decoder layers are used.

t-SNE: Similar to other works (Geng et al., 2022), we visualize the t-SNE on a subset of ImageNet classes to show the quality of the embeddings are good. t-SNE visualizes the embeddings in 2D, which allows us to see clustered points for concepts that are similar according to the trained model. We will visualize the embeddings of images of the first 50 classes of ImageNet, including labels, and verify points that close to each other can be related to each other.

4 Results

We will first discuss the quantitative results, in which we directly compare results between CLIP and MaskCLIP. Afterwards, we will show the qualitative results to give an idea of model's visual grounding capabilities.

4.1 Quantitative

We will go over the quantitative evaluation results, showing improved zero-shot and improved downstream performance when using MaskCLIP.

General zero-shot: Table 1 shows zero-shot performance for MaskCLIP using random masking (referred to as MaskCLIP-r) and using similarity masking (referred to as MaskCLIP-s). We compare

with CLIP that has been trained on the same data for the same duration. MaskCLIP outperforms CLIP in all zero-shot benchmarks (except MaskCLIP-r for Winoground Text, likely because of more focus on background information) using both random and similarity masking strategies. Using similarity masking, we push the zero-shot performance even further.

Table 1: **MaskCLIP outperforms CLIP on all zero-shot tasks**. I→T: Image to Text, T→I: Text to Image.

Model	COCO		FLICKR		Winoground			ImageNet	VALSE
	I→T	T→I	I→T	T→I	Text	Image	Group		
CLIP	51.6	34.1	73.5	55.6	26.5	9.7	7.2	57.8	60.4
MaskCLIP-r	**53.3**	**35.2**	74.2	57.7	24.8	10.8	**8.7**	58.2	**62.8**
MaskCLIP-s	53.0	**35.2**	**75.7**	**57.9**	**29.8**	**11.2**	8.3	**58.6**	61.4

COCOA: Table 2 shows COCOA zero-shot results. Again, we see an improvement over CLIP for all MaskCLIP configurations. MaskCLIP using random masking performs best on top-1 accuracy, while MaskCLIP using similarity masking outperforms the other settings in top-5 accuracy. This is likely caused by similarity masking reducing background focus and therefore predicting those objects with lower confidence than more obvious objects.

Table 2: **MaskCLIP improves zero-shot on COCOA: occluded COCO classification**. The different levels refer to occlusion levels of the target object. A low level has lower occlusion.

Model	COCOA Top-1					Top-5				
	level 0	level 1	level 2	level 3	average	level 0	level 1	level 2	level 3	average
CLIP	34.0	31.7	23.2	21.4	27.6	48.8	47.4	38.6	33.0	42.0
MaskCLIP-r	**35.1**	32.9	**29.0**	**23.8**	**30.2**	53.0	52.1	42.3	**39.3**	46.7
MaskCLIP-s	34.5	**33.5**	26.8	23.3	29.5	**54.5**	**52.3**	**43.4**	38.3	**47.1**

VALSE: Table 3 shows an improvement on the zero-shot VALSE benchmark in comparison to CLIP. We argue this improvement is caused by the required local attention for these tasks as very often, only a small part of the sentence is changed. We see that similarity masking hurts performance, likely because background details are less important during this pre-training setup, but very important for this benchmark.

Table 3: **MaskCLIP also outperforms on zero-shot VQA**.

Model	VALSE						
	actant-swap	action-replacement	counting	existence	plurals	relations	average
CLIP	54.5	76.5	61.2	62.9	61.3	53.5	60.4
MaskCLIP-r	**58.1**	**78.2**	**61.8**	**68.9**	**64.0**	**56.0**	**62.9**
MaskCLIP-s	57.1	77.0	60.0	60.7	62.9	54.1	61.2

Linear-probing: Table 4 shows linear-probing on most tasks of VTAB. MaskCLIP outperforms CLIP in all linear-probing tasks. Random masking outperforms similarity masking when evaluating on structured down-stream tasks, while similarity masking improves performance on natural tasks. Often, the structured tasks require more attention to details in the image than natural tasks (which can be more object-centric). Two tasks that make use of the CLEVR dataset are included in the VTAB-structured group (clevr-closest and clevr-count), but we also evaluate linear-probing on the original CLEVR VQA benchmark and show an improvement in performance when using MaskCLIP.

VQA fine-tuning We show that using a pre-trained decoder outperforms all other settings for both VQA tasks. Next to this, exploiting the generative aspect of the pre-training tasks, shows an even bigger increase in performance. Using our multi-pass generative setup, we improve multi-token answer generation and get the best performance (see Appendix G for generated answer examples). See Table 5 for all VQA results. MaskCLIP outperforms CLIP by a significant amount on CLEVR (Johnson et al., 2017), and slightly improves the result on VQAv2 (Goyal et al., 2017). Using similarity-masking (between brackets) improves VQAv2 results, but lowers CLEVR results by a

Table 4: **MaskCLIP outperforms CLIP substantially on linear-probe tasks**. Especially structured tasks benefit from MaskCLIP.

	VTAB Natural						Specialized			Structured								CLEVR	Imagenet
	caltech101	cifar-100	dtd	oxford-flowers102	oxford-iiit-pets	svhn	Eurosat	Patch Camelyon	Resisc45	clevr-closest	clevr-count	dmlab	dsprites-orientation	dsprites-xpos	kitti-closest-vehicle	smallnorb-azimuth	smallnorb-elevation		
CLIP	94.8	78.3	80.0	97.8	90.3	58.9	95.4	83.2	91.7	46.0	65.5	49.3	47.6	56.2	46.7	26.5	41.8	50.7	74.7
MaskCLIP-r	95.5	78.6	81.0	98.1	**90.9**	**69.3**	96.8	**84.4**	**93.2**	58.4	**74.5**	**53.1**	**58.7**	**71.8**	**52.0**	**46.3**	**54.1**	**51.4**	76.9
MaskCLIP-s	**95.6**	**79.3**	**81.3**	**98.2**	89.8	68.6	**96.9**	84.0	93.1	**58.9**	74.2	52.9	56.8	70.0	46.4	43.6	56.0	51.3	**77.0**

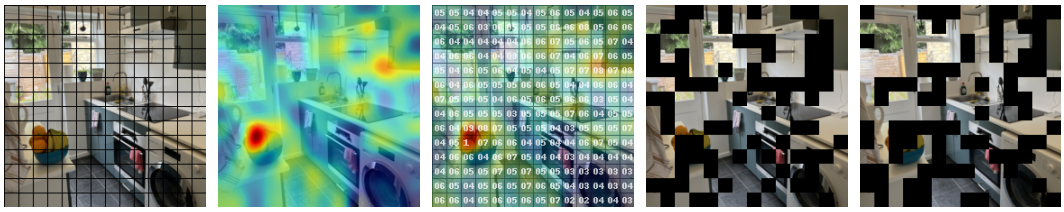
slight amount. We argue CLEVR requires a good understanding of the whole scene, where all objects are visible, which means any area-focus hurts performance. VQAv2 on the other hand is more focused on a central object in the scene. We confirm that the task is not solvable just by having enough capacity in the decoder, as not using any pre-trained encoder hurts performance significantly.

Table 5: VQA Finetuning results after 50 epochs. Encoders are always frozen. Showing MaskCLIP with random masking strategy by default, similarity masking between brackets.
Abbreviations: **Class**ification, **Gen**erative and **M**ulti-**P**ass.

Model	**Encoder**	**Decoder**	**Task**	**CLEVR**	**VQAv2**
	None	Simple	Class.	52.8	45.7
CLIP	Pre-trained	Scratch	Class.	91.3	61.8
		Simple	Class.	78.5	62.3
MaskCLIP	Pre-trained	Scratch	Class.	95.3 (93.3)	61.3 (61.7)
		Simple	Class.	95.1 (94.2)	62.4 (62.6)
		Pre-trained	Class.	96.4 (95.7)	62.3 (62.3)
			Gen.	96.6 (96.0)	61.9 (61.9)
			Gen. MP	**97.2 (97.1)**	**62.5 (62.9)**

4.2 Qualitative results

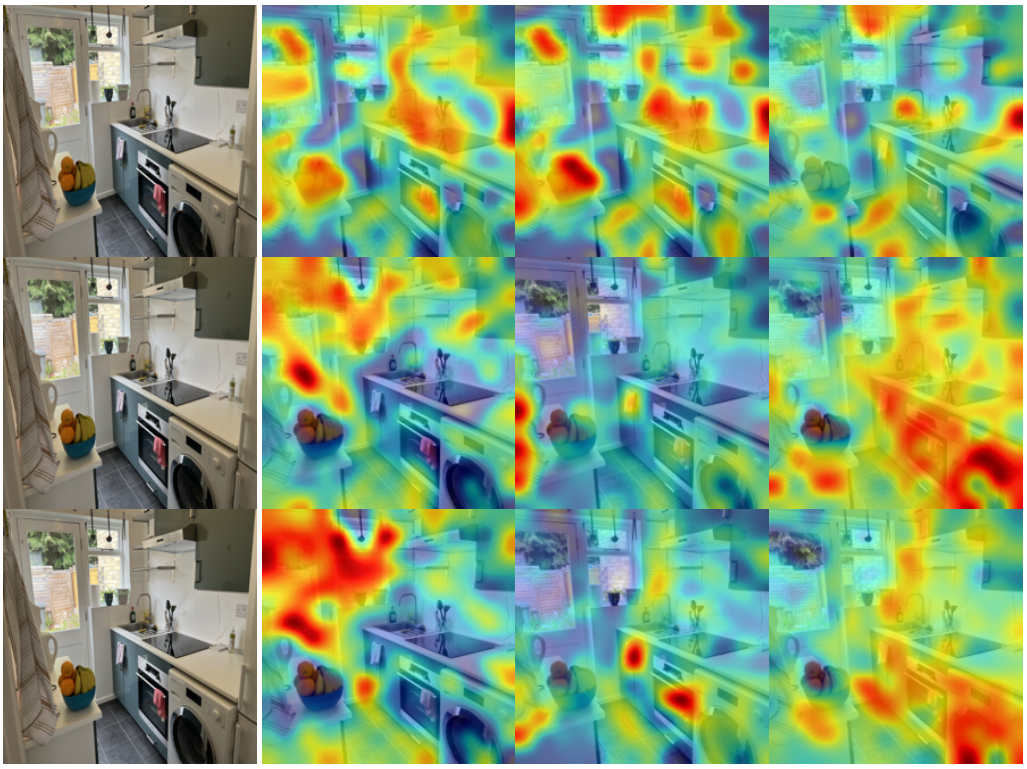
Similarity masking: Figure 6 shows the different steps involved in the creation of a mask based on similarity-masking. We can see the similarity map highlighting the mentioned object from the caption, increasing the probability of masking it. As the similarity masking only increases probabilities, it is not guaranteed to mask the highlighted area.



(a) Ground truth (b) Similarities (c) Probabilities (d) Mask (e) Mask

Figure 6: Similarity-masking with query: *oranges*. Probabilities should be prefixed with 0.00.

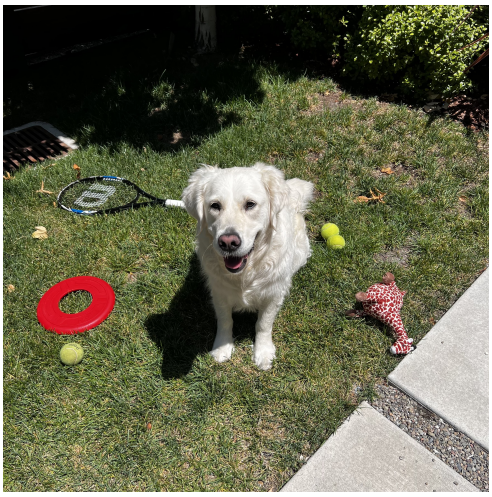
Similarity maps: Similarity maps in the image domain are shown in Figure 7. It can be seen that MaskCLIP has improved patch similarities when comparing with CLIP. Figure 8 shows two examples with text similarity scores for all three model types. It is not as clear as in the image domain, as quality differs more per query.



(a) Ground truth (b) Garden door (c) Towel (d) Washing machine

Figure 7: Image patch-similarity visualizations for different queries. From top to bottom: CLIP, MaskCLIP-r and MaskCLIP-s. MaskCLIP has improved per-patch similarity scores.

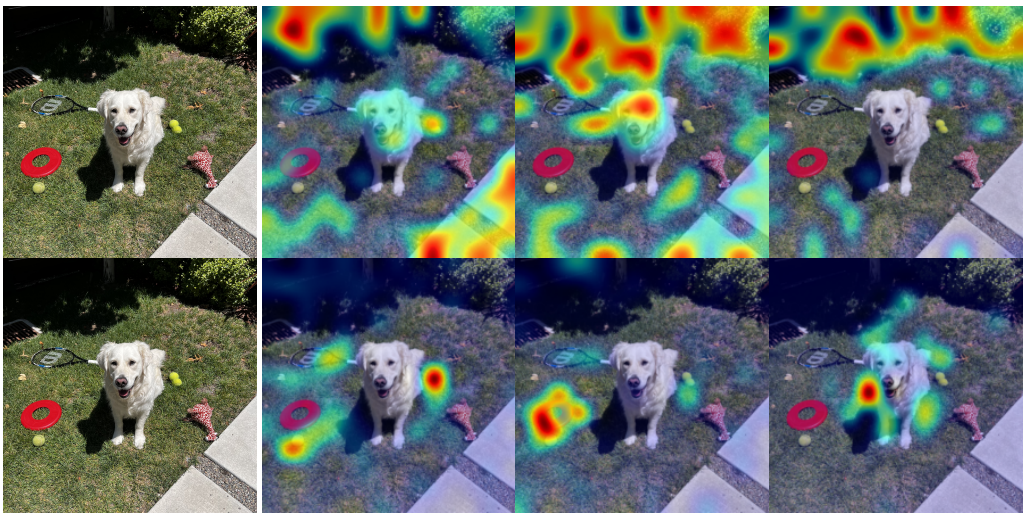
(a) <BOS> kitchen with a bowl of fruit <EOS>
<BOS> a kitchen with a bowl of <EOS>
<BOS> a kitchen with a bowl of <EOS>



(b) <BOS> dog has a lot of toys <EOS>
<BOS> dog has a lot of toys <EOS>
<BOS> the dog a lot of toys <EOS>

Figure 8: Similarity score maps for text token and whole image (from top to bottom: CLIP, MaskCLIP-s and MaskCLIP-r). A high similarity between a text token and the image is shown by a darker font.

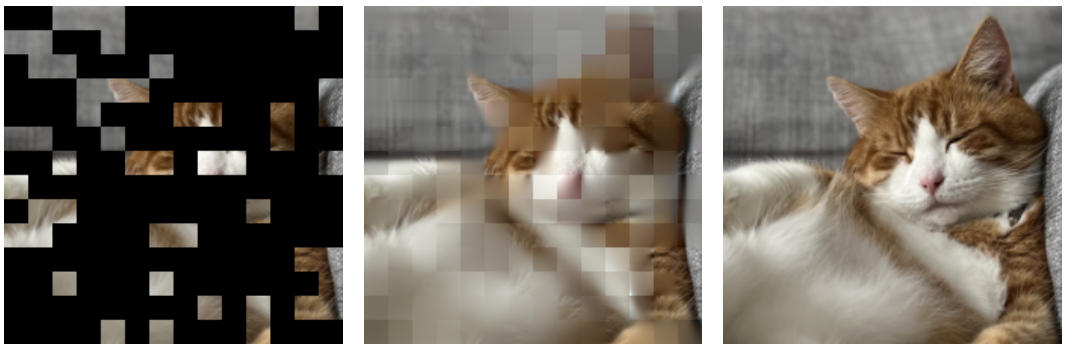
GradCam: Figure 9 shows GradCam applied to the image encoder. We can see an improved localization when comparing MaskCLIP with CLIP using the same queries. This confirms that MaskCLIP has improved visual grounding.



(a) Ground truth (b) Tennis ball (c) Red frisbee (d) Dog

Figure 9: GradCAM (CLIP on top, MaskCLIP at the bottom)

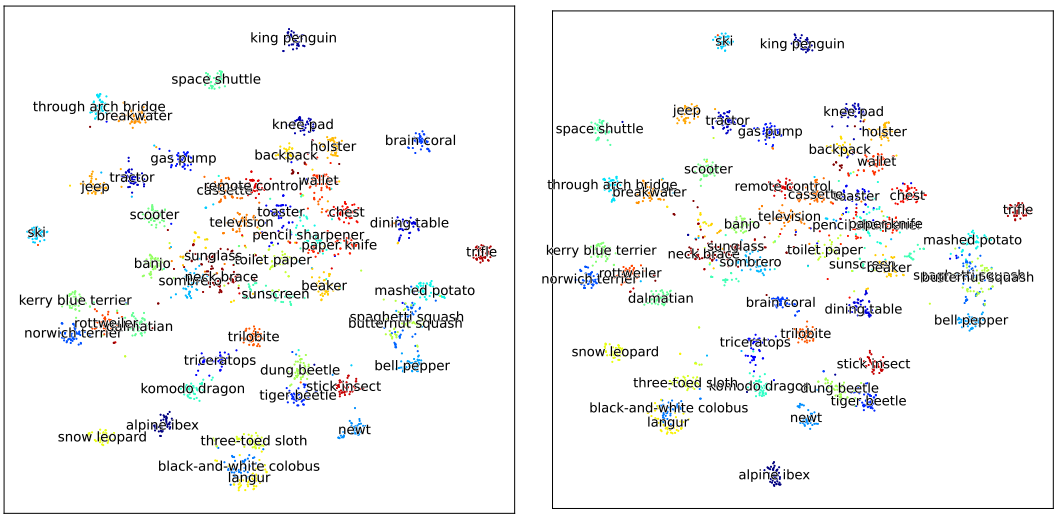
Generative results: We show a generative example in Figure 10 (more examples can be found in Appendix F). While details are lost in the image prediction, overall shapes are approximated. We also see the use of cross-modal attention, as the predicted text contains information ('sleeping') that is only available in the image. Note that we have seen image patch prediction results with more details, when weighting the generative task heavier, at the cost of zero-shot performance.



(a) <><> on<> couch (b) cat sleeping on a couch (c) a cat on a couch

Figure 10: Image and text prediction visualization. From left to right: masked input, prediction and the ground truth. Note that target colors are normalized per-patch and ground-truth patches are included in the prediction column. <> is a mask token.

t-SNE Section 4.2 shows the T-SNE visualization of CLIP and MaskCLIP. The labels are placed at the median of all embeddings that have the same ground-truth class. No visible improvement or deterioration is visible when comparing MaskCLIP versus CLIP, but related subjects are close to each other in both models.



(a) CLIP

(b) MaskCLIP

Figure 11: T-SNE on the embedded images for the first 50 ImageNet classes.

4.3 Ablation Study

To show that all our deviations from related work are required, we ablate these changes and show how the performance results are affected. All ablations are performed on an image encoder based on ViT-B/16, with a 12-layer, text encoder with 8 attention heads and 512 embedding dimensions. We use a noisy web-crawled dataset of over 1.4 billion image and text pairs that is composed of LAION-400M (Schuhmann et al., 2021), CC3M (Sharma et al., 2018), CC12M (Changpinyo et al., 2021) and internal datasets. This bigger dataset prevents us from comparing ablation results with our best result, so we stick to comparing relative performance between ablations. We evaluate ablations at 60k steps as the order of model performance does often not change afterwards.

Task combination strategy: We found the task combination strategy to be important for zero-shot performance, where using all tasks with equal weights would hurt all tasks. We show a different task combination strategy hurts zero-shot performance and leaves a gap between MaskCLIP and CLIP. Table 6 shows ablation results. Multiplying the generative image loss by 0.1 and the generative text loss by 0.05 only during global contrastive loss, was found to give the best performance. By scaling down the generative loss, down-stream linear-probing performance is hurt, but zero-shot performance improves. Using the local warm-up steps to kick-off the generative task without scaling it down helps for both zero-shot as down-stream linear-probing on VTAB tasks.

Following Alayrac et al. (2022), we always accumulate the losses.

Local		Global		ImageNet		VTAB
Image	Text	Image	Text	30K	60K	
1	1	1	1	31.3	35.3	71.5
1	1	0.1	0.1	42.6	47.2	72.0
1	1	0.05	0.1	43.0	47.1	72.1
1	1	0.1	0.05	**44.9**	**49.2**	**72.9**
1	1	0.05	0.05	44.3	48.2	72.1
2	2	1	1	31.7	35.3	71.9
2	2	0.1	0.05	44.5	48.7	71.9

Table 6: Multi-task weighting ablations, referring to w_i and w_t in Equation (7). Scaling down the generative loss improves zero-shot performance, but hurts linear-probing results. A warm-up during local contrastive loss without scaling performs best.

Contrastive Head: While other contrastive works use either a separate `cls` token or Multi-Headed Attention (MAP), we note that using GAP helps the zero-shot performance in our multi-task setting. We evaluate the contrastive head on the zero-shot ImageNet performance, as we found this is a good indication for other benchmarks. Both a separate `cls` token and a Multi-Headed Attention pooling mechanism hurt 0-short performance. We argue that in this case, the contrastive task and

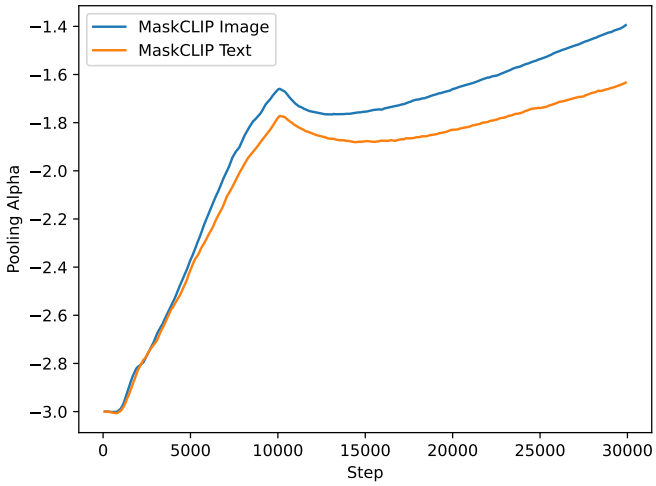


Figure 12: Use a learnable α that affects using the combination of GAP and MAP. In both modalities, it moves towards using MAP.

the generative task are fighting for capacity in the encoders, as there is opportunity for them to separate within the encoder. Using Global Average Pooling, the two tasks are forced to use the same model space. Especially in smaller architectures, like ViT-S, we see that using GAP brings a significant improvement in zero-shot performance. To allow the model to be more flexible, we also evaluate a contrastive head that has a learnable α per encoder. In this case, the `cls` token is computed as follows.

$$cls = layernorm(sigmoid(\alpha) * MAP + (1 - sigmoid(\alpha)) * GAP)$$

Figure 12 shows the change of α while training, it moves more towards MAP. Due to time-constraints, we only evaluated at 30k on zero-shot ImageNet, which resulted in an accuracy of 44.3. See Table 7 for the results when using MAP as the encoder head. We conclude that in the multi-task setting, MAP does not help in improving the zero-shot performance.

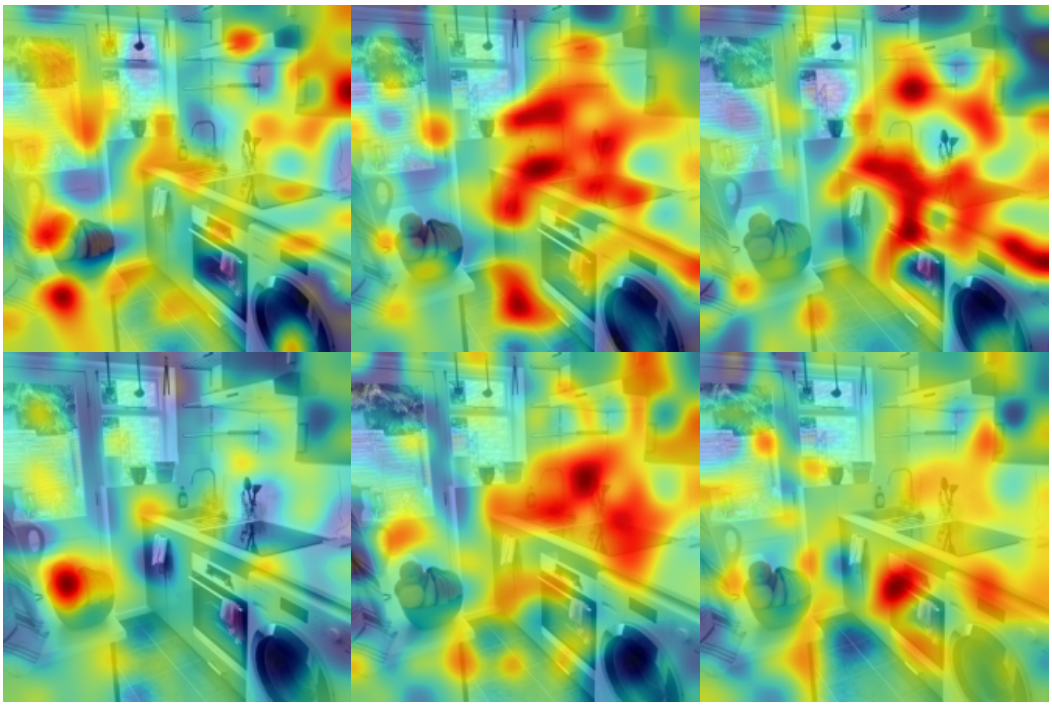
All of these alternatives perform worse than using GAP for both contrastive heads.

Contrastive Head	**ImageNet**		**VTAB**
	30K	**60K**	
MAP	41.0	45.9	67.5
GAP	**44.9**	**49.2**	**72.9**

Table 7: Using GAP in the encoders performs best in both zero-shot on ImageNet and linear-probing on VTAB.

Normalization: To improve our similarity-score based masking, we took a closer look at how we process our encoded patches to a `cls` token using GAP. In existing works, GAP is applied on the output of the encoders, where afterwards a LayerNorm (Ba et al., 2016) and linear projection is used to get the tokens from different modalities in the same space. We note that by using a LayerNorm, all patches are normalized to a sphere, which makes it unusable for per-patch similarity. Instead, we normalize the output of the encoders by dividing by the maximum patch norm. The maximum patch norm is computed by taking the maximum of the Frobenius norm on the hidden dimension of the patches. By doing this, normalization still allows the model to ignore patches in GAP by lowering their value. See Figure 13 for qualitative examples on the improved similarity-score maps for images.

Per-sample Masking Strategy: We can see in Table 8a that while zero-shot performance is improved by lowering the image masking ratio and while lowering the text masking ratio slightly improves VTAB results, using a high masking ratio for both modalities gets a good performance in both zero-shot and linear-probing. A higher masking ratio also reduces the amount of memory



(a) Oranges (b) Door (c) Oven

Figure 13: Similarity score maps for image patch and whole text (trained using layer-norm on top, max patch norm at the bottom). Models were trained for 2 epochs (60k steps).

Table 8: Ablation studies using ViT-B16 with ImageNet zero-shot and VTAB linear-probing performance using MaskCLIP with random masking. Best results are in bold, second-best are underlined.

(a) Per-sample masking ratios: How much is the input masked?

Masking Ratio		ImageNet		VTAB
Image	Text	30K	60K	
0.5	0.75	<u>44.6</u>	**49.6**	72.4
0.75	0.5	44.2	48.6	**73.0**
0.75	0.75	**44.9**	<u>49.2</u>	<u>72.9</u>

(b) Per-batch masking strategies: How often is a modality masked within a batch?

Masking Strategy		ImageNet		VTAB
Image	Text	30K	60K	
0.5	0.5	42.5	48.2	71.8
0.75	0.75	<u>43.8</u>	<u>48.6</u>	<u>72.7</u>
1	1	**44.9**	**49.2**	**72.9**

required in the encoders, as only visible patches are used as input. For smaller models, we have seen that lower masking ratios perform better, likely due to the difficulty of the task being related to the model capacity.

Per-batch Masking Strategy: Because of our high masking ratio, we also verify that masking both modalities at the same time still performs best. Table 8b shows how we mask a subset of the batch for a modality (denoted by a value smaller than 1). In case of 0.5, both modalities are never masked at the same time. In case of 0.75, we mask both modalities in half the batch, only a single modality in the other half. Finally, for values of 1, we always mask both modalities. Even though our masking ratio is high, masking both modalities at all times substantially outperforms any other strategy. We note that masking both modalities at all times also reduces memory usage, as only visible patches are used in the encoder.

Pre-training dataset: Due the data-efficiency of the generative task, we also show how the pre-training dataset quality and size influences results. While a large noisy dataset still performs best due the large number of domains it contains, we see that MaskCLIP performs increasingly better on smaller datasets than CLIP. We pre-train MaskCLIP using the publicly available dataset CC12M (Changpinyo et al., 2021) and we show that MaskCLIP outperforms CLIP and still gets reason-

able down-stream performance, with a zero-shot performance of 22.3 (+2.6%) on ImageNet, and an average linear-probing down-stream performance of 70.5 (+7.5%). All results can be found in Appendix D. More details about the training configuration can be found in Appendix C.

5 Discussion

We have showed how MaskCLIP improves data efficiency and visual grounding compared to only using a contrastive loss, by an improved down-stream performance with a lower pre-training time. While some architectural modifications were required, like a global average pooling layer to get a class token, it shows both good zero-shot performance and good down-stream performance on a variety of tasks. Next to that, we show initial research on how the similarity map can be used to guide masking to learn more important areas, without any manual labeling.

5.1 Research Questions

In this section, we will go over the different research questions and explicitly answer them.

How can we improve data efficiency and visual grounding for visual models trained on natural language supervision via a contrastive loss, by incorporating a masked patch prediction problem in the style of BERT/MAE?
We show that our proposed model, MaskCLIP, improves visual grounding. It outperforms CLIP, which only relies on the contrastive task, in both CLEVR and VQAv2 tasks that rely on attention to details. Next to this, visual inspection of the GradCAM shows an improved localization ability. Finally, similarity maps are less noisy in our MaskCLIP model, again showing the improved capability of MaskCLIP to relate parts of the image to text.

We also run MaskCLIP on CC12M, a publicly available dataset that is smaller, and show its improved data-efficiency. Even when using substantially less pre-training data, we get linear-probing results on VTAB that are competitive with results from CLIP pre-trained on the full-dataset.

How do we combine the high-level contrastive task with the low-level generative task, while making sure both tasks keep their beneficial properties?
To combine the two tasks and ensure both tasks align well, we need to weight the generative tasks lower when using global contrastive loss. We show that using the encoder heads that are used in CLIP hurts performance in our multi-task setup, and using global average pooling helps align the two tasks.

How do we guide the generative task to learn more details in important areas, without any additional data pre-processing or annotation?
By modifying the final normalization layer to a max-norm, we improve the patch-or-token-wise similarity scores, without the need of extra labeling. Using these improved similarity scores, we mask areas that are more similar to the whole other modality more often, therefore guiding the model towards learning more details of these areas. By doing so, we see zero-shot improvements, especially in whole-image classification tasks. However, it is tricky to design a masking strategy that improves performance across all tasks, especially on those tasks that require more attention to background details. We therefore note that in many cases, using random masking still remains the safest bet.

5.2 Limitations

Even though we have proposed an architecture that allows the generative and contrastive task to work together, we expect there exists a training technique that makes the two tasks work together in a more efficient way. We have decided on per-task weights using the total gradient norm in baselines, but a more dynamic approach that changes the weights as needed in case of architecture changes is desired.

We also note that even though we evaluated on a variety of different benchmarks to ensure fair comparison with baselines, more evaluation might be needed to get a better indication of weaker downstream tasks.

Finally, while similarity masking showed to be promising, it did not outperform random masking in all down-stream tasks. We have performed many experiments using different masking strategies, but we do expect a masking strategy to exist that outperforms random masking by improving the alignment between the two tasks. A balance between an improved masking strategy, based on e.g. patch-level similarities and random masking as a fallback, should benefit both zero-shot and linear-probing results.

5.3 Concurrent work

M3AE (Duan et al., 2022) and VL-BEiT (Bao et al., 2022) published concurrently to our work and both propose a single stream masked patch prediction modeling on both vision and language modalities, without contrastive loss. Similar to our work, they see that high masking ratios in both modalities improve performance. Next to that, SupMAE (Liang et al., 2022), confirmed our results that using Global Average Pooling (GAP) to extract a class token from the image encoder brings better results than using a separate class token.

5.4 Future work

MaskCLIP's multi-modal decoder might allow the model to move parts of its embedding logic from the encoders to the decoder. Intuitively, it would be better to force an improved embedding from the encoders, with a decoder only working on the generative task. Instead of using a full cross-modal decoder, it would be interesting to see how CoCa's (Yu et al., 2022) approach would perform using MaskCLIP's tasks. We would use frozen embeddings from the other modality and reduce the number of layers having access to this other embedding, forcing a higher quality representation of the other modality from the encoder to be useful.

Because of MaskCLIP's improved per-patch similarity score, we should investigate any potential for zero-shot object detection. Next to that, fine-tuning the per-patch similarity by using known object bounding boxes (either from a labeled dataset, or detected by a trained object detector) and their labels, we can exploit the good per-patch similarity for rough bounding box predictions (similar to (Yao et al., 2021)). Concurrent to our work, Hou et al. (2022) presented MILAN, a masked image pre-training setup that uses the CLIP Image encoder to guide masking. MaskCLIP's improved visual grounding could benefit MILAN, by guiding the masking to more accurate areas.

We see good down-stream performance on VQA tasks when exploiting the generative aspect of MaskCLIP. In MaskCLIP, all encoders and decoders are non-causal, which makes it harder for the model to generate full correct sentences. Using a causal encoder/decoder might make sense for more VQA targeted models.

6 Conclusion

We propose MaskCLIP, a multi-task foundational model that extends a contrastive-only model, by adding a generative task. Adding the masked patch prediction task forces the model to use local attention and we show an improved visual grounding capability in down-stream tasks. Moving to GAP instead of using MAP as the encoder head forces the model to share capacity and thus improve both tasks. Next to that, weighting the generative loss lower in the multi-task setting helps the model to keep its high zero-shot performance. MaskCLIP shows performance improvements in many zero-shot tasks, linear-probing tasks and VQA fine-tuning tasks. By exploiting the patch-wise similarity scores, we steer the model towards training the in-the-text mentioned parts of the image (and text that is visualized in the image) and thus guide it towards learning more important areas of the image/text. While this does improve zero-shot performance for whole-image classification, it is tricky to stop it biasing the generative decoder too far towards the contrastive task, concluding that in many cases a random masking strategy remains the safest bet.

Acknowledgments

This research was done with help from various colleagues from Apple. A big thanks to Albin Madappally Jose for his work on the creation of the large training dataset and the removal of all test images from this dataset. We thank Angelos Katharopoulos, Tom Nickson and Arthur van Hoff

for the engaging team brainstorms and we are grateful to Yinfei Yang for his advice on the project. Finally, a special thanks to Samuel Albanie from Cambridge University for his insights on related literature.

REFERENCES

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. CM3: A Causal Masked Multimodal Model of the Internet. *arXiv:2201.07520 [cs]*, January 2022. URL http://arxiv.org/abs/2201.07520. arXiv: 2201.07520.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning. Technical Report arXiv:2204.14198, arXiv, April 2022. URL http://arxiv.org/abs/2204.14198. arXiv:2204.14198 [cs] type: article.

Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. B2T2: Fusion of Detected Objects in Text for Visual Question Answering. *arXiv:1908.05054 [cs]*, November 2019. URL http://arxiv.org/abs/1908.05054. arXiv: 1908.05054.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. pp. 2425–2433, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/Antol_VQA_Visual_Question_ICCV_2015_paper.html.

Tarik Arici, Mehmet Saygin Seyfioglu, Tal Neiman, Yi Xu, Son Train, Trishul Chilimbi, Belinda Zeng, and Ismail Tutar. MLIM: Vision-and-Language Model Pre-training with Masked Language and Image Modeling. *arXiv:2109.12178 [cs]*, September 2021. URL http://arxiv.org/abs/2109.12178. arXiv: 2109.12178.

Sara Atito, Muhammad Awais, Ammarah Farooq, Zhenhua Feng, and Josef Kittler. MC-SSL0.0: Towards Multi-Concept Self-Supervised Learning. *arXiv:2111.15340 [cs]*, November 2021. URL http://arxiv.org/abs/2111.15340. arXiv: 2111.15340.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. Technical Report arXiv:1607.06450, arXiv, July 2016. URL http://arxiv.org/abs/1607.06450. arXiv:1607.06450 [cs, stat] type: article.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. data2vec: A General Framework for Self-supervised Learning in Speech, Vision and Language. pp. 13, January 2022.

Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT Pre-Training of Image Transformers. *arXiv:2106.08254 [cs]*, June 2021. URL http://arxiv.org/abs/2106.08254. arXiv: 2106.08254.

Hangbo Bao, Wenhui Wang, Li Dong, and Furu Wei. VL-BEiT: Generative Vision-Language Pre-training. Technical Report arXiv:2206.01127, arXiv, June 2022. URL http://arxiv.org/abs/2206.01127. arXiv:2206.01127 [cs] type: article.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya

Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the Opportunities and Risks of Foundation Models. *arXiv:2108.07258 [cs]*, August 2021. URL http://arxiv.org/abs/2108.07258. arXiv: 2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. GPT-3: Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, July 2020. URL http://arxiv.org/abs/2005.14165. arXiv: 2005.14165.

Shuhao Cao, Peng Xu, and David A. Clifton. How to Understand Masked Autoencoders. *arXiv:2202.03670 [cs]*, February 2022. URL http://arxiv.org/abs/2202.03670. arXiv: 2202.03670.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. SwAV: Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. *arXiv:2006.09882 [cs]*, January 2021a. URL http://arxiv.org/abs/2006.09882. arXiv: 2006.09882.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. DINO: Emerging Properties in Self-Supervised Vision Transformers. *arXiv:2104.14294 [cs]*, May 2021b. URL http://arxiv.org/abs/2104.14294. arXiv: 2104.14294.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. Technical Report arXiv:2102.08981, arXiv, March 2021. URL http://arxiv.org/abs/2102.08981. arXiv:2102.08981 [cs] type: article.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. SimCLR: A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1597–1607. PMLR, November 2020a. URL https://proceedings.mlr.press/v119/chen20j.html. ISSN: 2640-3498.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. SimCLRv2: Big Self-Supervised Models are Strong Semi-Supervised Learners. *arXiv:2006.10029 [cs, stat]*, October 2020b. URL http://arxiv.org/abs/2006.10029. arXiv: 2006.10029.

Xinlei Chen and Kaiming He. SimSiam: Exploring Simple Siamese Representation Learning. *arXiv:2011.10566 [cs]*, November 2020. URL http://arxiv.org/abs/2011.10566. arXiv: 2011.10566.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO Captions: Data Collection and Evaluation Server. *arXiv:1504.00325 [cs]*, April 2015. URL http://arxiv.org/abs/1504.00325. arXiv: 1504.00325.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. MoCov2: Improved Baselines with Momentum Contrastive Learning. *arXiv:2003.04297 [cs]*, March 2020c. URL http://arxiv.org/abs/2003.04297. arXiv: 2003.04297.

Xinlei Chen, Saining Xie, and Kaiming He. MoCov3: An Empirical Study of Training Self-Supervised Vision Transformers. *arXiv:2104.02057 [cs]*, August 2021. URL http://arxiv.org/abs/2104.02057. arXiv: 2104.02057.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: UNiversal Image-TExt Representation Learning. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pp. 104–120, Cham, September 2020d. Springer International Publishing. ISBN 978-3-030-58577-8. doi: 10.1007/978-3-030-58577-8_7. URL https://link.springer.com/chapter/10.1007/978-3-030-58577-8_7.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. Technical Report arXiv:2003.10555, arXiv, March 2020. URL http://arxiv.org/abs/2003.10555. arXiv:2003.10555 [cs] type: article.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848. ISSN: 1063-6919.

Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. *arXiv:2006.06666 [cs]*, September 2021. URL http://arxiv.org/abs/2006.06666. arXiv: 2006.06666.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. URL http://arxiv.org/abs/1810.04805. arXiv: 1810.04805.

Jianfeng Dong, Yabing Wang, Xianke Chen, Xiaoye Qu, Xirong Li, Yuan He, and Xun Wang. Reading-strategy Inspired Visual Representation Learning for Text-to-Video Retrieval. *arXiv:2201.09168 [cs]*, January 2022. URL http://arxiv.org/abs/2201.09168. arXiv: 2201.09168.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. ViT: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv:2010.11929 [cs]*, June 2021. URL http://arxiv.org/abs/2010.11929. arXiv: 2010.11929.

Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, Zicheng Liu, and Michael Zeng. METER: An Empirical Study of Training End-to-End Vision-and-Language Transformers. *arXiv:2111.02387 [cs]*, November 2021. URL http://arxiv.org/abs/2111.02387. arXiv: 2111.02387.

Jiali Duan, Liqun Chen, Son Tran, Jinyu Yang, Yi Xu, Belinda Zeng, and Trishul Chilimbi. Multi-modal Alignment using Representation Codebook. *arXiv:2203.00048 [cs]*, March 2022. URL http://arxiv.org/abs/2203.00048. arXiv: 2203.00048.

Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. SplitMask: Are Large-scale Datasets Necessary for Self-Supervised Pre-training? *arXiv:2112.10740 [cs]*, December 2021. URL http://arxiv.org/abs/2112.10740. arXiv: 2112.10740.

Yuying Ge, Yixiao Ge, Xihui Liu, Alex Jinpeng Wang, Jianping Wu, Ying Shan, Xiaohu Qie, and Ping Luo. MILES: Visual BERT Pre-training with Injected Language Semantics for Video-text Retrieval. *arXiv:2204.12408 [cs]*, April 2022. URL http://arxiv.org/abs/2204.12408. arXiv: 2204.12408.

Xinyang Geng, Hao Liu, Lisa Lee, Dale Schuurmans, Sergey Levine, and Pieter Abbeel. M3AE: Multimodal Masked Autoencoders Learn Transferable Representations. Technical Report arXiv:2205.14204, arXiv, May 2022. URL http://arxiv.org/abs/2205.14204. arXiv:2205.14204 [cs] type: article.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv:1706.02677 [cs]*, April 2018. URL http://arxiv.org/abs/1706.02677. arXiv: 1706.02677.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. VQA2: Making the v in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. pp. 6904–6913, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Goyal_Making_the_v_CVPR_2017_paper.html.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. BYOL: Bootstrap your own latent: A new approach to self-supervised Learning. *arXiv:2006.07733 [cs, stat]*, September 2020. URL http://arxiv.org/abs/2006.07733. arXiv: 2006.07733.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. MoCov1: Momentum Contrast for Unsupervised Visual Representation Learning. pp. 9729–9738, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/He_Momentum_Contrast_for_Unsupervised_Visual_Representation_Learning_CVPR_2020_paper.html.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. MAE: Masked Autoencoders Are Scalable Vision Learners. *arXiv:2111.06377 [cs]*, November 2021. URL http://arxiv.org/abs/2111.06377. arXiv: 2111.06377.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. *arXiv:2009.03300 [cs]*, January 2021. URL http://arxiv.org/abs/2009.03300. arXiv: 2009.03300.

Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. MILAN: Masked Image Pretraining on Language Assisted Representation. Technical Report arXiv:2208.06049, arXiv, August 2022. URL http://arxiv.org/abs/2208.06049. arXiv:2208.06049 [cs] type: article.

Jeremy Howard and Sebastian Ruder. Universal Language Model Fine-tuning for Text Classification. *arXiv:1801.06146 [cs, stat]*, May 2018. URL http://arxiv.org/abs/1801.06146. arXiv: 1801.06146.

Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing Out of the Box: End-to-End Pre-Training for Vision-Language Representation Learning. pp. 12976–12985, April 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Huang_Seeing_Out_of_the_Box_End-to-End_Pre-Training_for_Vision-Language_Representation_CVPR_2021_paper.html.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. ALIGN: Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. *arXiv:2102.05918 [cs]*, June 2021. URL http://arxiv.org/abs/2102.05918. arXiv: 2102.05918.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1988–1997, 2017. doi: 10.1109/CVPR.2017.215. URL https://ieeexplore.ieee.org/document/8099698.

Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision. *arXiv:2102.03334 [cs, stat]*, June 2021. URL http://arxiv.org/abs/2102.03334. arXiv: 2102.03334.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. Technical Report arXiv:1412.6980, arXiv, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv:1412.6980 [cs] type: article.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big Transfer (BiT): General Visual Representation Learning. Technical Report arXiv:1912.11370, arXiv, May 2020. URL http://arxiv.org/abs/1912.11370. arXiv:1912.11370 [cs] type: article.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv:1910.13461 [cs, stat]*, October 2019. URL http://arxiv.org/abs/1910.13461. arXiv: 1910.13461.

Chenliang Li, Ming Yan, Haiyang Xu, Fuli Luo, Wei Wang, Bin Bi, and Songfang Huang. SemVLP: Vision-Language Pre-training by Aligning Semantics at Multiple Levels. *arXiv:2103.07829 [cs]*, March 2021a. URL http://arxiv.org/abs/2103.07829. arXiv: 2103.07829.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq Joty, Caiming Xiong, and Steven Hoi. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. *arXiv:2107.07651 [cs]*, October 2021b. URL http://arxiv.org/abs/2107.07651. arXiv: 2107.07651.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv:1908.03557 [cs]*, August 2019. URL http://arxiv.org/abs/1908.03557. arXiv: 1908.03557.

Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking Detection Transfer Learning with Vision Transformers. *arXiv:2111.11429 [cs]*, November 2021c. URL http://arxiv.org/abs/2111.11429. arXiv: 2111.11429.

Feng Liang, Yangguang Li, and Diana Marculescu. SupMAE: Supervised Masked Autoencoders Are Efficient Vision Learners. Technical Report arXiv:2205.14540, arXiv, May 2022. URL http://arxiv.org/abs/2205.14540. arXiv:2205.14540 [cs] type: article.

Ye Liu, Siyuan Li, Yang Wu, Chang Wen Chen, Ying Shan, and Xiaohu Qie. UMT: Unified Multi-modal Transformers for Joint Video Moment Retrieval and Highlight Detection. *arXiv:2203.12745 [cs]*, March 2022. URL http://arxiv.org/abs/2203.12745. arXiv: 2203.12745.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv:2103.14030 [cs]*, August 2021. URL http://arxiv.org/abs/2103.14030. arXiv: 2103.14030.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. *arXiv:1608.03983 [cs, math]*, May 2017. URL http://arxiv.org/abs/1608.03983. arXiv: 1608.03983.

Ilya Loshchilov and Frank Hutter. AdamW: Decoupled Weight Decay Regularization. *arXiv:1711.05101 [cs, math]*, January 2019. URL http://arxiv.org/abs/1711.05101. arXiv: 1711.05101.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. *arXiv:1908.02265 [cs]*, August 2019. URL http://arxiv.org/abs/1908.02265. arXiv: 1908.02265.

Jianjie Luo, Yehao Li, Yingwei Pan, Ting Yao, Hongyang Chao, and Tao Mei. CoCo-BERT: Improving Video-Language Pre-training with Contrastive Cross-modal Matching and Denoising. *arXiv:2112.07515 [cs]*, December 2021. URL http://arxiv.org/abs/2112.07515. arXiv: 2112.07515.

Thomas Mensink, Jasper Uijlings, Alina Kuznetsova, Michael Gygli, and Vittorio Ferrari. Factors of Influence for Transfer Learning across Diverse Appearance Domains and Task Types. *IEEE transactions on pattern analysis and machine intelligence*, PP, November 2021. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3129870.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748 [cs, stat]*, January 2019. URL http://arxiv.org/abs/1807.03748. arXiv: 1807.03748.

Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. VALSE: A Task-Independent Benchmark for Vision and Language Models Centered on Linguistic Phenomena. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8253–8280, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.acl-long.567.

Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc V. Le. Combined Scaling for Open-Vocabulary Image Classification. Technical Report arXiv:2111.10050, arXiv, April 2022. URL http://arxiv.org/abs/2111.10050. arXiv:2111.10050 [cs] type: article.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models. pp. 2641–2649, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/Plummer_Flickr30k_Entities_Collecting_ICCV_2015_paper.html.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018. Publisher: Technical report, OpenAI.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. GPT-2: Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. CLIP: Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020 [cs]*, February 2021. URL http://arxiv.org/abs/2103.00020. arXiv: 2103.00020.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, January 2015. ISSN 0893-6080. doi: 10.1016/j.neunet.2014.09.003. URL https://www.sciencedirect.com/science/article/pii/S0893608014002135.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs. *arXiv:2111.02114 [cs]*, November 2021. URL http://arxiv.org/abs/2111.02114. arXiv: 2111.02114.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. pp. 14, August 2022.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *International Journal of Computer Vision*, 128(2):336–359, February 2020. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-019-01228-7. URL http://arxiv.org/abs/1610.02391. arXiv:1610.02391 [cs].

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. GCC: Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238. URL http://aclweb.org/anthology/P18-1238.

Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How Much Can CLIP Benefit Vision-and-Language Tasks? *arXiv:2107.06383 [cs]*, July 2021. URL http://arxiv.org/abs/2107.06383. arXiv: 2107.06383.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A Foundational Language And Vision Alignment Model. *arXiv:2112.04482 [cs]*, February 2022. URL http://arxiv.org/abs/2112.04482. arXiv: 2112.04482.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. *arXiv:1908.08530 [cs]*, February 2020. URL http://arxiv.org/abs/1908.08530. arXiv: 1908.08530.

Yihong Sun, Adam Kortylewski, and Alan Yuille. Amodal Segmentation through Out-of-Task and Out-of-Distribution Generalization with a Bayesian Model. Technical Report arXiv:2010.13175, arXiv, July 2022. URL http://arxiv.org/abs/2010.13175. arXiv:2010.13175 [cs] type: article.

Hao Tan and Mohit Bansal. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. *arXiv:1908.07490 [cs]*, December 2019. URL http://arxiv.org/abs/1908.07490. arXiv: 1908.07490.

Wilson L. Taylor. "Cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953. Publisher: SAGE Publications Sage CA: Los Angeles, CA.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality. pp. 11.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, \Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *arXiv:1804.07461 [cs]*, February 2019. URL http://arxiv.org/abs/1804.07461. arXiv: 1804.07461.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple Visual Language Model Pretraining with Weak Supervision. *arXiv:2108.10904 [cs]*, August 2021. URL http://arxiv.org/abs/2108.10904. arXiv: 2108.10904.

Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. MVP: Multimodality-guided Visual Pre-training. *arXiv:2203.05175 [cs]*, March 2022. URL http://arxiv.org/abs/2203.05175. arXiv: 2203.05175.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early Convolutions Help Transformers See Better. Technical Report arXiv:2106.14881, arXiv, October 2021. URL http://arxiv.org/abs/2106.14881. arXiv:2106.14881 [cs] type: article.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A Simple Framework for Masked Image Modeling. *arXiv:2111.09886 [cs]*, November 2021. URL http://arxiv.org/abs/2111.09886. arXiv: 2111.09886.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. Pre-layernorm: On Layer Normalization in the Transformer Architecture. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 10524–10533. PMLR, November 2020. URL https://proceedings.mlr.press/v119/xiong20b.html. ISSN: 2640-3498.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained Interactive Language-Image Pre-Training. *arXiv:2111.07783 [cs]*, November 2021. URL http://arxiv.org/abs/2111.07783. arXiv: 2111.07783.

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-ViL: Knowledge Enhanced Vision-Language Representations Through Scene Graph. June 2020. URL http://128.84.4.18/abs/2006.16934.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive Captioners are Image-Text Foundation Models. *arXiv:2205.01917 [cs]*, May 2022. URL http://arxiv.org/abs/2205.01917. arXiv: 2205.01917.

Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A New Foundation Model for Computer Vision. *arXiv:2111.11432 [cs]*, November 2021. URL http://arxiv.org/abs/2111.11432. arXiv: 2111.11432.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. *arXiv:2103.03230 [cs, q-bio]*, June 2021. URL http://arxiv.org/abs/2103.03230. arXiv: 2103.03230.

Yan Zeng, Xinsong Zhang, and Hang Li. X:VLM: Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts. *arXiv:2111.08276 [cs]*, February 2022. URL http://arxiv.org/abs/2111.08276. arXiv: 2111.08276.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. VTAB: A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark. *arXiv:1910.04867 [cs, stat]*, February 2020. URL http://arxiv.org/abs/1910.04867. arXiv: 1910.04867.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers. *arXiv:2106.04560 [cs]*, June 2021. URL http://arxiv.org/abs/2106.04560. arXiv: 2106.04560.

Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, and Jianfeng Gao. RegionCLIP: Region-based Language-Image Pretraining. *arXiv:2112.09106 [cs]*, December 2021. URL http://arxiv.org/abs/2112.09106. arXiv: 2112.09106.

A Related work

In this section, we will go more in-depth on some of the related work.

BERT

Devlin et al. (2019) introduces a non-autoregressive encoder-only model that is used in the text domain. Input is masked, sometimes replaced by a learnable mask token, a random other token or not replaced at all. The full input is used by an encoder that needs to predict the masked tokens. They show impressive results on both smaller training datasets, and show that scaling up improves results.

ViT

In ViT (Dosovitskiy et al., 2021), an image $x \in \mathbb{R}^{H \times W \times C}$ (where H and W refer to the height and width and C to the channels) is split into non-overlapping patches $p \in \mathbb{R}^{P^2 C}$ (where P is the patch size). The patches are linearly transformed to the embedding dimension D. A positional encoding is added to the transformed patches, to provide location information to the transformer encoder. The transformer encoder is based on Vaswani et al. (2017), in which each layer consist of a self-attention module, skip connection and multilayer perception. In MAE (He et al., 2021), the input image x is still split into non-overlapping patches p, linearly transformed to D-dimensional patch embeddings and added with a positional encoding. However, instead of using the full set of patches as input for the transformer encoder, only a subset (the non-masked patches) is used.

MAE

He et al. (2021) introduces a new non-autoregressive auto-encoder model that is used in the image domain. They propose an encoder-decoder structure that pre-trains by predicting masked patches. The input image x is still split into non-overlapping patches p, linearly transformed to D-dimensional patch embeddings and added with a positional encoding. However, instead of using the full set of patches as input for the transformer encoder, only a subset (the non-masked patches) is used. They use a masking high masking ratio of 75% and therefore lower the memory requirements for the encoder. They show good downstream results using a small amount of data. In down-stream tasks, only the encoder is used and fine-tuned.

B Evaluation benchmarks

As we make use of two benchmarks with sub-tasks, we will go over them in more detail.

B.1 VALSE

VALSE (Parcalabescu et al., 2022) is a zero-shot VQA benchmark. Each image has a correct description and a foil. Out of the given two texts, the model needs to predict which is the correct description.

It contains 6 different tasks. An example per task:

Existence There are [no animals]/[animals] shown.

Plurality A small copper vase with [some flowers]/[exactly one flower] in it.

Counting There are [four]/[six] zebras.

Relations A cat plays with a pocket knife [on]/[underneath] a table.

Actions A [man]/[woman] shouts at a [woman]/[man].

Coreference Buffalos walk along grass. Are they in a zoo? [No]/[Yes].

Fore more details on these examples, see Table 1 of Parcalabescu et al. (2022).

B.2 VTAB

VTAB (Zhai et al., 2020) is an image classification benchmark. It contains three categories, with a total of 19 tasks. The Diabetic Retinopathy and Sun397 tasks are not used due to time constraints.

Caltech101 102 classes, natural image classification.

CIFAR-100 100 classes, natural image classification. Low resolution, 32x32.

DTD Describable Textures Dataset, 47 classes, natural image classification. Inspired by human perception, e.g. 'wrinkled' and 'flecked'.

Flowers102 102 classes, natural image classification. 10 images per class.

Pets 37 classes, natural image classification.

SVHN Street View House Number, 10 classes (number 0-9). Low resolution, 32x32.

EuroSAT 10 classes, specialized image classification. Satelite pictures.

Resisc45 Remote Sensing Image Scene Classification, 45 classes, specialized image classification.

Patch Camelyon 2 classes (presence of metastatic issue or not), specialized image classification.

Clevr/count 8 classes, structured image classification. Count the number of objects in the scene.

Clevr/distance 6 classes, structured image classification. Predict the distance to the closest object in the scene.

dSprites/location 16 classes, structured image classification. Predict the x position of the sprite.

dSprites/orientation 16 classes, structured image classification. Predict the orientation of the sprite.

SmallNORB/azimuth 18 classes (0 to 340, every 20 degrees), structured image classification.

SmallNORB/elevation 18 classes (0 to 340, every 20 degrees), structured image classification.

DMLab Deepmind Lab, 6 classes, structured image classification. Predict distance between agent and objects present.

KITTI/distance 4 classes, structured image classification. Predict the distance to the closest vehicle in the scene.

See Table 2 of Appendix A in (Zhai et al., 2020) for more details.

C TRAINING CONFIGURATIONS

Table 9 shows the used hyperparameter setup for pre-training of both CLIP and MaskCLIP. Using warmup steps, where only a local contrastive loss is used instead of a global contrastive loss, helps the model to converge faster. In MaskCLIP, we simply sum the contrastive and generative losses when doing local contrastive loss, but multiply the generative image loss by 0.05 and the generative text loss by 0.1 when doing global contrastive loss, referring to w_i and w_t from Equation (7) respectively.

We use the same training configuration for the pre-training on CC12M (Changpinyo et al., 2021), with a few changes due the reduced number of total steps. We use 500 local contrastive loss steps, and only 200 warmup steps for the cosine learning rate scheduler.

D CC12M PRE-TRAINING

In this section, we will show all evaluation results when pre-training MaskCLIP and CLIP only on CC12M (Changpinyo et al., 2021). Table 10 shows MaskCLIP outperforms CLIP on zero-shot object retrieval and classification. We can also see a slight improvement in the zero-shot VQA benchmark VALSE. Winoground text performance is lower for MaskCLIP than CLIP, potentially indicating that more background information in the embedding hurts this task.

When comparing individual zero-shot VALSE tasks, as shown in Table 11, we can see an overall performance improvement when using MaskCLIP, except for the actant-swap task.

Component	Parameter	Value
Image Encoder	Depth	12
	Width	768
	MLP Heads	12
Text Encoder	Depth	12
	Width	512
	MLP Heads	8
Decoder	Depth	8
	Width	512
	MLP Heads	8
Model	Weight decay	0.1
	Base LR	5e-04
	LR Schedule	Cosine decay (Loshchilov & Hutter, 2017)
	LR Warmup steps	1000
	Local contrastive steps (Goyal et al., 2018)	10000
	Batch size	256
	Optimizer	AdamW (Loshchilov & Hutter, 2019)
	Optimizer momentum	$\beta 1$, $\beta 2$ = 0.9, 0.98
	Augmentation	RandomResizedCrop

Table 9: Parameters used for pre-training

Table 10: Zero-shot results, trained on CC12M. I→T: Image to Text, T→I: Text to Image.

Model	COCO		FLICKR		Winoground			ImageNet	VALSE
	I→T	T→I	I→T	T→I	Text	Image	Group		
CLIP	18.3	12.3	32.6	23.6	**19.5**	**9.5**	**5.5**	19.7	54.3
MaskCLIP	**23.5**	**15.2**	**40.2**	**29.7**	**19.5**	6.8	4.3	**22.3**	**54.5**

Table 11: Zero-shot results on VALSE, trained on CC12M.

Model	VALSE						
	actant-swap	action-replacement	counting	existence	plurals	relations	average
CLIP	**54.9**	58.2	51.5	54.7	57.2	53.1	54.3
MaskCLIP	51.3	**64.7**	**51.7**	**55.8**	**58.9**	**54.0**	**54.5**

Table 12 shows that MaskCLIP outperforms CLIP on the occluded object detection task. Note that the difference in performance for top-5 COCOA is lower than shown in Table 2, when trained on the large dataset.

Table 12: Zero-shot results on COCOA, trained on CC12M. The different levels refer to occlusion levels of the target object. A low level has lower occlusion.

Model	COCOA									
	Top-1					Top-5				
	level 0	level 1	level 2	level 3	average	level 0	level 1	level 2	level 3	average
CLIP	21.5	17.6	15.4	10.7	16.3	37.7	35.3	**34.2**	25.7	33.2
MaskCLIP	**25.7**	**24.6**	**18.0**	**14.1**	**20.6**	**42.1**	**39.2**	33.1	**28.6**	**35.8**

Finally, Table 13 shows the linear-probing results on most VTAb tasks. Again, we see that MaskCLIP outperforms CLIP on all tasks, supporting its argued data-efficiency.

Table 13: Linear-probe results on most VTAB tasks, trained on CC12M.

	VTAB																
	Natural						**Specialized**			**Structured**							
Model	caltech101	cifar-100	dtd	oxford-flowers102	oxford-iiit-pets	svhn	Eurosat	Patch Camelyon	Resisc45	clevr-closest	clevr-count	dmlab	dsprites-orientation	dsprites-xpos	kitti-closest-vehicle	smallnorb-azimuth	smallnorb-elevation
CLIP	85.4	58.6	63.1	84.6	73.2	50.4	94.3	81.2	84.9	55.3	56.8	44.1	50.1	60.7	45.4	36.6	47.1
MaskCLIP	**89.5**	**66.8**	**70.3**	**90.9**	**77.4**	**63.1**	**96.5**	**81.6**	**90.8**	**61.8**	**69.4**	**48.2**	**57.9**	**77.2**	**47.3**	**50.9**	**59.3**

E LINEAR-PROBING DETAILS

Following Radford et al. (2021), we take the output of the encoders before any linear projection to the shared contrastive embedding space. Different from Radford et al. (2021), we do apply L2 normalization to follow our pre-training setup. A logistic regression classifier is trained using scikit-learn's[1] L-BFGS implementation, with a maximum of 1000 iterations. The L2 regularization strength (argument C in scikit), is determined using a hyperparameter sweep on the validation sets. Similar to Radford et al. (2021), the strength is found by a binary search using $[10^{-6}, 10^{-4}, 10^{-2}, 1, 10^{2}, 10^{4}, 10^{6}]$ as starting values. Each iteration, the interval is halved around the peak until it reaches a resolution of 8 steps per decade. Hyperparameter search is always done using a validation set, when available. If no validation set is defined in the dataset, part of the training dataset is used. For the final result, both the training and validation split are used for training and accuracy is computed over the unseen test split.

[1] https://scikit-learn.org/stable/

F GENERATIVE RESULTS

This section will show various other generative examples. Table 14 shows masked text input and its prediction. Figure 14 shows the corresponding masked image and its prediction. Both modalities are masked and predicted at the same time. While image details are lost, the overall structure is regenerated. We can see cross-modality usage in the text domain, where e.g. the color of the dog or the activity of the cat is determined through the image. Because no causual encoder-decoder structure is used, sentence structure quality varies a lot and sentences are not consistent. Nevertheless, predictions are still close to correct sentences.

To show the effect of the individual decoder layers, we follow Cao et al. (2022) and visualize the internal representations in Figure 15. We can see shapes taking form after a few layers.

Masked input	Predicted	Ground truth
<><>on<>couch	cat sleeping on a couch	a cat on a couch
<><>sleeping	red dog sleeping	a dog sleeping
<>happy<>with<><><><>	a happy dog with a tennis tennis	a happy dog with a lot of toys
a<><><>in<><>	a kitchen or kitchen in the apartment	a bowl of fruit in a kitchen

Table 14: Text prediction visualization. <> is a masked token. These texts are paired with images shown in Figure 14. Note that the model is only trained to predict the masked patches.

G VQA FINE-TUNING

We evaluate two VQA benchmarks (Johnson et al. (2017) and Goyal et al. (2017)) using three different techniques.

G.1 LINEAR-PROBING

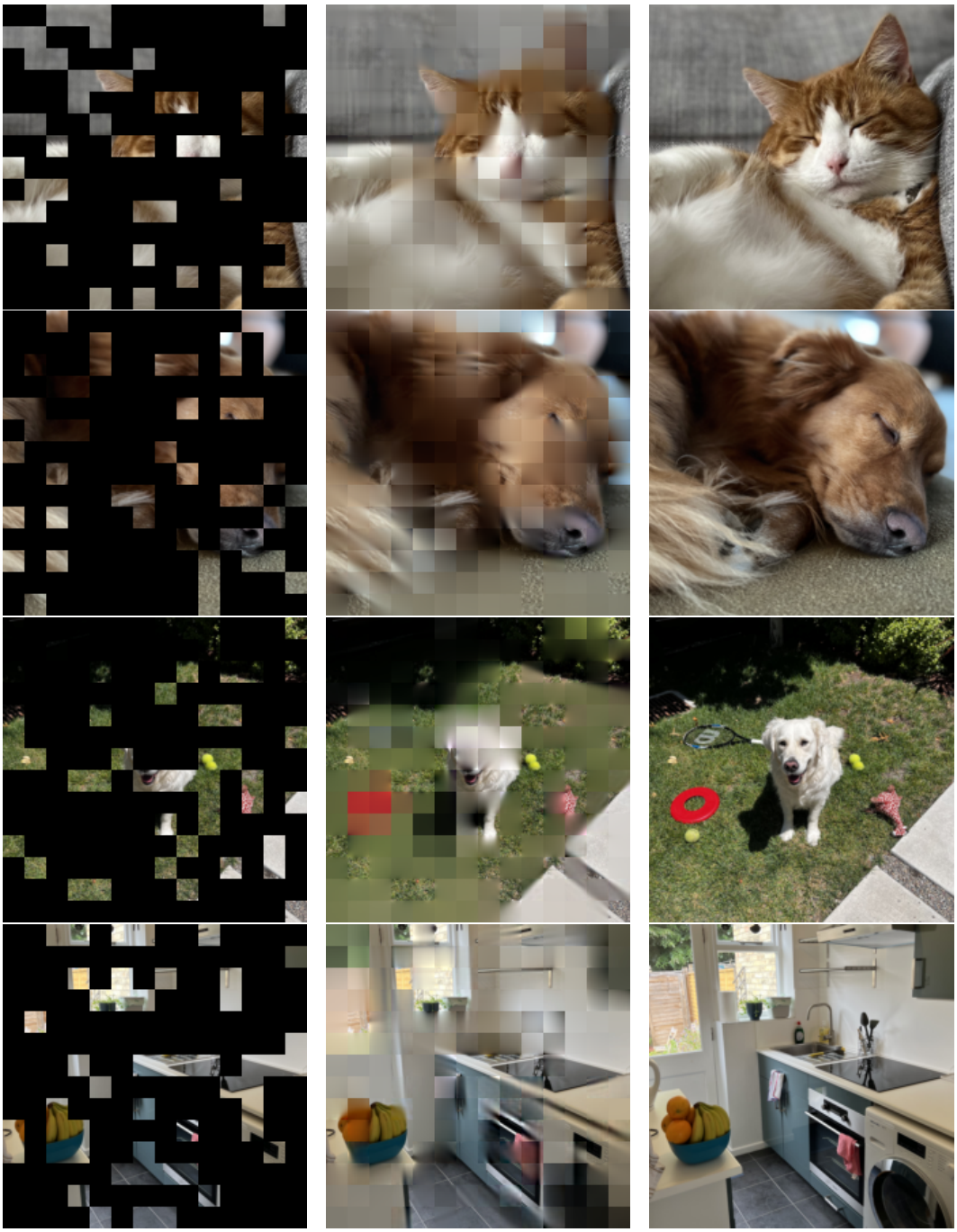
Figure 16 shows the linear-probing architecture for VQA. This is the simplest setup and only requires a linear head to be trained on top of the frozen encoders. All possible answers are mapped to classes to make it a classification task. We concatenate the output of the two `cls` tokens of the encoders as input for a linear projection to the required classes. This follows the same hyperparameter search as in Appendix E.

G.2 DECODER FINE-TUNING

While linear-probing is convenient, fine-tuning a larger set of components on top of the frozen encoders has been shown to outperform linear-probing significantly (Zeng et al., 2022). We fine-tune the decoder in both a classification way as a generative way.

Classification Similar to linear-probing, we convert the set of possible answers to classes. Following our pre-training setup, we concatenate the image and text embeddings, add positional encoding and a modal specific token before using it as input in the decoder. The `BOS` token is used as an output token and linearly projected to the possible classes. See Figure 17. We use both the exact decoder setup, and a simplified version that does not re-add positional encoding and modal specific tokens. We find that for fine-tuning, slightly better performance is found by using this simplified decoder.

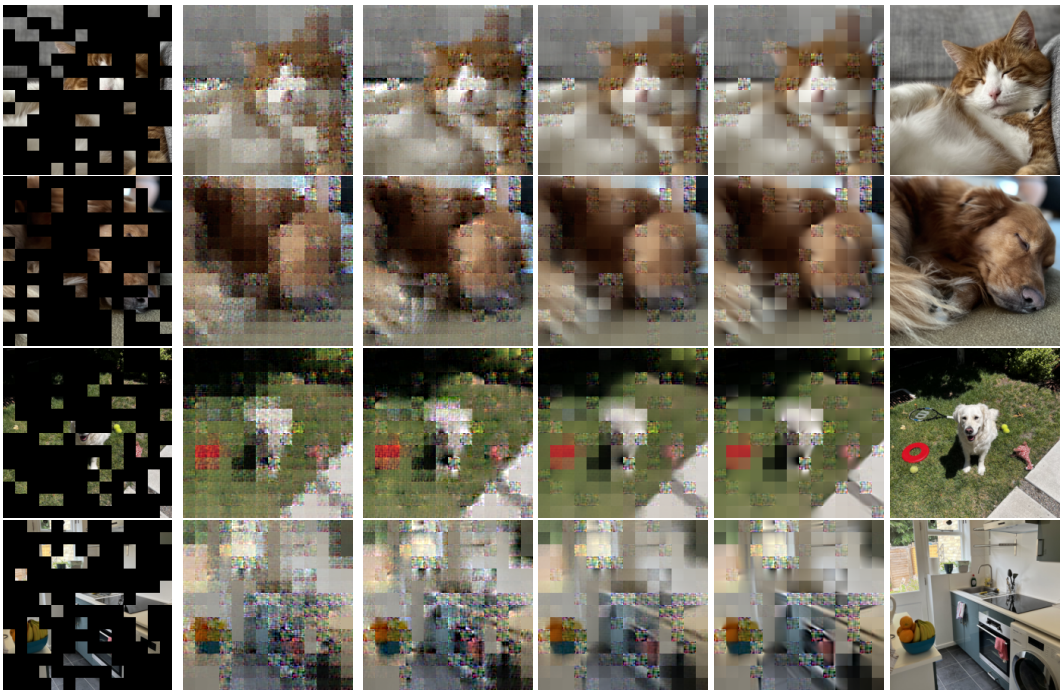
Generation We also exploit the generative task of the MaskCLIP model by fine-tuning predicting the raw text tokens of the answer. As can be seen in Figure 18, we concatenate the answer to the question to define the ground truth. We then mask the answer and the `EOS` token of the input. The image is not masked. The model is fine-tuned to predict the answer and the `EOS` token, as the answer can be of variable length. A predicted answer is defined correct if all answer tokens, including the `EOS`, match the ground truth. Since the transformer models are non-causal, it is hard for the model to form correct answer sentences. To improve performance, we re-run the decoder for every token that needs to be predicted, replacing masked tokens as they are predicted. We notice that if we do not do this, answers that require multiple tokens only have the first token completely correct, but keep predicting the first token at all places.



(a) Masked input (b) Prediction (c) Ground truth

Figure 14: Image prediction visualization. Note that target colors are normalized per-patch and ground-truth patches are included in the prediction column.

In both classification and generation, we train using AdamW. Encoders are always frozen and we apply layer-wise learning rate decay (Clark et al., 2020) on the decoder, following Bao et al. (2021). See Table 15 for all training details.



(a) Masked input (b) Layer 1 (c) Layer 2 (d) Layer 5 (e) Layer 8 (f) Ground truth

Figure 15: Visualization of the decoder. No ground truth patches/tokens are inserted in the visualization.

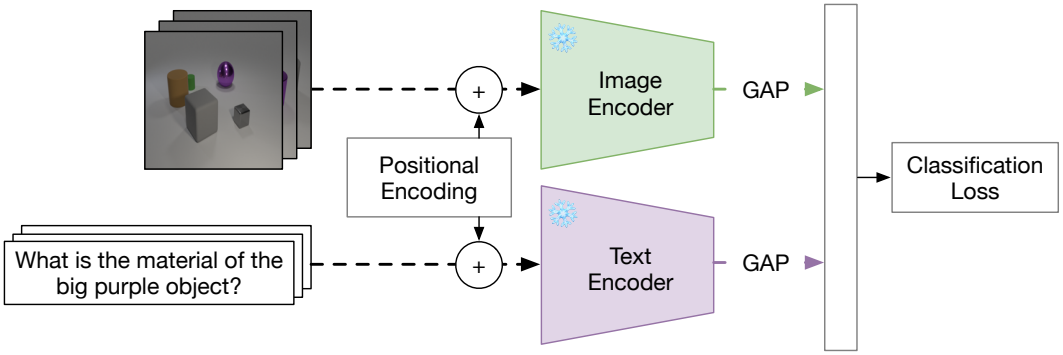


Figure 16: VQA Fine-tuning: linear probing. Encoders are frozen, only a linear head is trained.

While the highest performance is found when using the generative aspect of the MaskCLIP model, we also note that this is a more complicated fine-tuning setup that only performs slightly better than using a simple decoder with a classification task.

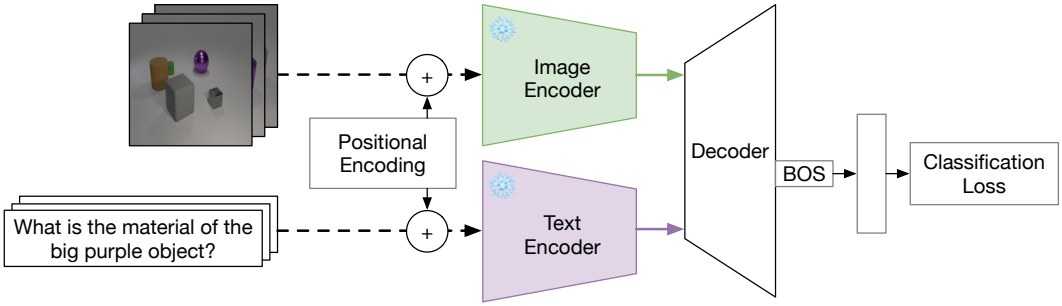


Figure 17: VQA Fine-tuning: decoder classification. Encoders are frozen, the BOS token of the decoder output is used as the classification token.

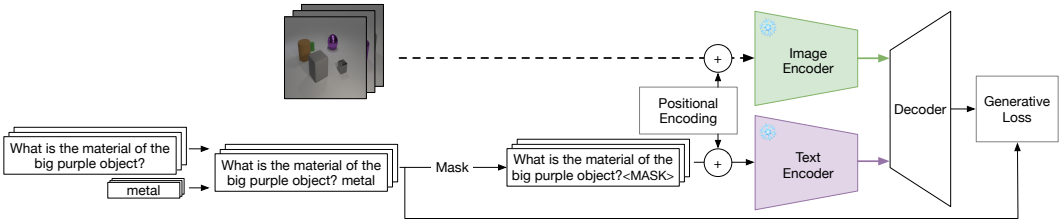


Figure 18: VQA Fine-tuning: generative decoder. Encoders are frozen, answer and EOS tokens are masked in the input. Decoder's masked token predictions are used as the answer, up to the EOS token.

Table 15: Parameters used when finetuning for VQA tasks.

Component	Parameter	Value
Image Encoder	Frozen	
Text Encoder	Frozen	
Decoder	Depth	8
	Width	512
	MLP Heads	8
Model	Weight decay	0.05
	LR Schedule	Stepwise scheduler
	Step gamma	0.1
	Layer Wise LR Decay	0.75
	Batch size	256
	Optimizer	AdamW
	Optimizer momentum	$\beta 1$, $\beta 2$ = 0.9, 0.999
	Augmentation	RandomResizedCrop
	Epochs	100