FREQUENCY IS WHAT YOU NEED: WORD-FREQUENCY MASKING BENEFITS VISION-LANGUAGE MODEL PRE-TRAINING

Mingliang Liang, Martha Larson
Radboud University
Nijmegen
{m.liang, m.larson}@cs.ru.nl

ABSTRACT

Vision Language Models (VLMs) can be trained more efficiently if training sets can be reduced in size. Recent work has shown the benefits of masking text during VLM training using a variety of approaches: truncation, random masking, block masking and syntax masking. In this paper, we show that the best masking strategy changes over training epochs and that, given sufficient training epochs, word frequency information is what you need to achieve the best performance. Experiments on a large range of data sets demonstrate the advantages of our approach, called Contrastive Language-Image Pre-training with word Frequency Masking (CLIPF). The benefits are particularly evident as the number of input tokens decreases. We analyze the impact of CLIPF vs. other masking approaches on word frequency balance and discuss the apparently critical contribution of CLIPF in maintaining word frequency balance across POS categories.

***K*eywords** Vision-Language Model · Multimodal Data · Word Frequency Masking.

1 Introduction

Vision-Language Models (VLMs) learn embeddings that represent both visual and textual modalities simultaneously, and are known for the ease with which their strong performance transfers from one domain to another [1, 2, 3, 4, 5, 6, 7, 8]. CLIP (Contrastive Language-Image Pre-training) [3, 9, 10] is a VLM that is trained on a very large number of image-text pairs collected online, where the text is considered a "natural language" annotation for the image.

This paper contributes to the recent trend of text-masking approaches that improve VLMs by removing words from the text data during training, speeding up training in the process [11, 12]. We introduce CLIPF (Contrastive Language-Image Pre-training with word Frequency Masking), which selects words to be masked on the basis of word frequency. We carry out a comparison with CLIPA masking [12], which selects words on the basis of four strategies that do not use frequency information: truncation, random word selection, random block selection, and part-of-speech (POS) information and is currently considered the state-of-the-art text-masking.

The comparison of the learning curves of our CLIPF with CLIPA masking in Fig. 1 illustrates two key contributions of this paper. First, we demonstrate that the number of training epochs is crucial to which masking strategy yields the highest performance. The finding is an important improvement on the state of the art of text masking for CLIP training, represented by [12]. In [12], CLIP is trained on a limited number of epochs and syntax masking is found to outperform truncation, random, and block masking. In contrast, we show that random and block masking outperform syntax masking after a sufficient number of epochs. Second, we show that when masking text for CLIP training, frequency is what you need. As evident in Fig. 1, our frequency-based CLIPF approach outperforms all CLIPA masking approaches with sufficient epochs of training, leading it to be the best performing text masking approach.

The history of text masking in language model training dates to [13], where subsampling of frequent words in the training text was used to improve the performance of skip-gram language models. The study of text masking for VLMs got off to a slow start, since initially [8] reported that text masking, although it promotes efficiency, deteriorated performance. Subsequently, [12] expanded the types of masking studied and demonstrated that masking can indeed improve performance. Simultaneously, frequency-masking was proposed at a workshop [11], but fell short of the

Table 1: **Illustration of different text masking methods and words masking probabilities.** Bold percentages or check marks provide an example of the words that would be retained by each text masking strategy. For the Truncation and Syntax methods, four words per text are retained as the input. For the Random, Block, and CLIPF methods, we present the word masking probabilities. The second row shows the part-of-speech (POS) category of each word.

Models	Methods	Words											
		a	**small**	**dog**	**sleeping**	**on**	**a**	**couch**	**next**	**to**	**a**	**remote**	**.**
	POS	Other	Adjective	Noun	Verb	Other	Other	Noun	Other	Other	Other	Noun	Other
CLIPA	**Truncation**	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
	Random	**50%**	50%	50%	**50%**	50%	50%	50%	**50%**	50%	50%	50%	**50%**
	Block	50%	50%	**50%**	**50%**	**50%**	**50%**	50%	50%	50%	50%	50%	50%
	Syntax	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗	✓	✗
CLIPF	**Frequency**	99.37%	96.28%	**96.13%**	**88.35%**	98.96%	99.37%	**87.03%**	93.69%	98.89%	99.37%	**81.11%**	99.37%

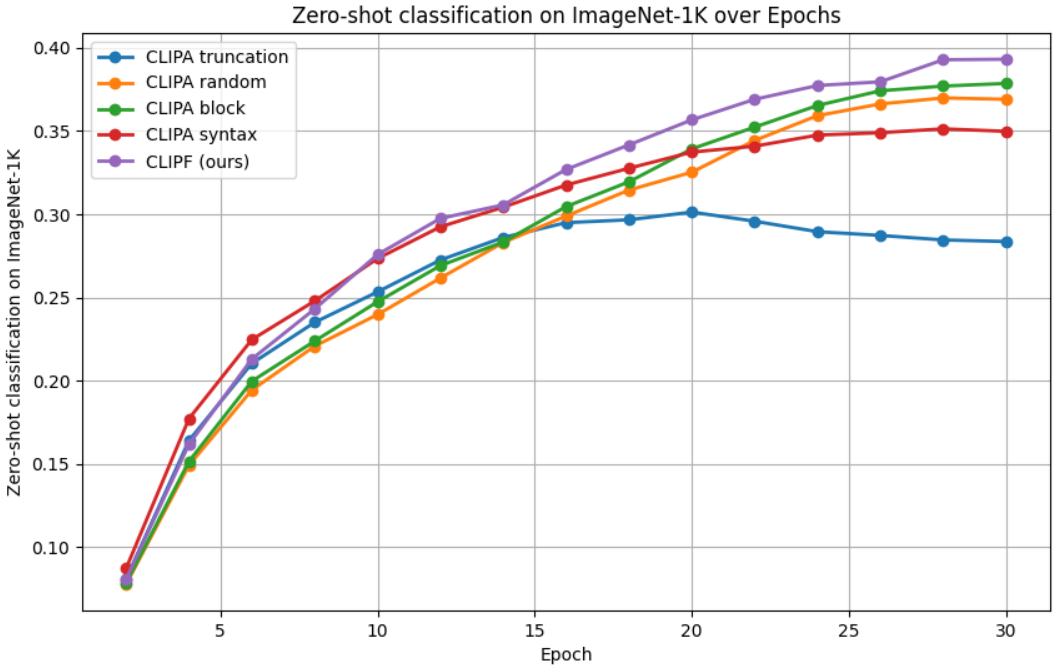

Figure 1: Zero-shot classification accuracy on ImageNet-1K over training epochs for CLIPF and CLIPA strategies. The text masking ratio is 75%. The backbone of the image encoder is ViT-B/16. The model is pre-trained on CC12M for 30 epochs. We use image masking (75%) to speed up training. All approaches are subsequently fine-tuned with one epoch on the full, unmasked data.

potential of frequency realized by CLIPF due to the use of a cut-off threshold. To our knowledge, our work reported in this paper is the first to attempt to discover the reasons that certain forms of text masking outperform others.

An initial illustration of the differences between task-masking strategies is provided in Table 1. Truncation consistently retains the beginning of the text. With the random (random word selection) and block (random block selection) methods, each word has an equal probability of being masked, with block masking maintaining word neighborhoods. Syntax masking prioritizes the retention of nouns, followed by other part-of-speech (POS) categories. Masking based on frequency (CLIPF) has the potential to address the shortcomings of the other approaches. Unlike truncation, it does not prioritize words that tend to occur at the beginning of the sentence. In contrast to random and block masking, it represses words with the POS category "Other", which are often stop words and for this reason high frequency. However, it does not give an overly high priority to nouns and adjectives, as with syntax masking.

In sum, the contributions of this paper are as follows:

- We point out, to our knowledge for the first time, the importance of the number of training epochs for comparing text masking approaches.

- We introduce CLIPF, a frequency-based masking approach that outperforms previously proposed approaches.
- We carry out an analysis that provides insight into why CLIPF outperforms other approaches.

Our work is reproducible because it uses training data sets that are openly available and small enough to allow researchers with limited computational resources to repeat our experiments and because we release our code[1]

2 Related work

2.1 Vision-Language Models

Vision-language models (VLMs), such as CLIP, ALIGN, LiT, and FLIP [3, 4, 10, 8], learn image embeddings from large-scale natural language supervision using contrastive learning [14] to grants the models remarkable transferability. Because CLIP does not release its pre-training dataset, OpenCLIP pre-trained models on several large-scale public datasets [15, 16, 17, 18] to reproduce CLIP. This implementation adheres to the scaling law observed in Neural Language Models [19] and Vision-Language Models [3], where loss decreases according to a power-law relationship with model size, dataset size, and computational resources utilized during training. While large language models (LLMs) have advanced rapidly, we focus on vision-language models (VLMs) like CLIP, as they are essential tools for bridging visual and textual modalities in current research. Although CLIP and OpenCLIP are pre-trained on very large-scale datasets, recent works like DeCLIP [20] and SLIP [5] demonstrate that Vision-language Models (VLMs) can be trained efficiently with smaller datasets, enabling researchers with limited computational resources to contribute. Moreover, masking image patches or reducing image resolution during VLM pre-training also substantially decreases computational costs while maintaining or enhancing performance, as shown by techniques like FLIP, RECLIP, CLIPA [8, 21, 12].

2.2 Text masking for VLM training

Masking text presents another viable option to speed up the pre-training of VLMs, and is particularly effective alongside strategies that employ a high ratio of image patch masking. FLIP further explored text masking techniques, namely, random and truncation masking, to enhance efficiency [8]. While random text masking involves masking text arbitrarily, truncation masking in FLIP strategically eliminates padding tokens and retrains the words that preceded them.

Subsequently, CLIPA [12] introduces block masking and syntax masking. As already mentioned in Section 1, CLIPA [12] shows text masking during training can improve the performance of a VLM. CLIPA syntax masking prioritizes the retention of nouns, followed by adjectives and then other words. In [12], syntax masking outperforms the truncation, random, and block approaches. However, [12] does not vary the number of training epochs. An important contrast with our work, as previously mentioned, is our attention to the number of training epochs. Further, CLIPF, as we will see, counters the danger of giving overly high priority to nouns, leading to overfitting, and also avoids overly suppressing words of the POS category "Other".

An initial effort to exploit frequency information for text masking during VLM training was made by SW-CLIP [11]. Unfortunately, SW-CLIPF introduces a frequency cutoff threshold, which reduces performance, as we demonstrate later in this paper. In order to exploit frequency information effectively each training text needs to fill as many available input tokens as possible, as is done in CLIPF.

3 Method

In this section, we describe various text masking strategies, including our CLIPF method. All these strategies accelerate pre-training by masking specific words in the text.

3.1 Text Masking

Consider a text T_i consisting of words $\{w_1, w_2, w_3, \ldots, w_n\}$, where n represents the total number of words in T_i. Text masking involves selecting a subset of these words to create a new version of the text, denoted as

$$T_{new\_i} = f(T_i\{w_1, w_2, w_3, \ldots, w_n\}). \tag{1}$$

This function f determines which words are masked to get the modified text T_{new_i}. And, we define the number of input tokens as k. We will sequentially introduce our masking strategies: random, truncation, block, syntax, and frequency. It is important to note that the number of padding tokens remains consistent across all these strategies.

[1] https://github.com/MingliangLiang3/CLIPF

Table 2: Dataset Statistics for Pre-training datasets. Caption length refers to the number of words in the text.

Dataset	Samples	Total words	Caption length
CC3M	2721877	28022082	10.30 ± 4.72
CC12M	9295444	205716854	22.15 ± 17.20

Random Masking: In the random text masking strategy, the function f operates to randomly select words from the text T_i. Each word w_i has an equal probability of being masked, independently of its position or frequency in the text.

Truncation Masking: In truncation masking, the function f reduces the text by masking words from the end part of the text. The number of text tokens retained depends on the input length of the text encoder.

Block Masking: Block masking is similar to truncation masking but starts at a random index within the text. If the text tokens are longer than the input tokens, we choose a random starting index j that is less than the length of the text tokens minus the length of the input tokens k.

Syntax Masking: To implement syntax masking [12], each word w_i in the text T_i is first tagged using a natural language processing tool like NLTK [22] to identify its part of speech. The words are then sorted according to their syntax role. This ordering strategy $O(w_i)$ reflects the importance of nouns, adjectives, and verbs in the text. Words are masked based on their assigned order, with priority given to retaining nouns, followed by adjectives, verbs, and then other parts of speech. This method ensures that the core semantic information of the text is less likely to be masked, especially when learning the relationship between the image and the text, as nouns are more relevant to the object of the image.

3.2 Word Frequency Masking

As in previous work, subsampling of frequent words was introduced by [13] in the natural language processing (NLP) domain to accelerate the learning process for word representations. The goals are to accelerate training and to address the imbalance between frequent and infrequent words in the training corpus. In our work, we use the formula (Formula 2) introduced in [13] to determine the probability of masking a word during training. Thus, the complete formula to calculate the probability $P(w_i)$ that accounts for all scenarios is:

$$P(w_i) = \begin{cases} 1 & \text{if} N < 5 \\ 0 & \text{if} f(w_i) < t \\ 1 - \sqrt{\frac{t}{f(w_i)}} & \text{otherwise} \end{cases} \tag{2}$$

Here, the masking probability of the word w_i is $P(w_i)$, t is the threshold, and $f(w_i)$ is the frequency of the word. Additionally, if the total number of words (N) is fewer than 5, deemed too infrequent for effective learning, the probability that the word will be masked is 1.

Following FLIP [8], we fine-tune the model with an additional epoch to eliminate the distribution gap between the masking and unmasking.

4 Experimental Setup

Our implementation follows CLIP [3], OpenCLIP [9], FLIP [8] and CLIPA [12]. In this section, we present the details of the training and testing.

4.1 Implementation

Dataset We pre-train the models on Conceptual Captions 3M (CC3M) [23] and Conceptual 12M (CC12M) [15]. These datasets were selected due to their diverse range of real-world concepts and scenes. Popular VLMs such as OpenCLIP, DeCLIP, and SLIP [9, 20, 5] also employ these datasets for training, leveraging their extensive and diverse content. Due to the expiration and inaccessibility of some URLs in the training set, we have downloaded 2.7 million image-text pairs for CC3M and 9.3 million image-text pairs from CC12M. The characteristics of the dataset are presented in Table 2. The CC12M dataset is 3.42 times larger than the CC3M dataset, with captions that are on average 2.15 times longer. Since the number of input tokens is fixed, the impact of different text masking ratios varies across datasets due to differences related to caption length.

Table 3: Details of the pre-training and fine-tuning setups.

Config	**Pre-training**	**Fine-tuning**
optimizer	AdamW [27]	AdamW [27]
learning rate	1e-3	1e-5
weight decay	0.2	0.2
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.98$ [28]	$\beta_1, \beta_2 = 0.9, 0.98$ [28]
learning rate schedule	cosine decay [29]	cosine decay [29]
warmup steps	10k	10%
epoch	30	1
t	10^{-6}	—
τ	0.07	—
numerical precision	amp	amp
augmentation	RandomResizedCrop	RandomResizedCrop

Table 4: During pre-training, each model processes a certain number of image and text tokens. The first and second columns of the table display the image and text masking ratios, respectively. The third and fourth columns show the number of image and text tokens processed after masking. The fifth column presents the total number of tokens (image + text) after masking. The last column provides the percentage of total tokens relative to the unmasked case.

Image masking	Text masking	Image tokens	Text tokens	Total	Percentage
0%	0.00%	196	32	228	100%
75%	0.00%	49	32	81	35.53%
	50.00%	49	16	65	28.51%
	75.00%	49	8	57	25.00%
	81.25%	49	6	55	24.12%
	87.50%	49	4	53	23.25%

Architecture For the image encoder, we use ViT-B/16 [24] architectures with global average pooling and learnable positional embeddings. The input image size is 224. For the text encoder, we use a Transformer-based model [25] and byte-pair encoding with a 49K token vocabulary. We chose these encoders because they are the largest encoders that we can use given our resources. When using full text for training, the maximum context length is 32. Zero-padding is applied to input text that is shorter than the maximum token length of the model.

We trained the models using 8 RTX A5000 GPUs with the same settings to ensure consistent conditions across all models. This allows us to effectively measure and compare the performance of each model in processing text tokens during training. We experiment with different text token input sizes, namely, 32, 16, 8, 6, and 4 text tokens. As the input size gets smaller, we can increase the batch size, maximizing computational memory usage. We used batch sizes of 664, 832, 896, 928, and 944, respectively. Note that CLIP (and therefore CLIPA and CLIPF) is trained using contrastive learning, which benefits from larger batch sizes. Note also that although we used different text masking with the same settings and batch sizes, syntax masking was slower than the other text masking strategies when conducting the experiments because POS processing for each word is time-consuming.

We pre-train the model using the InfoNCE loss [14] with a learnable temperature parameter τ [26, 3]. To classify images, we calculate the cosine similarity between the image and text embeddings.

Training We follow the setup of OpenCLIP [9], FLIP [8] and CLIPA to pre-train our model. Following OpenCLIP [9], we pre-train our model for 30 epochs on the CC3M and CC12M datasets. Details of the pre-training configuration are given in Table 3.

Similar to FLIP, to speed up training, we apply 75% image masking to the image encoder as the baseline model. As a result, the speedup is about 4$\times$ compared to training without image masking, while the reduction in the zero-shot performance of ImageNet-1K classification remains within reasonable bounds.

During text masking, we reduce the number of tokens from 32 to 16, 8, 6, and 4. To measure the training speed of CLIPF, CLIP, and CLIPA, we compare the number of text tokens processed by each model. As shown in Table 4, the number of text tokens processed by each model during pre-training is calculated based on different image and text masking ratios. When we pre-trained the model using image masking, the total number of tokens is 81 and the percentage of text tokens is 39.5%. When we apply 50% text masking, the total number of tokens is 65. Compared to training without text masking, this results in a speed increase of approximately 20%. Moreover, when we apply 87.5%

Table 5: **Comparison of CLIPF, CLIP, and CLIPA for zero-shot classification on ImageNet-1K.** reported as top-1 accuracy. All models use a ViT-B/16 image encoder backbone pre-trained on **CC3M** and **CC12M** for 30 epochs with 75% image masking for speedup (pre-train) and fine tuned for an additional epoch on all data (fine-tune). "Image tokens" and "text tokens" reflect the amount of image and text processing during pre-training.

Models	Masking	Image Tokens	Text Tokens	CC3M		CC12M	
				pre-train	fine-tune	pre-train	fine-tune
CLIP	✗	197	32	18.6	✗	36.6	✗
FLIP	✗	49	32	14.1	14.2	32.0	33.7
CLIPA	truncation	49	16	13.8	13.8	32.8	32.8
	random			13.9	13.9	33.7	34.3
	block			13.9	13.9	34.3	34.8
	syntax			13.3	12.8	32.2	34.4
CLIPF	frequency			**14.0**	**14.0**	**35.5**	**36.0**
CLIPA	truncation	49	8	10.8	12.0	25.4	28.4
	random			**17.6**	17.4	34.5	36.9
	block			16.2	16.6	35.5	37.9
	syntax			17.2	**17.5**	28.5	35.0
CLIPF	frequency			16.8	17.0	**36.6**	**39.3**
CLIPA	truncation	49	6	8.4	9.4	15.3	23.2
	random			12.8	17.9	26.9	34.6
	block			12.9	17.0	28.6	35.9
	syntax			12.2	15.7	25.2	32.6
CLIPF	frequency			**14.4**	**18.2**	**30.3**	**37.8**
CLIPA	truncation	49	4	3.8	8.2	5.3	19.8
	random			5.4	14.6	14.0	27.1
	block			7.5	14.5	**18.7**	26.6
	syntax			8.9	13.0	14.2	24.6
CLIPF	frequency			**10.9**	**16.0**	17.0	**30.9**

text masking, the total number of tokens is 53, resulting in a speed increase of approximately 34% compared to training without text masking.

Fine-tuning Following FLIP, and CLIPA, in order to bridge the gap between pre-training and evaluation, we fine-tuned the model without images and text masking. The details of the fine-tuning configuration are provided in Table 3.

4.2 Evaluation

Evaluation setup Following the CLIP [3], FLIP [8], and CLIPA [12] methods, several classification benchmarks were used to evaluate the zero-shot capability of CLIPF. Among these benchmarks is ImageNet-1K, a widely recognized dataset in computer vision. It is frequently used for image classification and VLMs tasks and comprises 50K validation samples across 1K different classes. We filled each class into the templates provided by CLIP [3] to calculate the average of the text embeddings. We use the same evaluation settings as CLIP [3] to evaluate the other downstream datasets.

5 Experimental Results

5.1 Zero-short performance on ImageNet

Table 5 presents results for zero-shot classification on ImageNet-1K for models trained on both the CC3M and CC12M data. The experiments compare different masking approaches and different masking ratios, indicated in the table with the number of input text tokens (Text Tokens column). Models are pre-trained for 30 epochs (pre-train column) and then fine tuned for one epoch (fine-tune column) with the full, unmasked data. Note that since the average text length in CC3M is less than 16 tokens, the benefits of fine-tuning a model pre-trained with 16 text tokens are limited.

We see that CLIPF outperforms the other CLIPA matching approaches in nearly all cases across text masking ratios (input sizes) as reflected by the bolded accuracy scores. In general, random, syntax, and block strategies exhibit similar performance; however, truncation performs worse. Contrary to the conclusion of the original CLIPA work [12], syntax masking does not yield the best performance. As mentioned in Section 1, we observe that the relative performance

Table 6: **Zero-shot robustness evaluation.** Comparison of the zero-shot accuracy performance of CLIP, FLIP, CLIPA, and CLIPF on various datasets. The models are pre-trained on **CC12M** [15] for 30 epochs with image masking (75%) to speed up training and fine-tune the model an additional epoch without image and text masking.

Models	Masking	Image Tokens	Text Tokens	ViT-B/16					
				IN-A	IN-O	IN-R	IN-S	IN-V2	ON
CLIP	✗	197	32	8.97	37.85	49.11	25.70	31.48	24.20
FLIP	✗	49	32	6.90	40.30	36.90	17.60	28.70	18.10
	truncation			7.10	37.70	36.40	17.50	27.40	17.20
	random			8.05	38.60	40.00	20.34	29.61	20.84
CLIPA	block	49	16	8.05	39.60	39.17	19.31	29.70	19.72
	syntax			8.00	**40.70**	39.41	18.61	29.17	19.96
CLIPF	frequency			**8.28**	39.10	**40.86**	**20.48**	**30.78**	**21.15**
	truncation			6.71	34.05	37.18	15.99	24.49	15.98
	random			9.16	38.15	43.35	22.53	31.24	**22.17**
CLIPA	block	49	8	9.05	**39.95**	43.32	22.58	32.42	21.58
	syntax			8.50	38.40	39.90	19.50	29.80	19.50
CLIPF	frequency			**9.56**	39.90	**45.76**	**23.62**	**34.16**	21.59
	truncation			6.70	26.60	33.20	13.30	20.20	14.80
	random			8.10	36.10	41.80	21.10	30.20	19.80
CLIPA	block	49	6	8.32	38.20	42.99	21.73	30.82	19.76
	syntax			7.40	34.10	39.60	18.50	28.60	19.80
CLIPF	frequency			**8.69**	**38.55**	**45.09**	**23.45**	**32.59**	**20.04**
	truncation			5.30	21.70	25.40	9.00	17.10	11.20
	random			6.10	30.30	33.80	14.90	23.50	15.30
CLIPA	block	49	4	6.08	30.55	34.69	15.54	22.99	15.10
	syntax			6.50	29.10	32.40	13.30	22.00	16.00
CLIPF	frequency			**7.05**	**32.40**	**39.90**	**18.94**	**26.72**	**17.14**

of different text masking approaches depends on the number of training epochs. In [12], CLIPA only pre-trained the models for 6.4 epochs. We discuss the learning curve in Fig.1 further in Section 6.2.

It is particularly interesting to observe from the results in Table 5 the fact that models trained with text masking (CLIPA and CLIPF) are competitive with, or superior to, models trained on the original, unmasked text data (CLIP, FLIP). Note that at a text masking ratio of 75% (corresponding to 8 tokens), the masking approaches are 1.3 $\times$ faster than FLIP and 4.0 $\times$ faster than CLIP.

We first consider the comparison with CLIP. On CC3M, CLIPF approaches the performance of the original CLIP approach with a 81% text masking ratio (4 input tokens). On CC12M, CLIPF outperforms the original CLIP (no masking) with a 75% masking ratio (8 input tokens). Fine-tuned CLIPF with a 75% masking ratio achieves an accuracy of 39.3% compared to 36.6% for the original CLIP trained on the full data.

Next we consider the comparison with FLIP, which is the same as CLIP except that it uses a 75% image masking ratio (49 image tokens). On CC3M, fine-tuned CLIPF is competitive with or outperforms fine-tuned FLIP at all masking ratios. On CC12M, fine-tuned CLIPF outperforms fine-tuned FLIP, except for the highest masking ratio (87.5% corresponding to 4 input tokens).

Finally, we consider the pattern of the text masking approaches as the masking ratio increases, i.e., as we decrease the input size from 16 to 4 tokens. Here, we focus on the results of the fine-tuned model. Moving from 16 to 8 input tokens benefits most approaches. Recall that the reduction of the number of input tokens allows for an increase of the batch size. Moving from 8 to 6 to 4 tokens, we see that CLIPF maintains its performance surprisingly well. As the masking ratio becomes higher, the margin by which CLIPF outperforms other masking strategies increases. In short, CLIPF is better positioned to deliver benefit as the number of input tokens decreases, reflecting the strength of frequency information in identifying important words to retain during masking.

5.2 Zero-shot performance on different data sets

Following the methodology outlined in CLIP and FLIP [3, 8], we evaluated various masking strategies across data sets to identify their robustness. Table 6 presents results for zero-shot classification for six commonly used data sets. The performance of these strategies on multiple datasets is similar to the behavior observed on the ImageNet-1K

Table 7: **Zero-shot Image-Text Retrieval.** We evaluated CLIP, FLIP, CLIPA, and CLIPF image-text retrieval performance on COCO and Flickr30k datasets. The models are pre-trained on **CC12M** [15] for 30 epochs with image masking (75%) to speed up training and fine-tune the model an additional epoch without image and text masking.

Models	Masking	Text Tokens	Text Retrieval		Image Retrieval	
			Flickr30k R@1	COCO R@1	Flickr30k R@1	COCO R@1
CLIP	✗	32	62.62	35.54	45.42	24.22
FLIP	✗	32	54.73	29.34	38.62	20.88
CLIPA	truncation	16	44.67	25.54	34.99	19.64
	random	16	**58.48**	**32.36**	43.61	**22.81**
	block	16	56.51	30.82	**44.06**	22.66
	syntax	16	54.54	29.60	41.16	21.40
CLIPF	frequency	16	57.89	31.52	42.72	22.57
CLIPA	truncation	8	30.47	16.92	23.96	12.78
	random	8	58.58	**36.24**	43.79	23.16
	block	8	60.06	35.88	**45.98**	**24.65**
	syntax	8	50.30	29.64	38.46	20.28
CLIPF	frequency	8	**60.36**	35.28	44.81	23.66
CLIPA	truncation	6	32.25	16.18	22.39	11.93
	random	6	55.23	32.58	**42.47**	21.34
	block	6	54.44	**33.12**	41.68	**21.73**
	syntax	6	46.15	26.78	34.08	17.99
CLIPF	frequency	6	**56.02**	32.32	41.05	21.28
CLIPA	truncation	4	30.97	17.44	21.09	11.07
	random	4	41.62	24.14	30.26	15.63
	block	4	40.83	23.56	30.77	14.98
	syntax	4	38.66	21.68	26.08	14.14
CLIPF	frequency	4	**45.07**	**25.60**	**31.72**	**15.96**

dataset. Across different text masking ratios and across different data sets, CLIPF outperforms the other masking strategies in nearly all cases. Reducing the text tokens in CLIPF from 16 to 8 results in substantial performance improvements compared to the other text masking approaches across almost all datasets: ImageNet-A [30] by 1.28%, ImageNet-O [31] by 0.80%, ImageNet-R[32] by 4.9% , ImageNet-Sketch [33] by 3.14%, ImageNet-V2 [34] by 3.38%, and ObjectNet [35] by 0.44%.

Again, we see that the benefits of CLIPF are particularly evident as the masking number of input tokens decreases. In other words, we observe that the gap in performance widen between CLIP and the other strategies as the masking ratio increases.

We also observe that CLIPF is able to improve over the FLIP baseline without masking in a majority of cases. However, despite being the strongest text-matching strategy, the results reveal that it is challenging for CLIPF to improve over the original CLIP. In only a handful of cases does it outperform the original CLIP. However, it is important to recall that training with full images, rather than using image masking, can also be combined with text masking. We did not test this setting here for computational reasons, but depending on the computational resources in an application setting it could be relevant.

5.3 Zero-shot Image-Text Retrieval

Table 7 shows image/text retrieval results on Flickr30k [36] and COCO [37]. These results demonstrate the ability of text masking to improve the performance of a VLM. The performance improvement is particularly notable in comparison to FLIP, which like our CLIPA and CLIPF uses a 75% image-masking ration. When using 16 text tokens, all strategies except the truncation method outperform FLIP which is pre-trained on the full text. The performance is also quite strong in comparison to CLIP, which uses no masking. When using 16 text tokens, text masking strategies, except for truncation, are able to approach and in a few cases beat the accuracy of CLIP.

In contrast to the previous experiments, with image-text retrieval, we observe that CLIPF does not clearly dominate the other text masking strategies. The CLIPA random, CLIPA block, and CLIPF (frequency) strategies exhibit comparable performance in image and text retrieval across both datasets, but CLIPF does not consistently outperform the other masking strategies. The performance of the syntax strategy is similar to that of FLIP but is lower than that seen with the

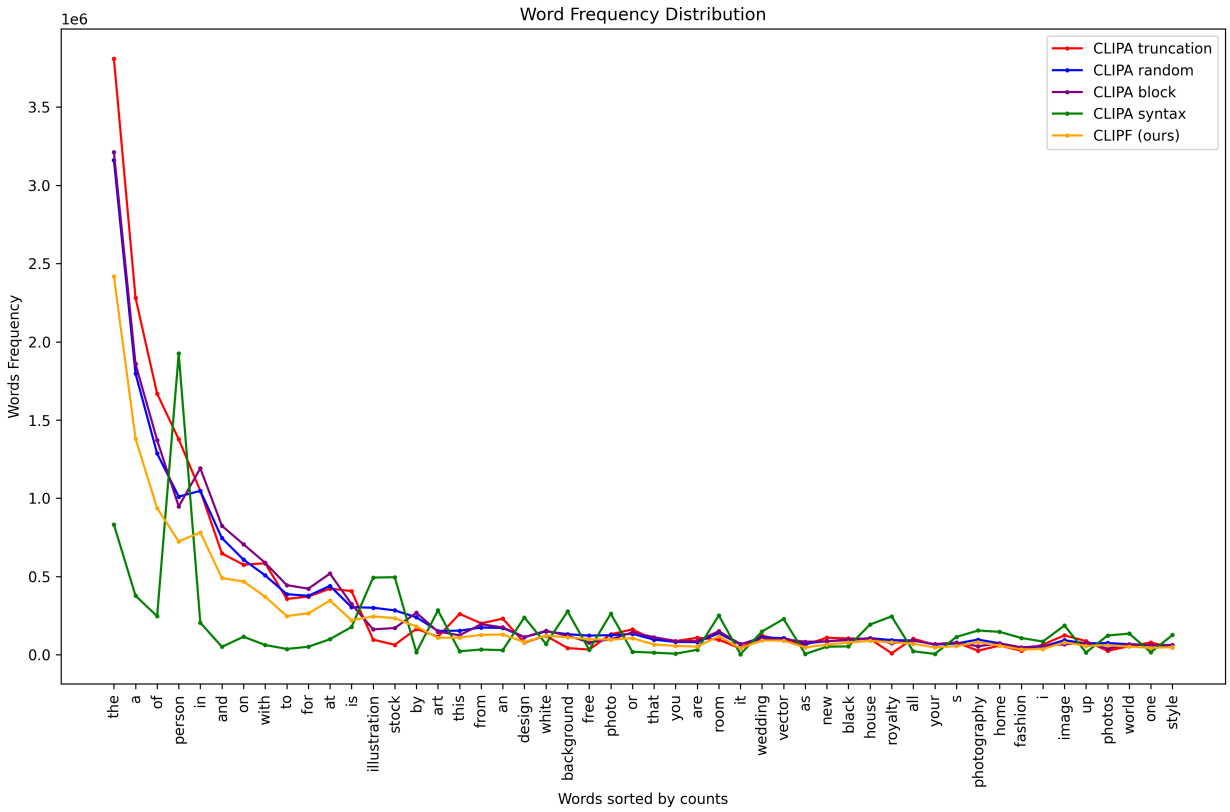

Figure 2: The figure illustrates the distribution of top-50 words in the text before and after applying various text masking strategies, we set the text length after text masking to 6. The x-axis represents the word index which is sorted by counts of the original data, and the y-axis shows the word frequency. The dataset used is CC12M and the value of t of formular 2 is set to 10^{-6}. We remove special characters from the vocabulary.

random, block, and CLIPF. The notable exception is the performance of CLIPF in the case that the text masking ratio is very high, i.e., only 4 input text tokens are used. Here, the benefit of CLIPF's frequency-based approach is clearly evident.

6 Analysis and Further Experiments

6.1 Word Distribution Analysis

In this section, we provide an analysis of how different text masking strategies affect the distribution of words By looking at the differences in distribution we can gain insight into why some text-masking strategies work well, and some less well, and, in particular, why CLIPF delivers the best performance.

Fig. 2 shows the distribution of words in the text before and after applying various text masking strategies. We see that in general there is quite a large difference between the frequency of the highest-frequency words and the frequency of words with lower frequency. Such a long-tail distribution is characteristic of words occurring in text, which is known to be Zipf distributed. However, we also observe differences between the different text masking approaches.

The truncation strategy leads to frequent words being more frequent than they are with the other strategies. This effect might seem surprising at first, but can be explained if we consider that texts typically start with the same words, e.g., "A person is..." We conjecture that truncation does not perform well as a text-masking strategy because the VLM overfits to these words.

The CLIPA random and block strategies exhibit nearly identical distributions, since each word has a 50% probability of being masked in both cases. They lie somewhat below the CLIPA truncation line, and we describe them as being more "balanced" not in the sense that the distributions are flat, but rather that the differences between the frequencies of very

Table 8: To examine the distribution of words belonging to different POS categories, we calculate the percentage of words in the training data belonging to each category in before and after applying text masking. The dataset is CC12M, and we retain 6 words for each text after applying text masking.

Masking	**NN (%)**	**JJ (%)**	**VB (%)**	**OTHER (%)**
Before masking	50.30%	4.98%	5.18%	39.55%
Truncation	51.72%	5.17%	5.91%	37.20%
Random	51.21%	4.82%	5.04%	38.93%
Block	49.37%	4.74%	5.24%	40.66%
Syntax	88.53%	2.69%	2.92%	5.87%
CLIPF	60.43%	5.06%	6.20%	28.24%

high-frequency words and the of the rest of the words is less extreme. Even more balanced is the CLIPF distribution, which lies below CLIPA random and block.

The distribution of the CLIPA syntax strategy is quite different. Because this strategy retains more nouns, the articles "the" and "a" or less frequent, but the nouns create spikes. The rationale for prioritizing nouns is convincing. Nouns are closely related to the depicted content of images and for this reason are considered to be more helpful. Indeed, the helpfulness of nouns for the VLM is reflected in the fact that Syntax masking learns faster than other text masking strategies. In Fig. 1 we saw that syntax masking outperforms the other masking approaches during the early epochs and CLIPA [12] found that syntax masking was the best approach, while testing on a limited number of epochs. However, we believe that retaining nouns can have two adverse effects. First, the model can overfit to certain high frequency nouns. In Fig. 2, we see that CLIPA syntax retains more occurrences of "person," "illustration," and "stock" than the other strategies. Second, retaining nouns means necessarily dropping words from other part-of-speech (POS) categories.

To dive deeper into the importance of the distribution of words over POS categories, we provide statistics for each of the different text masking strategies in Table 8. Consulting the column 'Other' we see and interesting pattern. CLIPA syntax masking changes the proportion of words in the "Other" category radically with respect to the data before masking, but CLIPA truncation, random and block do not. CLIPF appears to provide a happy medium. CLIPF retains more nouns than CLIPA truncation, block, and syntax, but not so many that the "Other" category becomes poorly represented. We believe that it is this balance that this balance is an important factor in CLIPF's strong performance.

6.2 Learning Curve

Next, we return to consideration of the learning curve that compares CLIPF with the other text-masking strategies, which is shown in Fig. 1. As already mentioned, syntax is a strong performer in the earlier epochs, until epoch 20 remains the best of the CLIPA approaches. We conjecture that CLIPA block and CLIPA random pull ahead of syntax because they retain "Other" words (i.e., words that are not nouns, adjectives or verbs) and do not overprioritize nouns.

The importance of "Other" words is surprising because the category includes many words, which like articles, are said to lack semantic content and have no direct connection to images. In fact, in many NLP applications, articles and other "stop words" are simply removed. We believe that the transformer that CLIP uses to embed text is actually able to exploit the context of content words, particularly words in "Other" category, given enough epochs. CLIPA block masking retains this context a little better than random masking. Surprisingly, the frequency masking of CLIPF also retains this context. Under CLIPA syntax masking a phrase like "light in the house" would always be reduced to "light house", but with CLIPF frequency masking the reduction would not happen in every case. In short, CLIPF achieves a productive balance in POS categories.

6.3 Comparison with SW-CLIP

In this section, we provide a comparison of CLIPF with SW-CLIP [11], which also uses frequency-based sampling but imposes a threshold on the frequency score. The effect of this threshold is that input tokens will go unused if not enough words in a training text surpass the threshold. In contrast, CLIPF calculates the word frequency by the threshold to prioritize words and maximize the input slots. For instance, using CC3M as an example, the average length of text before masking is 10.31 ± 4.7, and after SW-CLIPF masking, it is 4.26 ± 2.6 [11]. It is clear that the use of the input slots is not maximized when the input length of text tokens is set to 6 or longer. For a fair comparison with CLIPF, we pre-train SW-CLIP using the same setup: 75% image masking and 81.25% text masking. Subsequently, we fine-tuned both models with an additional epoch without any image or text masking. The results in Table 9 show that after fine-tuning CLIPF outperforms SW-CLIP.

Table 9: Comparison of the zero-shot accuracy performance of SW-CLIP and CLIPF on ImageNet-1k.

Models	Datasets	Image Tokens	Text Tokens	ViT-B/16	
				pre-train	fine-tune
SW-CLIP	CC3M	49	6	**14.9**	16.8
CLIPF				14.4	**18.2**
SW-CLIP	CC12M	49	8	35.9	38.5
CLIPF				**36.6**	**39.3**

Table 10: We pre-train CLIPF on CC12M dataset across different thresholds in Equation 2. The models are pre-trained on **CC12M** [15] for 30 epochs with image masking (75%) to speed up training and fine-tune the model an additional epoch without image and text masking.

model	Image tokens	Text tokens	Threshold	ViT-B/16	
				pre-train	fine-tune
CLIPF	49	8	1e-5	35.6	38.4
			1e-6	**36.6**	**39.3**
			1e-7	36.1	38.6

6.4 Threshold Analysis

To investigate the impact of the threshold t in Equation 2 on model performance, we pre-trained models with varying thresholds. As shown in Table 10, thresholds of $1e-5$, $1e-6$, and $1e-7$ yield comparable results, all outperforming the other text masking strategies. This indicates that CLIPF is relatively insensitive to small variations in thresholds, which are intended to maintain the word masking probability within the range of 0 to 1. However, using too small a threshold reduces the differences in text masking probabilities; therefore, we do not recommend using an excessively low threshold.

7 Conclusion and Outlook

In this paper, we have introduced CLIPF an approach for text masking based on word frequency that improves the performance of CLIP. Unlike syntax-based masking methods, CLIPF does not require part-of-speech (POS) information, which is time-consuming to process. Unlike truncation, random, and block text masking strategies, which treat all words the same, CLIPF assigns different probabilities based on word frequency.

In the zero-shot transfer evaluation, CLIPF demonstrates a notable ability to rapidly learn image and text representations when compared to a model that pre-trains on the full text and the other text masking strategies. Finally, we analyze the word distribution before and after applying text masking to provide insight into why CLIPF is effective. We establish that CLIPF balances retention of nouns, similar to syntax masking, with retention of other parts of speech, similar to block and random masking, effectively providing the best of both worlds.

Future should investigate two directions. First, CLIPF should be tested on larger data sets, that are beyond the reach of our own computational resources. The fact that CLIPF is effective both for CC3M as well as for the larger CC12M data set suggests that CLIPF scales well. We believe that CLIP could be adapted to any larger-scale image-text dataset; however, this should be shown empirically. We see no intrinsic reason why the advantages of CLIPF should not extend to large data sets, in which the frequency of the most frequent words can be expected to be even larger than for small datasets. Second, now that we have gained insights into the distributional properties of text data that appear important for successful text masking, future work can attempt to optimize masking directly to achieve these properties. In the meantime, frequency is what you need.

References

[1] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomás Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, 2013.

[2] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision*, pages 181–196, 2018.

[3] 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. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[4] 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*, 2021.

[5] Norman Mu, Alexander Kirillov, David A. Wagner, and Saining Xie. SLIP: self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, 2022.

[6] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11162–11173, 2021.

[7] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. In *Machine Learning for Healthcare Conference*, pages 2–25, 2022.

[8] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[9] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021.

[10] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[11] Mingliang Liang and Martha Larson. Subsampling of frequent words in text for pre-training a vision-language model. In *Proceedings of the 1st Workshop on Large Generative Models Meet Multimodal Applications*, page 61–67. Association for Computing Machinery, 2023.

[12] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. In *NeurIPS*, 2023.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 2013.

[14] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, 2019.

[15] 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 *IEEE/CVF Computer Vision and Pattern Recognition Conference*, 2021.

[16] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 2016.

[17] 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, 2021.

[18] 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 *Conference on Neural Information Processing Systems, Datasets and Benchmarks Track*, 2022.

[19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[20] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *International Conference on Learning Representations*, 2022.

[21] Runze Li, Dahun Kim, Bir Bhanu, and Weicheng Kuo. RECLIP: Resource-efficient CLIP by training with small images. *Transactions on Machine Learning Research*, 2023.

[22] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[23] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Annual Meeting of the Association for Computational Linguistics*, 2018.

[24] 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. In *International Conference on Learning Representations*, 2021.

[25] 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*, 2017.

[26] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[28] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, 2020.

[29] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

[30] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.

[31] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021.

[32] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[33] 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*, volume 32, 2019.

[34] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classifiers Generalize to ImageNet? In *International Conference on Machine Learning*, pages 5389–5400. Proceedings of the 36th International Conference on Machine Learning, 2019.

[35] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in Neural Information Processing Systems*, 32, 2019.

[36] 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*, pages 67–78, 2014.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[38] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[39] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014.

[40] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2012.

[41] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, 2011.

[42] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[43] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[44] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

Table 11: The example provided showcases from the CC12M dataset before and after text masking is applied, reducing the text from 32 tokens to 8 tokens. The first row displays the original text, while the subsequent rows show potential outcomes after applying text masking. The value of t of formular 2 is set to 10^{-6}.

Text	Walk of the happy young couple and Siberian dog. The handsome man is hugging the smiling red head girl
Truncation	walk of the happy young couple
Random	the happy and the is the
Block	couple and siberian dog . the
Syntax	walk dog man smiling head girl
CLIPF	siberian handsome man hugging smiling red

Table 12: The probability of masking words in the text is calculated using the formula provided in Equation 2. The example is selected from CC12M [15]. The value of t of formular 2 is set to 10^{-6}.

Word	Probability
walk	0.926171
of	0.992838
the	0.995064
happy	0.951695
young	0.957311
couple	0.941174
and	0.991695
siberian	0.75092
dog	0.960531
.	0.993678

A APPENDIX

B Text Masking Analysis

B.1 Text Masking Cases

Table 11 presents the potential text resulting from the text masking technique. Since truncation is fixed in each epoch, it may result in the loss of important information at the end of the text. Certain words tend to appear in specific positions within the text; for example, “the” and “a” are most likely to be the first words of a sentence. As shown in Figure 2, truncation retains more occurrences of “the” and “a” than other text masking strategies. In contrast, random and block strategies can generate different texts in each epoch for text data augmentation, but they may retain some words with a high frequency. Syntax masking retains the nouns in the text and remains the same in each epoch. However, this strategy may cause the model to overfit on the frequency of noun words. In contrast, CLIPF varies the text in each epoch, as words may be retained or removed according to their frequency. This strategy serves two primary purposes: it enhances text diversity and reduces the risk of overfitting to frequent words. Another advantage of CLIPF is that it can remove frequent prepositions that are less directly relevant to the objects in the image, such as “a,” “in,” “of,” and others. This helps the model focus on the most helpful aspects of the content. Table 12 shows the masking probabilities for certain words. High-frequency words such as “of" “the" “and" and “." are highly likely to be masked from the text.

B.2 Word Distribution Analysis for CLIPF

In addition, we analyzed frequency text masking strategies that varied according to the number of words used, as shown in Fig. 3. As the number of words decreased from 16 to 8 and then to 6, more frequent words were masked. However, reducing the number to 4 words led to a smaller vocabulary size, resulting in the loss of some important information. Consequently, the performance of the model pre-trained with 4 words was substantially lower compared to the model pre-trained with 6 words. We recommend setting the number of text words in a frequency-based text masking strategy to strike a balance between frequent and infrequent words and to maintain a larger vocabulary. Based on our experiments, the optimal number of text masking was found to be approximately 40-60% of the average length of the original text. This configuration helps achieve a balanced word distribution, which is beneficial for pre-training VLMs.

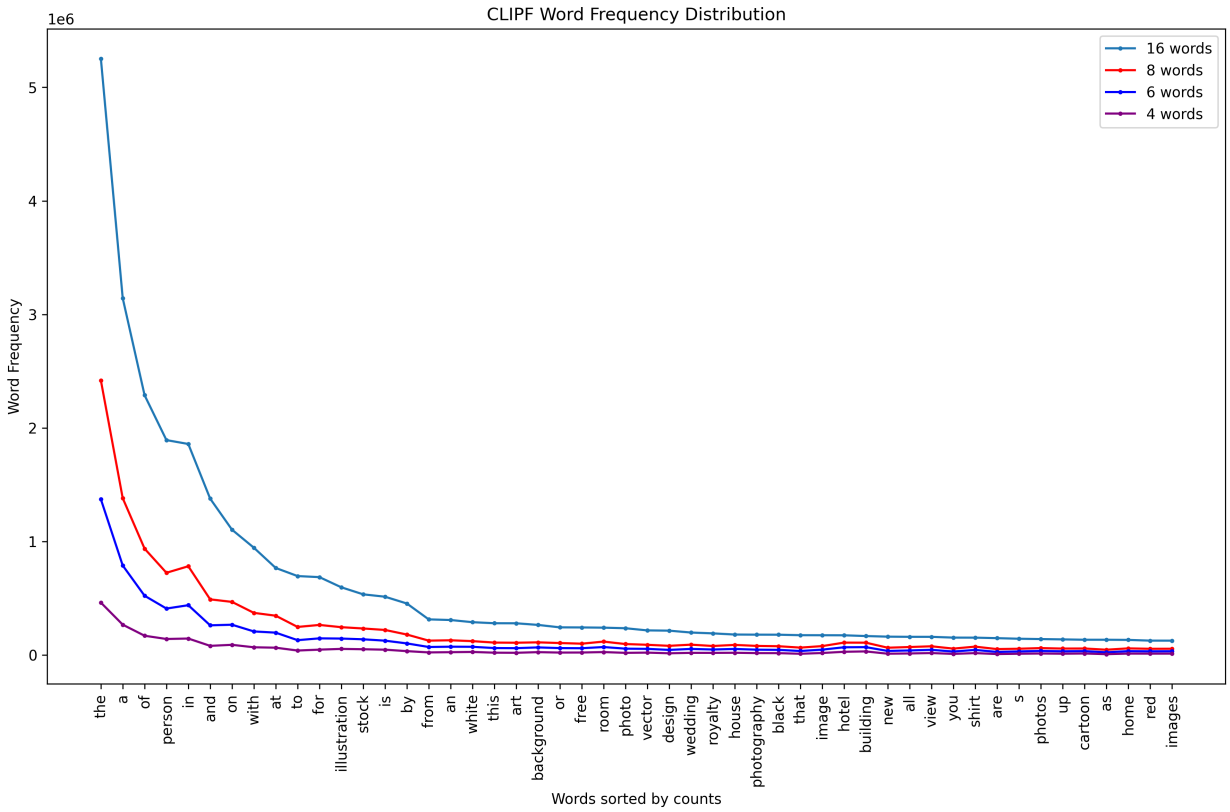

Figure 3: Frequency-based text masking strategies vary according to the number of text words used during pre-training. The dataset used is CC12M and the value of t of formular 2 is set to 10^{-6}.

Table 13: Distribution of syntax counts and percentages before and after applying text masking. The dataset is CC12M and we retain 6 words for each text after applying text masking.

Masking	NN Count	JJ Count	VB Count	OTHER Count	NN (%)	JJ (%)	VB (%)	OTHER (%)	Total
Before masking	103,469,117	10,245,828	10,649,943	81,351,966	50.30%	4.98%	5.18%	39.55%	205,716,854
Truncation	28,628,129	2,859,462	3,272,401	20,585,364	51.72%	5.17%	5.91%	37.20%	55,345,356
Random	28,334,735	2,666,847	2,790,600	21,550,517	51.21%	4.82%	5.04%	38.93%	55,342,699
Block	27,314,723	2,624,594	2,897,434	22,510,031	49.37%	4.74%	5.24%	40.66%	55,346,782
Syntax	48,989,317	1,483,666	1,618,088	3,245,892	88.53%	2.69%	2.92%	5.87%	55,336,963
SW-CLIP	34,645,367	3,346,676	4,073,825	6,062,345	71.98%	6.95%	8.47%	12.60%	48,128,213
CLIPF	33,516,439	2,803,409	3,473,119	15,666,310	60.43%	5.06%	6.20%	28.24%	55,459,277

B.3 Word Distribution of CLIPF and SW-CLIP

We also compare the word distributions between CLIPF and SW-CLIP. As shown in Figure 4, SW-CLIP exhibits a flatter word distribution than CLIPF.

B.4 Word Category Distribution Across Different Text Masking Strategies

As shown in Table 13, we provide the word type counts of Table 8. After applying text masking strategies such as truncation, random, block, syntax, and CLIPF, the word counts remain similar. However, SW-CLIP does not maximize the use of the input slots, utilizing only 85% of the words compared to other text masking strategies. Additionally, SW-CLIP masks a high percentage of other type words, which may impact the model's zero-shot ability.

Figure 5 shows the word cloud before and after applying text masking. In these visuals, the font size represents the relative frequency of each word. Words such as "the," "of," and "person" appear larger than others, indicating their higher frequency. After applying frequency word masking, the word cloud in Table 5(b) shows a more balanced distribution of word frequencies compared to before the masking. The word "the" appears smaller than in other word

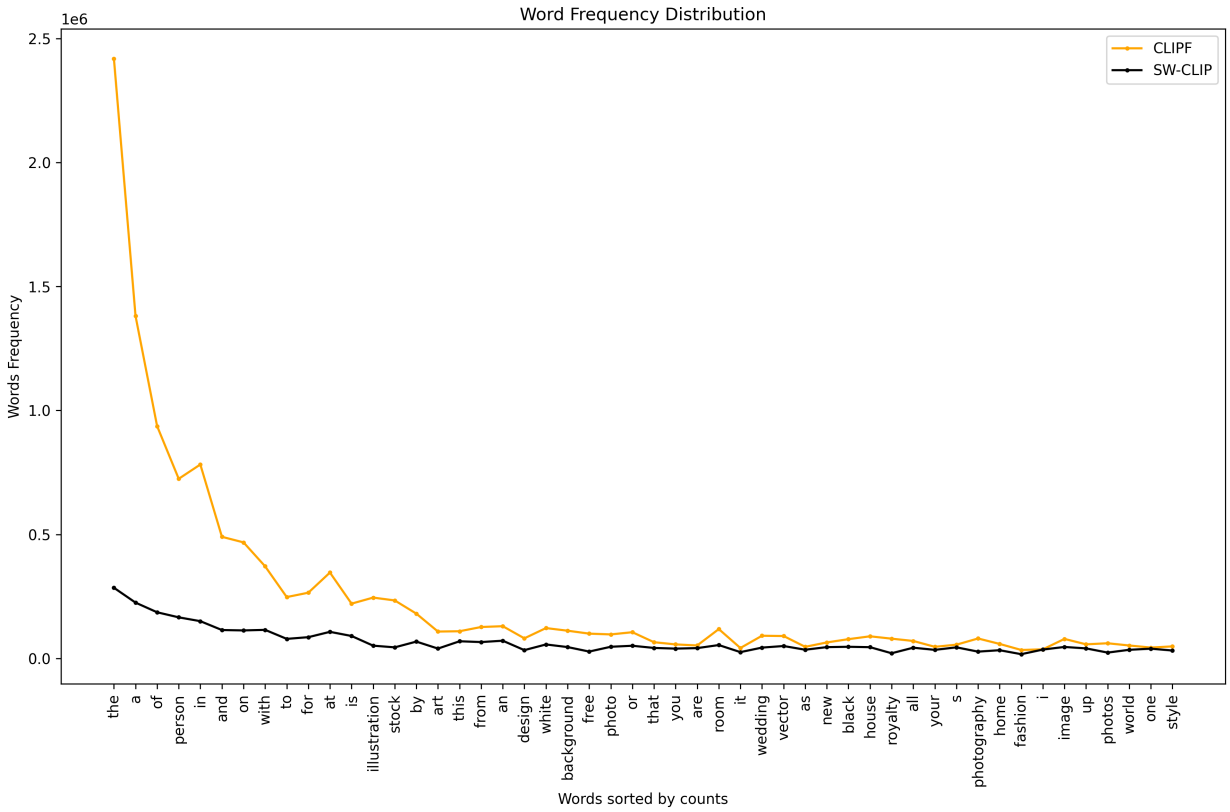

Figure 4: The figure illustrates the distribution of top-50 words in the text for SW-CLIP and CLIPF.

clouds. The reduction in frequent words allows infrequent words to become more prominent. The truncation, random, and block masking techniques result in word cloud patterns similar to those seen with the full-text. As depicted in Table 5(f), syntax masking tends to retain more nouns such as “person," “stock," and “illustration" that show a different pattern from the others.

Figure 5 shows word clouds before and after applying text masking. In these visuals, the font size represents the relative frequency of each word. Words such as “the,” “of,” and “person” appear larger than others, indicating their higher frequency. After applying word frequency masking, the word cloud in Figure 5(b) displays a more balanced distribution of word frequencies compared to before masking. The word “the” appears smaller than in other word clouds. The reduction in frequent words allows infrequent words to become more prominent. The truncation, random, and block masking techniques result in word cloud patterns similar to those seen with the full text. As depicted in Figure 5(f), syntax masking tends to retain more nouns such as “person”, “stock” and “illustration” resulting in a different pattern from the others.

B.5 Bi-gram word Distribution

As shown in Figure 6, we also calculated the bigram distribution of the text after applying different text masking strategies. Truncation and block masking better retain information about the order of the words in the text because the words remain contiguous after masking. Random and CLIPF masking also preserves the text order better than syntax masking. Syntax masking retains more continuous bigrams—such as “stock illustration,” “vector illustration,” and “stock photography”—than other text masking strategies. These bigrams might be domain specific and not contribute to the transferrability of the VLM to other data sets.

B.6 Impact of Symbolic Characters

We note the high frequency of symbolic characters appearing in the dataset, these symbolic characters are not directly related to the depicted content of the images. Since CLIPF tends to mask these characters with high probability, we

(a) Full Text (b) Frequency

(c) Truncation (d) Random

(e) Block (f) Syntax

Figure 5: The word cloud on the left displays the full text, while the one on the right shows the text after applying text masking, where we retain 6 words. The font size of the word represents the relative word frequency for each figure. The dataset used is CC12M and the value of t of formular 2 is set to 10^{-6}.

Table 14: **We remove symbolic characters and stop words during the model's pre-training to evaluate their zero-shot classification performance on ImageNet-1K.** The backbone of the image encoder is ViT-B/16, and the model is pre-trained on **CC3M** [23] for 30 epochs with image masking (75%) to speed up training and fine-tune the model additional epoch without image and text masking.

Models	Masking	Image Tokens	Text Tokens	Remove	ViT-B/16	
					pre-train	fine-tune
FLIP	✗	49	32	✗	14.0	14.2
FLIP	✗	49	32	✓	13.9	13.7

removed them during the pre-training of the model to evaluate their significance. As shown in Table 14, removing symbolic characters and stop words does not affect the model's performance.

B.7 Word and Token Analysis

Since CLIP encodes text using byte-pair encoding (BPE), some words are encoded into more than one token. Therefore, in this experiment, we calculate the masking probability based on tokens rather than words. As shown in Table 15, we compare the performance of both approaches.

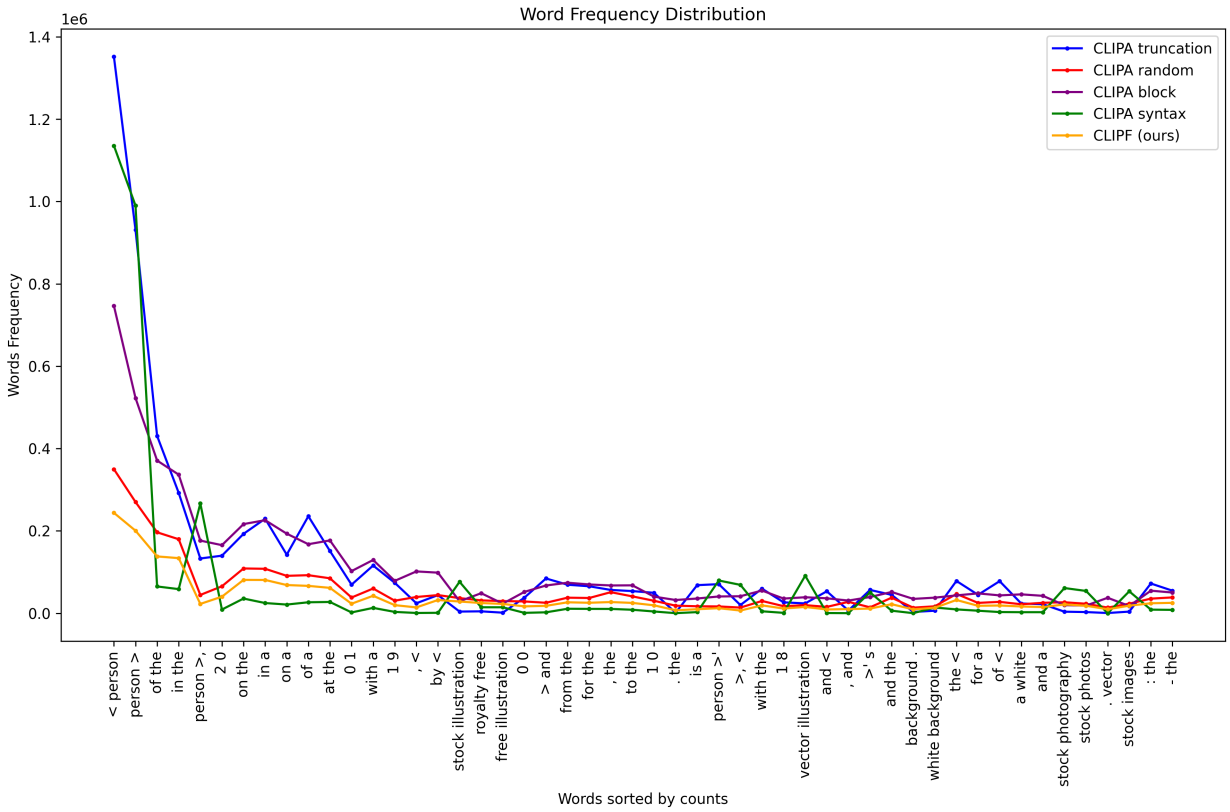

Figure 6: The figure illustrates the distribution of top-50 bi-grams in the text with different text masking strategies.

Table 15: **Comparison of CLIPF pre-trained with token masking and word masking on ImageNet-1K for zero-shot classification.**

Models	Masking	Image Tokens	Text Tokens	ViT-B/16	
				pre-train	fine-tune
CLIPF	frequency-token	49	8	36.0	38.0
	frequency-word	49	8	36.6	**39.3**

Table 16: **Zero-shot accuracy on more classification datasets**. Comparison of the zero-shot accuracy performance of CLIP, FLIP, CLIPA, and CLIPF on various datasets.

Models	Method	Text Tokens	Food-101	CIFAR-10	CIFAR-100	CUB200	SUN397	Cars	Aircraft	DTD	OxfordPets	Caltech-101	Kinetics700	Flowers102	MNIST	STL10	EuroSAT	Resisc45	GTSRB	KITTI	Country211	PCAM	UCF101	CLEVR	HatefulMemes	SST2	ImageNet
CLIP	✗	32	45.46	66.42	32.45	7.90	49.27	16.17	2.28	15.48	57.51	71.91	24.84	2.12	8.77	91.29	19.08	37.38	5.09	29.81	4.45	50.05	40.52	21.84	52.66	48.76	36.61
FLIP	✗	32	36.26	58.09	26.89	8.73	38.56	8.16	2.98	19.95	51.85	67.17	21.57	2.94	9.35	81.89	18.02	32.30	**9.68**	21.39	3.63	50.26	36.51	14.67	51.37	**50.08**	33.70
CLIPA	truncation	16	35.53	61.24	26.54	9.42	39.98	7.28	2.57	15.80	46.85	68.28	20.47	1.94	8.32	81.89	18.48	33.75	2.33	17.98	3.79	58.56	30.85	13.31	53.00	**50.08**	32.83
	random	16	38.48	53.61	29.46	7.97	41.92	8.53	**2.79**	21.54	**56.71**	69.67	22.88	2.23	9.73	84.86	17.90	**36.41**	7.28	32.95	**4.77**	55.00	35.29	16.38	49.93	**50.08**	34.26
	block	16	38.69	57.51	29.71	8.23	44.54	**9.85**	2.45	21.33	54.01	70.61	23.00	2.79	10.09	**85.04**	**31.88**	29.98	**9.07**	25.07	4.59	**60.68**	**38.07**	**18.72**	51.24	49.64	34.78
	syntax	16	35.94	58.78	31.86	8.91	41.92	8.39	1.53	17.07	47.05	67.59	21.34	**3.10**	**10.45**	76.92	26.68	32.65	8.73	**39.37**	4.23	49.96	32.91	15.55	**53.30**	49.92	34.41
CLIPF	frequency	16	**39.47**	**64.43**	**32.55**	**10.15**	**45.51**	9.22	2.43	**21.70**	55.62	**70.69**	**23.08**	2.77	9.56	83.00	21.02	33.30	7.20	27.47	4.12	50.24	37.72	13.70	52.70	49.92	**35.96**
CLIPA	truncation	8	36.28	61.99	32.24	8.65	44.40	5.56	2.04	18.56	45.67	65.93	22.68	2.52	10.51	87.96	34.80	29.37	3.35	**46.06**	4.59	51.55	32.38	12.41	53.46	50.08	30.30
	random	8	45.10	**68.54**	**34.04**	9.03	**51.54**	8.64	3.23	**30.53**	59.29	70.06	**27.29**	**4.21**	9.82	90.46	28.72	32.54	5.99	30.61	**7.95**	58.00	42.90	12.87	53.40	50.08	37.25
	block	8	43.70	66.66	32.62	8.08	51.07	9.27	2.54	28.99	57.88	70.89	26.72	2.65	**18.10**	**90.84**	**31.88**	**36.94**	**9.02**	33.36	7.52	**61.91**	40.10	**14.66**	51.80	**53.27**	37.88
	syntax	8	37.92	65.25	31.72	8.28	45.74	8.73	2.96	17.07	46.54	68.73	23.52	2.59	11.30	88.30	29.90	30.81	6.47	30.82	5.02	54.62	36.58	12.51	52.74	50.08	34.97
CLIPF	frequency	8	**45.75**	59.00	30.76	**10.60**	50.06	**10.32**	**3.20**	29.89	**67.54**	**73.35**	26.96	4.09	10.10	90.54	20.80	36.30	7.16	40.31	7.10	57.23	**43.09**	12.27	**54.54**	50.03	**39.30**
CLIPA	truncation	6	23.54	60.27	23.52	5.70	36.54	3.82	2.03	10.32	31.12	61.15	17.77	1.86	9.94	85.25	20.62	23.68	7.13	16.71	3.45	50.23	24.53	12.61	53.62	50.08	23.28
	random	6	38.45	**71.57**	33.89	7.73	51.81	6.67	2.72	29.47	59.92	68.65	25.89	2.42	**12.28**	89.73	26.90	33.17	8.89	29.75	7.23	56.87	38.62	**12.63**	51.96	50.08	34.62
	block	6	39.87	66.60	**35.31**	7.58	**53.23**	7.49	**2.89**	27.87	56.74	**69.40**	25.97	**2.95**	10.06	89.30	27.90	32.40	7.31	33.76	6.73	**63.10**	39.20	12.65	**55.41**	**52.83**	35.89
	syntax	6	34.18	63.46	29.83	7.54	43.54	7.92	1.85	17.23	44.97	68.37	22.75	1.99	11.37	89.19	17.56	26.13	3.84	33.16	4.98	55.25	35.40	12.30	53.44	50.80	32.57
CLIPF	frequency	6	**45.33**	67.64	32.83	**9.98**	50.86	**9.61**	2.19	**31.33**	**66.03**	69.24	**26.60**	2.94	10.71	**90.98**	**33.64**	**33.22**	**8.93**	**36.70**	**7.32**	59.96	**40.60**	12.45	54.15	**50.08**	**37.81**
CLIPA	truncation	4	20.64	51.67	22.10	4.19	37.16	3.11	1.18	15.16	29.36	53.99	16.17	1.04	10.10	86.34	17.16	15.81	7.48	30.15	3.49	**61.43**	24.95	12.63	50.76	50.08	19.80
	random	4	28.82	61.90	28.98	5.35	46.39	4.03	1.42	25.16	43.26	62.87	22.03	2.10	**12.24**	88.54	23.82	24.46	**8.07**	39.30	5.59	50.62	33.52	12.65	51.86	50.08	27.13
	block	4	28.08	60.63	27.10	5.09	45.61	3.82	2.43	23.24	36.63	64.15	21.80	2.29	8.41	87.04	24.44	23.95	7.55	**43.05**	5.51	55.80	32.62	9.49	**53.55**	50.08	26.66
	syntax	4	25.59	54.13	27.46	4.73	42.26	6.26	**2.66**	15.27	32.52	58.93	19.58	2.20	10.42	90.11	**26.02**	20.62	5.19	37.50	4.32	51.12	26.82	**12.93**	52.77	50.08	24.61
CLIPF	frequency	4	**36.09**	**69.76**	**31.14**	**8.41**	**47.72**	**7.01**	2.24	**25.64**	**54.23**	**64.55**	**23.55**	**2.95**	10.18	**90.25**	20.78	**26.52**	6.90	36.63	**6.40**	55.71	**35.71**	12.65	49.63	50.08	**30.93**

Table 17: **Zero-shot Image-Text Retrieval,** we evaluated CLIP, FLIP, and CLIPF image-text retrieval performance on COCO and Flickr30k datasets. The backbone of the image encoder is ViT-B/16, and the model is pre-trained on CC12M for 30 epochs with image masking (75%) to speed up training and fine-tune the model additional epoch without image and text masking.

Models	Masking	Text Tokens	Text Retrieval						Image Retrieval					
			Flickr30k			COCO			Flickr30k			COCO		
			R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP	✗	32	62.62	86.00	91.81	35.54	62.38	74.08	45.42	72.56	81.50	24.22	48.42	60.42
FLIP	✗	32	54.73	80.37	87.97	29.34	56.08	67.00	38.62	66.09	75.40	20.88	43.22	54.78
CLIPA	truncation	16	44.67	73.08	81.85	25.54	51.90	64.64	34.99	61.05	70.85	19.64	41.91	53.51
	random	16	**58.48**	**83.93**	**90.53**	**32.36**	**58.76**	69.98	43.61	70.67	80.04	**22.81**	46.00	57.71
	block	16	56.51	81.26	89.05	30.82	58.32	**70.38**	**44.06**	**71.20**	**80.10**	22.66	**46.24**	**58.06**
	syntax	16	54.54	81.07	88.36	29.60	56.52	68.54	41.16	68.32	77.51	21.40	44.82	56.56
CLIPF	frequency	16	57.89	84.62	90.04	31.52	58.38	70.30	42.72	69.61	78.60	22.57	46.15	57.95
CLIPA	truncation	8	30.47	59.47	70.51	16.92	38.62	51.34	23.96	47.29	57.85	12.78	31.66	42.92
	random	8	58.58	84.52	91.81	**36.24**	62.16	72.90	43.79	70.89	80.14	23.16	46.74	58.86
	block	8	60.06	85.01	**92.11**	35.88	62.58	**73.74**	**45.98**	**73.23**	**82.49**	**24.65**	**48.81**	**60.70**
	syntax	8	50.30	78.30	86.98	29.64	55.42	67.18	38.46	66.77	77.36	20.28	43.78	55.40
CLIPF	frequency	8	**60.36**	**85.80**	90.63	35.28	**62.70**	73.70	44.81	72.07	81.34	23.66	48.05	59.89
CLIPA	truncation	6	32.25	61.05	72.98	16.18	39.52	52.48	22.39	47.46	59.39	11.93	30.03	40.86
	random	6	55.23	82.54	89.25	32.58	57.54	68.68	42.47	70.12	**79.43**	21.34	44.79	56.26
	block	6	54.44	79.68	88.66	33.12	58.96	70.64	41.68	69.13	79.33	21.73	45.23	57.29
	syntax	6	46.15	76.04	84.81	26.78	52.56	64.62	34.08	61.76	73.25	17.99	40.22	52.05
CLIPF	frequency	6	56.02	82.05	88.56	32.32	58.58	70.00	41.05	69.17	78.88	21.28	44.55	56.05
CLIPA	truncation	4	30.97	56.51	67.06	17.44	38.48	49.76	21.09	43.77	55.70	11.07	27.98	38.54
	random	4	41.62	69.53	80.18	24.14	48.32	60.14	30.26	56.92	68.42	15.63	35.72	47.30
	block	4	40.83	69.03	79.09	23.56	47.94	59.20	30.77	57.85	69.29	14.98	35.64	47.35
	syntax	4	38.66	65.88	75.74	21.68	44.38	56.60	26.08	52.80	64.32	14.14	33.71	44.98
CLIPF	frequency	4	45.07	72.49	81.95	25.60	49.98	61.96	31.72	59.35	71.66	15.96	36.58	48.32

C More results

More detailed results are presented in this section.

C.1 Zero-shot Classification on More Datasets

In Table 16, we present the evaluation results of CLIPF following the approach of CLIP [3] on additional datasets. In almost all of the downstream datasets, when using six text tokens, CLIPF surpasses not only the baseline CLIP without image and text masking and FLIP with image masking and full-text tokens but also outperforms other text masking strategies. Similar to observations from the ImageNet datasets, reducing the number of text tokens from 16 to 8 substantially enhances CLIPF's performance across all datasets. Specifically, performance improvements are noted: SUN397 [38] by 6.39%, DTD [39] by 9.31%, OxfordPets [40] by 4.64%, STL10 [41] by 9.31, EuroSAT [42] by 6.60%, UFC101 [43] by 6.68% and ImageNet-1K [44] by 3.13%. The performance of other text masking strategies, including truncation, random, block, and syntax, also improves with this reduction. However, the performance gains are not as large as those observed with CLIPF. This improvement suggests that employing text masking strategies as a data augmentation technique can enhance VLMs' performance in zero-shot classification tasks. CLIPF, utilizing a word frequency masking method, demonstrates robust zero-shot classification performance and establishes a benchmark for state-of-the-art results.

C.2 Zero-shot Image-text Retrieval

We report the details of zero-shot image-text retrieval for Flickr30k and COCO datasets, the details are shown in Table 17.

C.3 Learn Curved before fine-tuning

In Figure 1, we show the learning curve after fine-tuning. Moreover, we show the learning curve before fine-tuning, they have a similar performance for every text masking strategy.

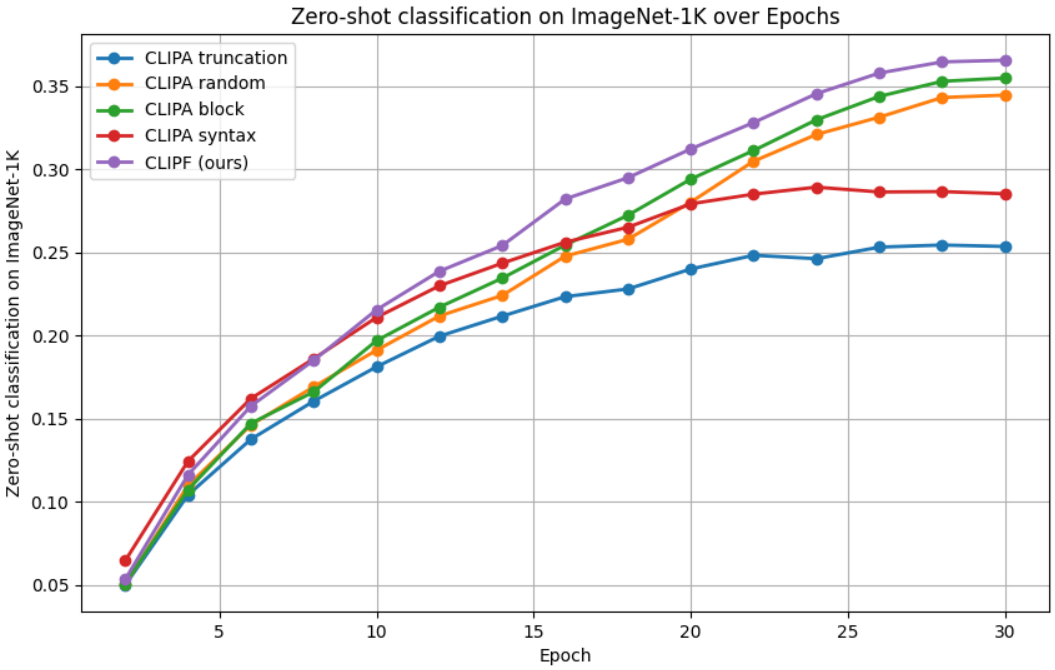

Figure 7: **Zero-shot classification accuracy on ImageNet-1K** over the epochs for the different text masking strategies before fine-tuning.