

Efficient Vision-Language Pre-training by Cluster Masking

Zihao Wei* Zixuan Pan* Andrew Owens
 University of Michigan

Abstract

We propose a simple strategy for masking image patches during visual-language contrastive learning that improves the quality of the learned representations and the training speed. During each iteration of training, we randomly mask clusters of visually similar image patches, as measured by their raw pixel intensities. This provides an extra learning signal, beyond the contrastive training itself, since it forces a model to predict words for masked visual structures solely from context. It also speeds up training by reducing the amount of data used in each image. We evaluate the effectiveness of our model by pre-training on a number of benchmarks, finding that it outperforms other masking strategies, such as FLIP, on the quality of the learned representation.

1. Introduction

Images contain a great deal of redundant information, making it challenging to efficiently learn representations from them at scale. Recent work has addressed this problem by masking image patches during vision-language contrastive learning [15, 33, 36, 70]. One simple approach is to drop a large fraction of the patches at random, making training more efficient by reducing the computational cost and memory usage in each training iteration [36]. An alternative strategy is to mask sets of semantically related patches [15, 33, 70], such as those that belong to the same object. This forces the learned model to predict words that describes missing scene structures from context, improving the learned representation. However, this approach requires a separate mechanism to group together semantically related patches, which adds considerable complexity to the learning procedure and is computationally expensive.

We propose a simple masking strategy for multimodal contrastive learning that avoids these shortcomings. During training, we mask random *clusters* of patches (Fig. 1). For this clustering, we use the patches’ raw RGB values as the feature representation. Our approach takes advantage of the fact that simple measures of visual similarity can often capture coherent visual structures, such as object parts [18, 53],



Figure 1. **Cluster masking.** We mask random clusters of visually similar image patches when training contrastive vision-language models (bottom). This masking strategy distinguishes our approach from methods that independently mask image patches for efficiency [36] (middle), while providing a similar improvement in training speed. It provides an extra learning signal, since it forces a model to predict words for missing scene structures solely from context.

especially when clusters are sampled randomly (Fig. 1). Our approach thus leads to more efficient training, like approaches that independently drop patches [36], while improving the learned representation via context prediction.

We take inspiration from masked region classification, a pre-training task widely used in vision-language models [9, 56, 57]. These models extract object features, then predict object labels for the randomly masked out regions. Our masking approach provides a similar training signal, since meaningful labels are included in the image caption. For example, as shown in Figure 1(a), the model is tasked with associating the words “fire hydrant” with an image even with the hydrant itself is mostly masked out.

*Equal contribution. Author order was determined by a coin flip.

We train our model on Conceptual 12M dataset [4] and evaluate our learned representation on a number of downstream tasks. These tasks include the zero-shot classification and linear probing on ImageNet [11], text and image retrieval on MS-COCO [38], and the SUGARCREPE language composition benchmark [25]. In our experiments, our model outperforms FLIP [36] and CLIP [49] on downstream performance, while having efficiency comparable with FLIP. We also show that the performance can further be improved by using the model’s learned feature embedding during clustering.

2. Related Work

Contrastive Vision-Language Pre-training. Vision-Language Pre-training (VLP) focuses on establishing connections between images or their components and human-interpretable language. This field initially evolved from transferring supervised learning models, which incorporated object detection modules to generate fine-grained visual labels [9, 56, 57]. Subsequently, there was a shift towards large-scale learning using noisy web data, moving away from reliance on fine-grained labels [1, 27, 34, 50, 69, 72, 73]. A significant development in this domain was CLIP [50], which applied contrastive learning techniques [6, 21] to train models to associate correct image-text pairs and dissociate incorrect ones. CLIP scaled contrastive visual-language models significantly beyond previous work, enabling strong feature learning and zero-shot performance. However, further scaling significantly increases the pre-training demands, requiring larger datasets and batch sizes.

In response to these challenges, recent research has explored incorporating masking into images to reduce training time and allow for more samples per batch [15, 20, 36, 70]. Methods such as MaskClip [15], FLIP [36], and VIOLET [20] have implemented random masking strategies. Yet, it has been noted that random masking may not be as effective on relatively small datasets [35, 70]. To address this, ACLIP [70] introduced a method of masking tokens with low cross-attention scores with text. However, this approach necessitates two forward passes to generate the attention map and requires additional computational modules [41]. In our work, we aim to avoid these limitations, proposing an effective masking method that is based on a patch’s raw RGB values.

Masked Image Modeling. In the field of language modeling, the effectiveness of models that learn to reconstruct corrupted inputs for generating robust features has been recognized [29, 39]. This approach, known as Mask Language Modeling (MLM), has been adapted in the realm of image processing as Mask Image Modeling (MIM). MIM techniques involve reconstructing either image patches or

their features [2, 5, 8, 14, 22, 63–65, 68]. The pioneering work in BEIT [2] introduced the reconstruction of discrete tokens, akin to VQ-VAE [59], using block-wise masking. This method demonstrated results on par with contrastive learning and self-distillation methods [3, 7] during model fine-tuning. Later approaches include PeCo’s [14] novel visual codebook learning method and BEIT V2’s [47] integration of self-distillation methods, using a teacher-student backbone and feature-level KL divergence loss [58]. Further exploration in this field has led to the use of natural image signals as reconstruction targets, moving away from learned features. Examples include SimMIM [68], which reconstructs pure RGB values, MaskFeat [64], introducing reconstruction of the Histogram of Oriented Gradients (HOG) features, and MAE [22], which reconstructs pixel-normalized RGB values. Our work draws inspiration from these studies, particularly in using pixel-normalized RGB values to compute patch similarities, arguing for a more effective distribution of patch features.

Masking Strategies in MIM. Parallel investigations have focused on masking strategies in MIMs [19, 28, 33, 37, 54, 66, 67]. Early works like BEIT and its successors used block masking, while others such as SimMIM, MaskFeat, and MAE applied random patch-wise masking. Attention-based masking strategies have also been explored, typically using attention maps from vision transformers. MST [37] masks less essential parts with low attention scores, using a reconstruction loss approach. In contrast, AttnMask [28] masks highly attentive patches and applies self-distillation loss. These methods involve simultaneous updates of attention maps and masks during training. A potential limitation of this approach is that insufficiently trained attention maps may not capture structured features effectively. SemMAE [33], starting with iBot features [75], adopts an easy-to-hard masking strategy, starting with masking parts within clusters and gradually expanding to entire clusters. Wilf et al. [66] introduce a unique entity-reinforced language model for masking objects in video frames. However, the reliance on pre-trained features or extracting attention maps can be computationally intensive. Evolved Part Masking [19] proposes using EM algorithm on attention maps to get a clustering before performing SemMAE style masking. Our approach also adopts a cluster based masking strategy in vision-language pre-training, enabling faster pre-training without requiring additional modifications to the model.

3. Method

We propose a cluster based masking strategy for contrastive vision-language pre-training, focusing on masking random clusters with visually similar semantics. Our method selects random anchor patches as cluster centers and computes pairwise patch distances to form clusters. These clus-

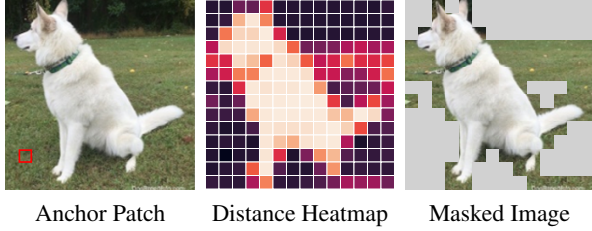


Figure 2. **Choosing clusters.** The process begins by randomly selecting anchor patches from the image. We then calculate the pairwise distances among all patches. Clusters formed within a distance threshold are masked out. We show cluster obtained from a single anchor patch.

ters are then masked entirely. To enhance accuracy in cluster formation, we introduce an adaptive layer for refining the distance matrix. Additionally, attention masks and a hard patch cutoff are used to ensure uniform input sizes are consistent in batches for auto differentiation.

3.1. Contrastive Vision-Language Pre-training

Our approach builds on contrastive vision-language pre-training methods, such as CLIP [49]. We use contrastive learning to align embeddings of matching text-image pairs and separate those of non-matching pairs. This process is steered by two symmetric InfoNCE losses [43]: the vision-to-language loss $\mathcal{L}_{v \rightarrow l}$ and its counterpart, the language-to-vision loss $\mathcal{L}_{l \rightarrow v}$. The vision-to-language loss is defined as:

$$\mathcal{L}_{v \rightarrow l} = -\log \frac{\exp(\text{sim}(I, T)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(I, T_j)/\tau)}, \quad (1)$$

where I and T are the embeddings for the image and text respectively, sim denotes the similarity function (we use a dot product), and τ is a temperature parameter. Similarly, $\mathcal{L}_{l \rightarrow v}$ is formulated by normalizing the loss using a batch of N image examples $\{I_j\}_{i=1}^N$ instead of text examples $\{T_j\}_{i=1}^N$.

3.2. Cluster Masking

We introduce a masking strategy that drops out random clusters. While one option would be to use an off-the-shelf clustering method, such as K-Means [40], we choose instead to use a simple and efficient method that results in a random clustering each training iteration (Figure 2). Our approach resembles a single iteration of K-Means, and works by selecting a set of exemplar patches, which each define a cluster. In our experiments, we also evaluate masking clusters obtained using K-Means as an alternative approach.

We split an input $H \times W$ image into patches, following [16]. We then compute the pairwise cosine similarity between every pair of normalized patches, which we use as a distance function $d(\mathbf{x}, \mathbf{y})$. We choose a small subset (less than 5%) of these patches at random to act as cluster centers. For each of these selected *anchor patches*, we define a cluster consisting of patches that lie within a distance

r . The cluster for an exemplar patch \mathbf{x} is represented by: $S_{\mathbf{x}} = \{\mathbf{y} \mid d(\mathbf{x}, \mathbf{y}) \leq r\}$ for image patches.

All patches within a cluster are masked out. The distance threshold r is automatically searched before training according to an average masking ratio. We provide a simplified pseudocode of the masking strategy in Algorithm 1.

Clustering Embedding Features. Another variant of patch feature is the combination of pure RGB values and patch embedding layer features from transformers [16]. When computing similarity scores, we integrate these two measures into a weighted sum, where the weight of each measure is determined by:

$$d(\mathbf{x}, \mathbf{y}) = \alpha \cdot d_{rgb}(\mathbf{x}, \mathbf{y}) + (1 - \alpha) \cdot d_{emb}(\mathbf{x}, \mathbf{y}), \quad (2)$$

where \mathbf{x} and \mathbf{y} represent two patches, d_{rgb} is the cosine similarity based on pure RGB values, and d_{emb} is the cosine similarity based on the transformer’s embedding features. The weight parameter α linearly increases from 0 to 1 during training.

The embedding layer is calculated before the patches enter the transformer, thus we could reuse the patch embeddings in the transformer without computing them twice. Using the embedding layer is advantageous because it incorporates positional encodings [60]. This integration potentially introduces spatial constraints, which we believe can further enhance our masking strategy.

Handling Batched Inputs. Deep learning libraries, like PyTorch [46], typically process batched inputs of uniform size. However, in our method, the mask ratio would vary across different images, leading to fluctuations in the number of patches. To accelerate the process, we introduce a *minimum mask ratio* threshold β for each image. If the calculated mask ratio for an image doesn’t meet this predefined threshold, we proceed to randomly drop patches until the desired ratio is achieved. Conversely, for images with patch counts less than the threshold, we use attention masks to avoid masked parts engaging in attention calculation [13].

4. Experiments

We present a comprehensive evaluation of our proposed algorithm to exhibit the performance, robustness, scalability, and efficiency of our framework.

4.1. Implementation Details

Datasets and Training Details. We train our model using the Conceptual 12M (CC12M) dataset[4], containing 12 million unique image-text pairs, for pre-training our vision-language models. We use ViT-B/16 as backbone for image encoder. The text encoder is a 12-layer transformer, equipped with 8 multi-head attention units and 512-dimensional embeddings. Input images are processed at a

Algorithm 1 Pseudocode of cluster based masking in a PyTorch-like style.

```
# img: the image for mask
# mask_ratio: the ratio for choosing anchor patches
# r: the threshold for computing cluster

def generate_mask(img, mask_ratio, r):
    # Make image into shape (N,L,C).
    # N: number of samples per batch
    # L: number of patches
    # C: size of feature dimension
    img = patchify(img)
    # Normalize each patch
    img = (img - img.mean(dim=-1)) / img.std(dim=-1)

    # Compute pairwise cosine similarity matrix
    x = img / img.norm(dim=-1)
    distance = bmm(x, x.transpose(-2, -1))

    # Generate a boolean masking matrix of shape (N, L)
    init_mask = random_patch_indicies(mask_ratio)
    init_mask = init_mask[:, :, None].float()
    # Get the cluster-based mask
    candidates = (distance * init_mask) >= r
    cluster_mask = candidates.sum(1) >= 1
    # Mask both the anchor patches and clusters
    mask = init_mask or cluster_mask

    return mask
```

bmm: batch matrix multiplication.

resolution of 224×224 , and text inputs are adjusted to 77 tokens, either by truncation or padding. A class token is transformed into a 512-dimensional feature embedding via a multi-layer perceptron (MLP). For optimization, we use the AdamW optimizer with a learning rate of 5×10^{-4} , $\beta_1 = 0.9$, and $\beta_2 = 0.98$. We use a batch size of 256 per GPU, and train using 8 NVIDIA A40 GPUs.

Our method comes in three variants: K-Means, RGB and Embedding. The RGB model clusters based on raw image patches, while the embedding model integrates patch embedding features with RGB for clustering. In the K-Means variant of the model, we mask out half of the clusters randomly. The model constructs 12 clusters and runs for a maximum of 10 iterations. For both RGB and embedding models, we set an average masking ratio of 50%, following the recommendations of FLIP [36] for optimal masking ratios. In the RGB approach, we use a 50% cutoff for Ours-RGB_{0.5} and 30% for Ours-RGB_{0.3}, whereas the Ours-Embedding model uses a 30% cutoff. Also, the RGB model selects anchor patches at a 3% ratio, compared to a 5% ratio in the embedding model.

Baselines. In our study, we establish baselines using three models: CLIP, FLIP, and FLIP_{Attn}, each trained from scratch on the CC12M dataset. These baseline models are derived from the open-source implementation of CLIP, known as OpenClip [10, 26, 48, 52]. For both FLIP and FLIP_{Attn}, we implement a patch dropout ratio of 50%. Specifically, FLIP uses a random dropout approach, whereas FLIP_{Attn} adopts an attention-based masking strategy inspired by ACLIP [70]. This strategy involves processing the image through the encoder and then averaging

across attention heads in the final transformer block to determine attention scores. Patches that receive the highest attention in relation to the [CLS] token are retained.

To ensure a fair comparison among these methods, we keep the number of patches consistent the same as ours within a single batch, which means for FLIP and FLIP_{Attn}, we apply a batch size of 256 for one GPU and for CLIP, we use 128 instead. Additionally, we apply a scaling law on learning rate across different models.

Evaluation Details. Our models are tested across various benchmarks to ensure its robustness and effectiveness. We conduct zero-shot image-to-text and text-to-image retrieval tasks on COCO [38] and Flickr [71], assessing its performance meticulously. Further more, we evaluate the models’ image representation quality by reporting both the zero-shot classification and linear probing performance on three mainstream datasets: ImageNet [11], CIFAR-10, and CIFAR-100 [30]. Zero-shot results of some other datasets like ImageNet variants [23, 24, 51, 55, 62], Caltech101 [17], Flowers [42] and Pets [61], are also reported to verify the method’s robustness.

For these tasks, our approach adheres strictly to the implementation used in the CLIP benchmark, ensuring consistency and reliability in the evaluation process. Furthermore, we assess the effectiveness of our methodology on language composition tasks using SUGARCREPE [25]. This evaluation aims to determine its adaptability and efficiency across various contexts, including object, attribute, and relation manipulations. Within the SUGARCREPE framework, models are tasked with identifying the correct caption that accurately describes an image, distinguishing it from a closely related but incorrect text hard negatives. The hard negatives is characterized by minor composition differences from the accurate caption.

4.2. Main Results

Visualization of Clusters. Figure 3 offers a visual depiction of our cluster based masking technique as outlined in the methodology section. For this illustration, we randomly select a number of image-text pairs from the COCO validation set and apply our masking method to the pure RGB data of the images. The visualization is showing the masking result of the two-stages. In the first stage, a subset of patches (5%) is randomly selected as anchor patches from the pool of all image patches, which are annotated with the red boxes. In the second stage, we visualize the masked clusters that are calculated based on the similarity matrix, where each cluster is represented by a distinct color.

Zero-shot Retrieval Results. In our investigation into the model’s understanding of the relationship between visual and linguistic representations, we conduct zero-shot

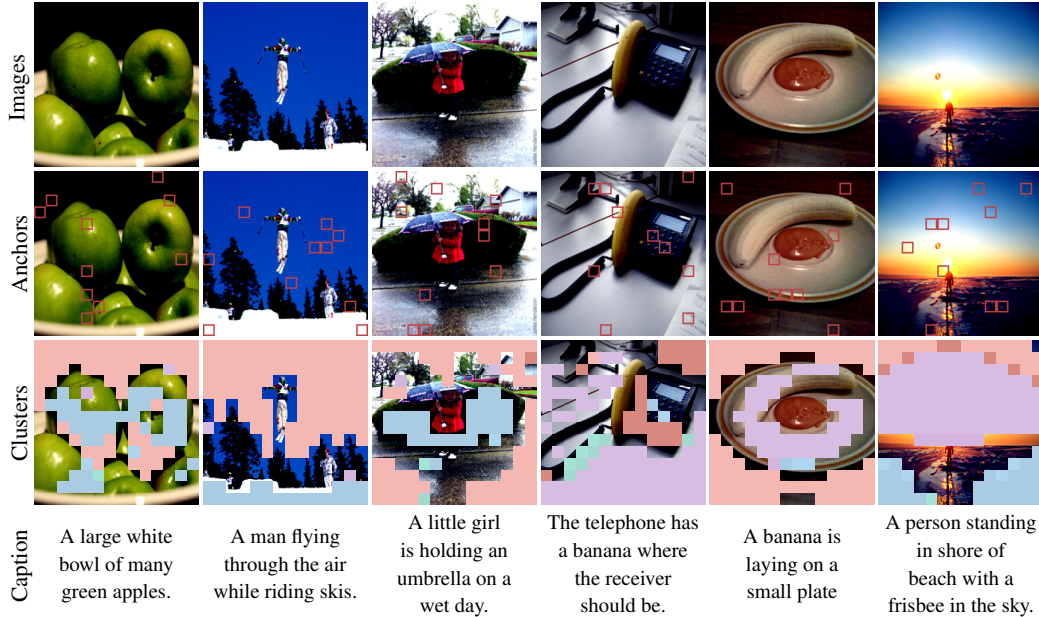


Figure 3. **Visualization of cluster masks.** Different colors represent distinct clusters formed by the similarity matrix calculated from the chosen anchor patches.

retrieval tests on several leading retrieval benchmarks. The results, detailed in Table 1, provide insights into the performance of our approach against others, particularly in the context of Image2Text and Text2Image’s recall precision at top1(R1), top5(R5) and top10(R10) metrics.

In the evaluation on the MS-COCO [38], Flickr8k, and Flickr30k [71] datasets, our model outperforms both the baselines in most parts. Notably, in the Image-to-Text tasks, our model performs best in most datasets, with the exception of a slight performance decrease compared to FLIP_{Attn} on the MS-COCO dataset. We attribute this success to our training strategy, which prioritizes primary clusters and minimizes the influence of noise. Furthermore, we observe that methods combining RGB information with token embeddings outperform those relying solely on RGB. We hypothesize that this is because the embedding layer, which contains slightly higher-level information.

When comparing FLIP to CLIP, FLIP’s performance is noticeably weaker, even with large batch sizes. We suspect that FLIP’s sub-optimal results in our experimental settings may not fully exploit its strengths. This aligns with findings from other studies, such as Yang et al.’s research on ACLIP [70], which also noted FLIP’s limitations. We observe that using attention scores for masking can improve performance compared to purely random masking. However, random masking still falls short of our cluster based masking or even the original CLIP method in some benchmarks.

Results on Zero-shot Classification and Linear Probing.

We evaluate our model on several widely recognized clas-

sification benchmarks. The zero-shot classification results are presented in Table 2, while the linear probing results can be found in Table 3. For better evaluating the time spent on training, we normalize all method’s training time by the CLIP’s training time, which is considered as $1\times$.

When comparing our model’s performance to CLIP (i.e., no masking), our model demonstrates superior results on the majority of test cases, showcasing an average improvement of $+2.1\%$, with about $+36\%$ speeding up. In comparison to the FLIP strategy, which has a similar training duration, our model has an improvement of $+5.5\%$. In comparison to the FLIP_{Attn}, our model does not need the attention map for guidance, which gives a much fast training speed, while having a performance of $+2.6\%$ on average.

Out of 12 datasets on the zero-shot classification benchmark, our RGB and embedding model achieves the top performance on 11 of them. In particular, it obtains strong performance on the ImageNet variants: ImageNet-A [24], ImageNet-O [55], ImageNet-R [55], and ImageNet-S [62], which often contain challenging and diverse images. The RGB version of our method also significantly outperforms FLIP and surpasses the CLIP model, especially on ImageNet and its variants, which demonstrates the effectiveness of our method with even the natural guidance.

The linear probing results further suggest the effectiveness of our method. Our models achieve $+1.8\%$ accuracy on ImageNet, $+3.1\%$ on CIFAR-10, and $+4.2\%$ on CIFAR-100.

Language Composition.

A potential drawback of our method might be understanding compositions of concepts

	text \rightarrow image									image \rightarrow text								
	MS-COCO			Flickr8k			Flickr30k			MS-COCO			Flickr8k			Flickr30k		
	R1	R5	R10	R1	R5	R10	R1	R5	R10	R1	R5	R10	R1	R5	R10	R1	R5	R10
CLIP [49]	34.60	61.98	72.72	55.70	81.60	89.90	58.50	83.80	89.10	23.49	47.80	59.66	40.54	68.90	80.20	43.18	70.44	80.40
FLIP [36]	32.62	59.14	70.64	55.00	80.90	88.90	53.80	80.80	88.50	22.56	46.08	58.09	40.32	68.10	78.64	41.52	67.90	77.46
FLIP _{Attn} [70]	33.66	60.18	71.02	53.70	80.09	87.99	55.29	81.40	87.60	23.89	48.33	60.04	40.58	68.88	78.86	43.06	70.10	78.82
Ours-Kmeans	33.68	61.14	71.97	55.10	83.30	90.90	55.40	82.00	88.90	22.60	46.93	58.84	40.10	69.86	79.91	41.32	68.50	77.58
Ours-RGB _{0.5}	32.82	60.20	71.40	52.10	81.20	89.50	54.90	81.20	88.00	22.97	47.29	59.21	40.84	68.72	79.14	41.80	70.20	79.42
Ours-RGB _{0.3}	35.87	61.50	72.34	54.70	81.40	90.70	57.60	83.90	90.30	23.65	47.54	59.25	40.90	68.22	79.00	42.92	69.96	79.12
Ours-Embedding	34.26	61.96	73.30	57.00	82.70	90.10	55.80	84.20	89.60	23.77	48.18	59.76	42.00	69.40	79.64	43.30	70.92	80.16

Table 1. **Zero-shot retrieval results.** We evaluate on MS-COCO [38], Flickr8k and Flickr30k datasets [71], where the Recall@1 (R1), Recall@5 (R5), and Recall@10 (R10) are reported.

Method	N.T.	IN-1K [11]	IN-A [24]	IN-O [55]	IN-R [23]	IN-S [62]	INv2 [51]	MNIST [12]	Cal101 [17]	CIFAR-10 [31]	CIFAR-100 [31]	Flowers [42]	Pets [61]	Average
CLIP [49]	1.00 \times	36.1	8.0	38.4	47.6	24.9	30.7	11.7	73.5	57.7	25.0	26.0	53.9	36.1
FLIP [36]	0.53 \times	34.4	7.1	39.5	41.4	20.1	29.5	10.4	70.4	52.8	24.5	25.3	46.0	33.5
FLIP _{Attn} [70]	1.06 \times	35.2	8.1	39.4	45.1	23.7	30.1	9.4	73.5	61.6	27.1	25.7	51.2	35.8
Ours-Kmeans	0.70 \times	35.5	7.2	38.3	43.0	22.2	30.1	9.7	69.9	61.4	25.3	25.6	52.1	35.0
Ours-RGB _{0.5}	0.54 \times	36.0	7.7	39.8	45.3	23.8	30.5	11.2	72.2	63.6	27.2	26.1	55.0	36.5
Ours-RGB _{0.3}	0.64 \times	36.6	8.8	39.9	45.9	24.9	31.8	9.4	72.3	63.1	26.3	25.4	57.3	36.8
Ours-Embedding	0.64 \times	36.3	8.1	39.6	47.9	25.4	30.7	11.0	73.7	70.7	32.0	28.4	55.4	38.2

Table 2. **Zero-shot classification result.** We evaluate popular datasets using clip-benchmark [32]. The training time is normalized according to the CLIP’s training time. Here N.T. represents normalized time against and IN represents ImageNet.

Method	CIFAR-10	CIFAR-100	IN-1K
CLIP	88.0	67.4	62.3
FLIP	85.9	65.5	61.3
FLIP _{Attn}	86.4	66.1	62.0
Ours-Kmeans	88.0	69.1	62.2
Ours-RGB _{0.5}	86.7	66.2	62.5
Ours-RGB _{0.3}	88.6	68.7	63.1
Ours-Embedding	89.0	69.7	62.7

Table 3. **Linear probing result.** All methods are trained for 10 epochs at learning rate of 1e-3. For CIFAR-10 and CIFAR-100, we use a batch size of 64 and for ImageNet-1k the batch size is 1024.

in language. As we mask out clusters, there is a risk that the model may increasingly adopt *bag-of-words* tendencies [74], which could impede its ability to learn the relationships between objects. For example, if an image is captioned with "dog on grass", the grass may be masked for a large portion in our model as they are highly similar to each other. This will make learning the relation "on" difficult. Therefore, we apply SUGARCREPE [25] benchmarks to test the model’s ability to understand language compositions. SUGARCREPE benchmarks assess this by generating negative captions through manipulations like adding, swapping, or replacing concepts in sentences, followed by text retrieval tests to evaluate the model’s accuracy in selecting the correct answer. From the our test results, which shown in Table 4, our model yields comparable results in Relation tests and demonstrates a significant enhancement in Object and Attribution tests, with an average improvement of **+3.9%** and **+3.0%** respectively, compared to FLIP. This improvement may stem from the masking of entire objects, which simplifies the challenge of contrastive learning by reducing ambiguity. This clarity facilitates the model’s learning of relationships, a crucial factor for composition

understanding.

Qualitative Comparison of Masking Strategies Our method outperforms the random masking strategy by preserving more semantic content in the unmasked image patches, a comparison showcased in Figure 1. The advantage of our technique is further explored by the captioning experiment detailed in Figure 4, wherein two sets of images, each masked differently, are fed into a captioner, GPT-4 [44, 45]. The captioner is prompted to generate MS-COCO-style captions for the unobscured sections. When comparing these captions to the standard references, it becomes clear that our cluster based masking not only retains key elements but also the interrelations among them. For instance, our approach accurately enables the captioning system to identify an airplane in the first example and to describe the baseball player’s action in the second, while the random masking strategy failed to achieve this clarity. These results indicate that our masking method provides a more detailed comprehension of the image.

4.3. Ablation Study

Ablation on Anchor Patch Ratio. In our study, we conducted an ablation on finding the optimal proportion of patches to serve as anchor patches. The results of this ablation are summarized in Figure 5. We use zero-shot learning results on the ImageNet-1k dataset as the benchmark for assessing the quality of the representations learned by our model. Additionally, we calibrate the threshold for each experiment to ensure that the average final masking ratio was maintained at 50%.

Our findings indicate that a smaller proportion of anchor patches tends to yield superior performance. We hypothesize that this improvement is due to the decreased ran-

	REPLACE			SWAP		ADD		Average		
	Object	Attribute	Relation	Object	Attribute	Object	Attribute	Object	Attribute	Relation
CLIP	85.77	79.18	64.51	61.78	58.71	74.24	68.35	73.71	68.75	64.51
FLIP	84.07	75.88	66.00	60.16	61.56	71.67	63.15	71.97	66.86	66.00
FLIP _{Attn-0.5}	86.62	75.50	63.22	52.44	63.06	71.65	66.76	71.57	68.39	63.22
FLIP _{Attn-0.3}	86.07	75.00	62.23	60.57	59.16	74.68	68.64	72.82	63.47	62.23
Ours-Kmeans	84.50	76.90	63.09	62.20	61.86	71.77	65.46	73.66	67.60	63.09
Ours-RGB _{0.5}	86.86	73.47	60.24	59.35	63.36	73.33	66.76	73.18	68.42	60.24
Ours-RGB _{0.3}	86.13	75.13	64.65	66.67	63.36	74.92	71.24	75.91	69.91	64.65

Table 4. **Language composition test result.** This table presents the performance of models on the SUGARCREPE evaluation, which involves replacing, swapping, or adding atomic concepts such as objects, attributes, and relations in a sentence to create mismatched captions.

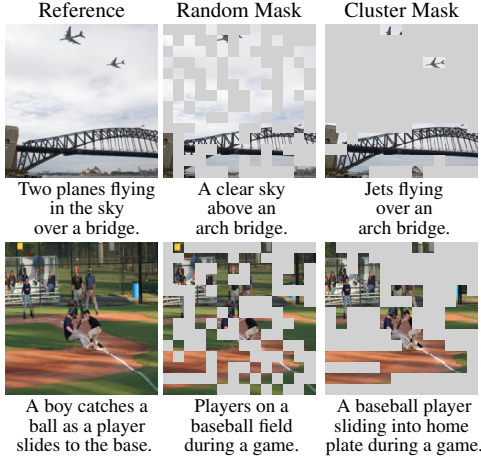


Figure 4. **Generated caption from visible patches.** We process the masked images through GPT-4 [44, 45] to create captions for the unmasked segments.

domness in the selection of anchor patches, which in turn enhances the clustering performance by providing a more stable set of reference points for the model to learn from. However, the masking ratio cannot be too small, as the similarity threshold will become too small, which may reduce the clustering quality.

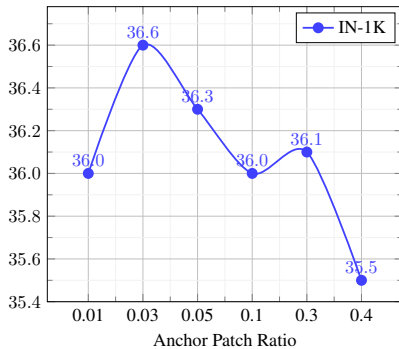


Figure 5. **Effect of anchor patch ratio.** All the final masking ratio is tuned to be 50%.

Ablation on Minimum Mask Ratio. We further demonstrate the capability of our method by setting the minimum mask ratio β the same as the average masking ratio of FLIP,

as shown in Table 1, 2, and Table 5. For our method, the cutoff ratio denotes the minimum mask ratio applied and the true visible patch ratio is shown in visible ratio. For the FLIP counterpart, it maintains a consistent mask ratio across all images. The results indicate that our method not only matches the speed of FLIP but also surpasses FLIP in zero-shot ImageNet-1K classification accuracy with a **+1.6%** improvement even by seeing fewer patches. The attention based masking method with less mask ratio achieves similar performance as ours but the speed is much slower.

These findings suggest that cluster based masking serves as an effective denoising technique for the dataset. A reason for this enhanced performance is that we could easily mask out typically irrelevant areas, such as uniformly colored backgrounds, which are less informative and oftentimes do not correspond to any word in the caption. This targeted approach enables the model to focus on more meaningful content within the images.

Additionally, our findings reveal an improved feature learning by the model when a smaller random masking is applied. By reducing the cutoff ratio from 50% to 30%, we observed a 1% enhancement in classification accuracy. Thus, there is some trade-off between the model’s performance and speed. Despite this, our model with larger masking cut off still remains significantly faster compared to attention-based masking or the original CLIP method.

Method	β	Visible Ratio	Normalized Time	IN-1K
FLIP	50%	50%	0.84×	34.4
FLIP	30%	70%	1.00×	35.4
FLIP _{attn}	50%	50%	1.73×	35.2
FLIP _{attn}	30%	70%	1.97×	36.6
Ours-RGB	50%	43%	0.84×	36.0
Ours-RGB	30%	50%	1.00×	36.6

Table 5. **Ablation on minimum mask ratio β .** Comparison of various methods against different minimum masking ratios. The zero-shot ImageNet-1k classification results are used as metric. The time is normalized to ours RGB model with $\beta=30\%$.

Ablation on Pixel Normalization. In our experiments, we incorporate pixel normalization (making each patch mean zero and unit standard deviation 1) into the process of computing the similarity matrix for images. This yields

a performance improvement of +1.1%, as shown in the results presented in Table 6a. The underlying rationale for this enhancement is attributed to the standardization of image patches. By using pixel normalization, we focus on the relative intensity of pixels, thereby diminishing the impact of lighting variations among different images.

This normalization process is particularly beneficial in scenarios where the dynamic range of pixel values varies significantly across different patches. By scaling the patches to a common range, pixel-norm mitigates the risk of disproportionate influence from patches with higher intensity values. Consequently, this leads to a more balanced and equitable comparison among patches, enhancing the model’s ability to discern and quantify similarities more effectively.

Method	IN-1K	k	IN-1K
w/o P.N.	35.5	0.5	36.1
w/ P.N.	36.6	1	36.3
		2	35.9

(a) Ablation on pixel normalization (P.N.).

(b) Ablation on polynomial coefficient k .

Table 6. **Ablation Study.** Table 6a presents an ablation study on the application of pixel normalization when calculating the similarity matrix for clustering. Table 6b explores the effects of varying the polynomial coefficient k , which adjusts the adaptive rate used when combining RGB and embedding features.

Effect of Features used in Clustering. In Tables 1 and 2, embedding-based methods surpass those dependent solely on RGB data, especially in image-to-text retrieval tasks. One reason for this may be the fact that the embedding model has access to the positional encoding, whereas the RGB-based model solely uses the appearance of each patch. Figure 6 qualitatively shows this advantage: while the RGB-only approach masks extra areas (such as hair or shadows in the first scenario; a laptop and phone in the second) due to color similarities, the embedding-based method masks more complete object parts.

Ablation on Adaptive Rate. In our approach, we interpolate between using RGB features and the patch embedding layer feature using a coefficient, α , which varies with each epoch. Denoting the current running epoch as E_c and the number of total training epochs as E_t , this coefficient is defined as $\alpha = \left(\frac{E_c}{E_t}\right)^k$. In addition to the linear method where $k = 1$, we explore other polynomial coefficients k for this combination, as summarized in Table 6b. Our findings suggest that the linear combination is most effective, likely due to its smooth transition.

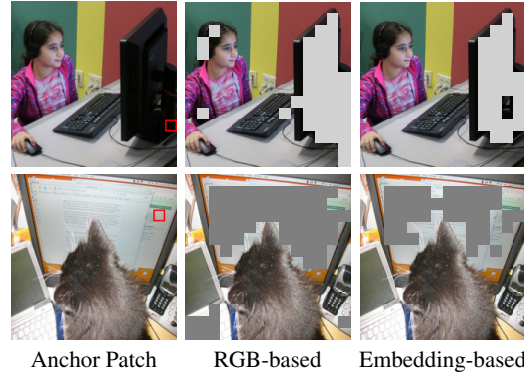


Figure 6. **Representation used in clustering.** We illustrate the distinctions between using pure RGB and ViT embeddings to compute the similarity matrix used in masking patches.

4.4. Limitations

Our methodology uses a uniform threshold for all images, a strategy that, while effective, may not be the most optimal. Future research could explore the implementation of individualized thresholds for each image, potentially leading to a more intelligent and adaptive masking process.

All of our approaches use the popular backbone architecture ViT-B/16 [16] and are trained solely on the CC12M dataset [4]. Expanding the scope of the experiments could offer additional insights.

5. Conclusion

In our study, we introduce a novel cluster based masking strategy designed for vision-language pre-training. Using either pure RGB values or shallow features from the patch embedding layer, our method effectively clusters image patches, maintaining essential visual semantics. We then randomly mask out these clusters, enabling efficient training. Our approach demonstrates success across various downstream evaluation tasks, including both pure image-based tasks such as image classification and multimodal tasks like image-text retrieval and language composition tests. We believe our work marks a considerable progression in this domain and anticipate that it will stimulate further research into optimizing masking strategies for similar applications.

Author Contributions All authors contributed to designing projects, launching experiments, and writing the paper. Zixuan Pan focused on algorithm optimization; Zihao Wei focused on code design; Andrew Owens supervised this project, offered feedback, and assisted in writing the paper.

Acknowledgements This research is supported by a Sony Research Award. We are grateful to Jeongsoo Park, Chao Feng, Yiming Dou, Daniel Geng, Ziyang Chen, and Ayush Shrivastava for their valuable suggestions and discussion.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. [2](#)
- [2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Bert: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. [2](#)
- [3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. [2](#)
- [4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3558–3568, 2021. [2](#), [3](#), [8](#)
- [5] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. [2](#)
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. [2](#)
- [7] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9620–9629, 2021. [2](#)
- [8] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. [2](#)
- [9] 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 *European conference on computer vision*, pages 104–120. Springer, 2020. [1](#), [2](#)
- [10] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. [4](#)
- [11] 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*, pages 248–255, 2009. [2](#), [4](#), [6](#)
- [12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. [6](#)
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018. [3](#)
- [14] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, Nenghai Yu, and Baining Guo. Peco: Perceptual codebook for bert pre-training of vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 552–560, 2023. [2](#)
- [15] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Maskclip: Masked self-distillation advances contrastive language-image pretraining, 2023. [1](#), [2](#)
- [16] 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. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. [3](#), [8](#)
- [17] Li Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006. [4](#), [6](#)
- [18] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59:167–181, 2004. [1](#)
- [19] Zhanzhou Feng and Shiliang Zhang. Evolved part masking for self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10386–10395, 2023. [2](#)
- [20] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021. [2](#)
- [21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. [2](#)
- [22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. [2](#)
- [23] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. [4](#), [6](#)
- [24] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. [4](#), [5](#), [6](#)
- [25] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *arXiv preprint arXiv:2306.14610*, 2023. [2](#), [4](#), [6](#)
- [26] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave,

- Vaishal Shankar, Hongseok Namkoong, John Miller, Hananeh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. If you use this software, please cite it as below. 4
- [27] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 2
- [28] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzas, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *Computer Vision – ECCV 2022*, pages 300–318. Springer Nature Switzerland, 2022. 2
- [29] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, page 2, 2019. 2
- [30] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 4
- [31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6
- [32] LAION-AI. Clip benchmark. https://github.com/LAION-AI/CLIP_benchmark, 2023. 6
- [33] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *Advances in Neural Information Processing Systems*, 35:14290–14302, 2022. 1, 2
- [34] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 2
- [35] Xianhang Li, Zeyu Wang, and Cihang Xie. Clipa-v2: Scaling clip training with 81.1 2
- [36] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking, 2023. 1, 2, 4, 6
- [37] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34:13165–13176, 2021. 2
- [38] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. Microsoft coco: Common objects in context. In *ECCV. European Conference on Computer Vision*, 2014. 2, 4, 5, 6
- [39] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 2
- [40] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 3
- [41] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training, 2021. 2
- [42] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, page 722–729, USA, 2008. IEEE Computer Society. 4, 6
- [43] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3
- [44] OpenAI. Gpt-4 technical report, 2023. 6, 7
- [45] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, pages 27730–27744. Curran Associates, Inc., 2022. 6, 7
- [46] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 3
- [47] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 2
- [48] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 4
- [49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 6
- [50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2
- [51] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishal Shankar. Do imagenet classifiers generalize to imagenet?, 2019. 4, 6
- [52] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 4

- [53] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. 1
- [54] Yuge Shi, N Siddharth, Philip Torr, and Adam R Kosiorek. Adversarial masking for self-supervised learning. In *International Conference on Machine Learning*, pages 20026–20040. PMLR, 2022. 2
- [55] Anugya Srivastava, Shriya Jain, and Mugdha Thigle. Out of distribution detection on imagenet-o, 2022. 4, 5, 6
- [56] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vi-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 1, 2
- [57] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 1, 2
- [58] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021. 2
- [59] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2
- [60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. 3
- [61] Andrea Vedaldi. Cats and dogs. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 3498–3505, USA, 2012. IEEE Computer Society. 4, 6
- [62] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019. 4, 5, 6
- [63] Yiqing Wang, Zihan Li, Jieru Mei, Zihao Wei, Li Liu, Chen Wang, Shengtian Sang, Alan Yuille, Cihang Xie, and Yuyin Zhou. Swinmm: Masked multi-view with swin transformers for 3d medical image segmentation. In *MICCAI*, 2023. 2
- [64] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022. 2
- [65] Zihao Wei, Chen Wei, Jieru Mei, Zeyu Wang, Xianhang Li, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. MAE are secretly efficient learners, 2023. 2
- [66] Alex Wilf, Syeda Nahida Akter, Leena Mathur, Paul Pu Liang, Sheryl Mathew, Mengrou Shou, Eric Nyberg, and Louis-Philippe Morency. Difference-masking: Choosing what to mask in continued pretraining. *arXiv preprint arXiv:2305.14577*, 2023. 2
- [67] Jiahao Xie, Wei Li, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Masked frequency modeling for self-supervised visual pre-training. *arXiv preprint arXiv:2206.07706*, 2022. 2
- [68] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 2
- [69] Fengyu Yang, Chao Feng, Ziyang Chen, Hyungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. *arXiv preprint arXiv:2401.18084*, 2024. 2
- [70] Yifan Yang, Weiquan Huang, Yixuan Wei, Houwen Peng, Xinyang Jiang, Huiqiang Jiang, Fangyun Wei, Yin Wang, Han Hu, Lili Qiu, and Yuqing Yang. Attentive mask clip, 2022. 1, 2, 4, 5, 6
- [71] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 4, 5, 6
- [72] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2
- [73] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2
- [74] Mert Yuksekogunul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2022. 6
- [75] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 2