Classification Done Right for Vision-Language Pre-Training

Zilong Huang **Qinghao Ye** **Bingyi Kang** **Jiashi Feng** **Haoqi Fan**
ByteDance Research
Code & Models: x-cls/superclass

Abstract

We introduce SuperClass, a super simple classification method for vision-language pre-training on image-text data. Unlike its contrastive counterpart CLIP [57] who contrast with a text encoder, SuperClass directly utilizes tokenized raw text as supervised classification labels, without the need for additional text filtering or selection. Due to the absence of the text encoding as contrastive target, SuperClass does not require a text encoder and does not need to maintain a large batch size as CLIP [57] does. SuperClass demonstrated superior performance on various downstream tasks, including classic computer vision benchmarks and vision language downstream tasks. We further explored the scaling behavior of SuperClass on model size, training length, or data size, and reported encouraging results and comparisons to CLIP .

1 Introduction

Pretraining methodologies [35, 57, 51, 60] that directly harness web-scale image-text dataset have transformed the field of computer vision in recent years. Among them, contrastive language image pretraining (CLIP) [57] has gained escalading popularity and become predominant due to the following reasons. First, it serves as the industry-standard pre-trained model that facilitates zero-shot visual recognition [50, 52] and finetuning on downstream tasks [19, 17]. Second, proper scaling behaviors [12] are observed such that CLIP can consistently benefit from larger models and bigger data to some extent. Third, it offers strong cross-modal abilities as it is inherently designed to understand and connect information across text and images. Therefore, CLIP-style models are the default choices for most modern Visual Language Models [47, 2, 1], which connect a vision backbone with a deep language model [69, 13].

Despite its success, CLIP necessitates very large batch sizes for training—typically over 64,000—to achieve optimal performance, along with substantial computational resources for text encoding. This high computational demand limits accessibility for researchers with limited resources and engineering expertise. In our work, we aim to address the heavy computational burden by replacing contrastive methodology with a simpler classification approach, eliminates the need for large contrastive batch sizes, and text encoders.

In this work, we revisit the classification method for pretraining on large-scale text-image pairs. Some previous works [54, 31, 39, 27, 51] attempt to tackle this by employing bag-of-words classification in a weak supervised learning manner. However, most of these studies have been conducted on a small scale, and there is no evidence demonstrating their scalability in terms of data and model size. In contrast, our method demonstrates the performance of SuperClass on a scale comparable to CLIP [57], achieving favorable model performance with 13 billion seen samples on 1 billion unique text-image pairs. Some other concurrent efforts [31] have also attempted to replace contrastive learning with classification. However, they rely heavily on preprocessing the text modality, using

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

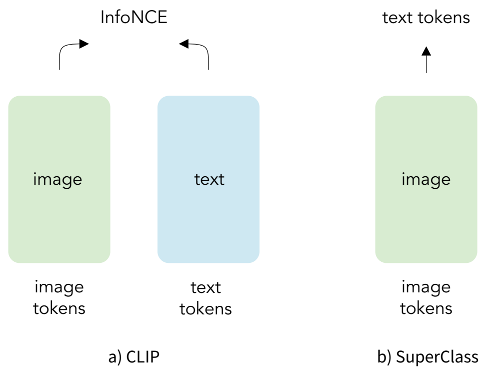

Figure 1: (left) CLIP uses two separate Transformer encoders to extract vector representations from image-text pairs. The text encoder operates on a subword-level tokenizer. (right) The proposed bag of subwords classification only uses the single Transformer encoder.

bag-of-words and other hand-crafted rules to convert text into semi-labels. Some common practices include filtering, word segmentation, lemmatization, and the removal of numbers and stopwords to create a unique vocabulary of clean words. We found the preprocessing often eliminates long-tailed words or stopwords that contain valuable information for representation learning (see Sec. 4.4). In contrast, SuperClass simply utilizes raw word tokens as supervision signals without requiring any hand-crafted preprocessing: no filtering or removal of stopwords. Hence SuperClass preserves all information from the original text descriptions as supervision signal.

We proposed a **Super**-simple-**Class**ification approach (**SuperClass**) that simply trains to classify raw text tokens and scales as good as CLIP. As shown in Figure 1, similar to CLIP, SuperClass directly operate on text tokens with any manual text filtering. Our comprehensive empirical study shows that even without the need for a text encoder, classification methods can achieve performance comparable to the contrastive approach in terms of both model capabilities and data scalability. We demonstrate that SuperClass is a competitive alternative to its contrastive counterpart on both image classification and vision & language tasks. Pretrained on the same Datacomp-1B [21] datasets with an equal number of seen samples, SuperClass dominantly outperforms its contrastive counterparts across various of vision only and vision & language scenarios. We further explore the scaling behavior of SuperClass concerning model size and number of seen samples. Experiments suggest that classification-based methods can exhibit competitive or even superior scaling behavior compared to their contrastive counterparts. We hope our method, experiments and analysis can encourage future potentials of classification-based methods as the foundational vision-language pretraining methods.

2 Related Work

With the growing availability of large-scale, web-sourced image-text datasets [57, 65, 6, 68, 21, 63, 4, 62], new methods have emerged to leverage this data as supervision for training deep representations. These approaches typically involve one of three strategies: using text as classification labels, implementing image-text contrastive learning, or treating text as autoregressive targets.

Text as classification labels. The exploration of image-text data for model training has deep roots, with early work like Image-to-Word[54] over two decades ago aiming to enhance content-based image retrieval. This study pioneered efforts to train models to predict nouns and adjectives in text documents linked to images. Building on these early ideas, subsequent research has sought to improve data efficiency[68, 43], model effectiveness [31, 27], and vocabulary expansion [27, 78, 51, 39]. With the recent develop of network architecture, Tag2Text [27] and RAM [78] have employed Vision Transformers (ViT) [18] as vision backbones, extracting nouns from the CC12M dataset [6] and, through a combination of rules and manual selecting, arrived at 6,449 words to use as classification categories. Similarly, CatLIP [51] has filtered out "gold labels" from the CC3M [65] and Datacomp-1B [21] datasets based on certain rules, and then trained visual models using even larger-scale image-text pair datasets.

Unlike previous classification methods that rely on complex rules or manual filtering to curate "gold labels" for classification vocabularies, our approach eliminates the need for such filtering. Instead, we directly leverage text tokens as classification categories, preserving valuable textual information that might otherwise be discarded.

Image-text contrastive learning. Large-scale contrastive vision-language pretraining gained traction with the introduction of CLIP [57] and ALIGN [30]. Since then, numerous approaches have focused on enhancing CLIP's performance [76, 45, 44, 11, 7, 74]. For instance, SigLIP [76] reduces the computational load of CLIP's softmax-based contrastive loss by employing a sigmoid loss for local pairwise similarity calculations. LiT [77] adopts pretrained vision and language backbones with contrastive training, while other methods [45, 44, 19] aim to enhance training efficiency in image-text pretraining. InternVL [11] further innovates by integrating a large language model as the text encoder within CLIP.

In our approach, we challenge the necessity of an additional backbone to encode text information for contrastive learning. Instead, we directly use text token input as the supervisory signal, eliminating the need for text encoding and avoiding the computational overhead of large contrastive operations. This streamlined setup achieves performance comparable to dual-backbone methods.

Text as autoregressive targets. Various recent studies [15, 61, 38, 32, 41, 26] have delved into employing image captioning for model pretraining. SimVLM [71] has innovated this field by pioneering the pretraining of a multimodal encoder-decoder that fuses vision and language at an early stage, leveraging a hybrid architecture for applications such as visual question answering (VQA). CapPa [70] demonstrates that a simple encoder-decoder setup can efficiently pretrain vision encoders solely through captioning. Furthermore, recently studies [75, 42, 37] combine contrastive learning with captioning objectives, occasionally incorporating an additional text encoder.

In this work, we revisit the classification-based approach using large-scale visual-language datasets. Unlike the image captioning methods mentioned earlier, our classification method integrates the text captioning decoder within the vision encoder, allowing a single vision encoder to connect both modalities. Experiments demonstrate that SuperClass achieves competitive, and often superior, performance across various downstream tasks.

3 A simple classification approach to pretrain vision encoders

In this section, we present our proposed approach, SuperClass, which employs a classification-based pretraining method using text supervision. We begin by outlining the general framework of SuperClass. Next, we explain how text is converted into category labels without the need to select "gold labels", allowing all text to supervise the training of the image encoder. Finally, we illustrate our choice of loss design among various classification losses. Additionally, recognizing the differing significance and discriminative power of each word, we incorporated inverse document frequency as class weights in the loss design.

Overview. We aim to establish a pretraining method based on image classification that matches CLIP in simplicity, scalability, and efficiency. To achieve this, we follow the standard protocol by utilizing Vision Transformer (ViT) backbones as vision encoders, followed by a global average pooling layer and a linear layer as the classification head to output the logit vector **x**. The supervision targets are derived from the text associated with the image, and the classification loss is computed using the text-derived classification labels and the predicted logits.

Texts as Labels. We directly use tokenized text as K-hot labels, where K is the number of tokens in the given sentences. More specifically, for a given image-text dataset $\mathcal{D} = \{(I_i, T_i) \mid i \in [1, N]\}$ with N pairs of images I and text captions T, we differ from previous classification-based methods by directly using an existing subword-level tokenizer, such as the one used in CLIP or BERT, with a vocabulary size V. This tokenizer inputs the text T and obtain the set $\mathbb{C}$ of corresponding subword IDs, which serves as the classification labels. The label in the set $\mathbb{C}$ satisfies $\{c \in [1, V]\}$. The classification labels $\mathbb{C}$ will be converted into K-hot vector $\mathbf{y} \in \mathbb{R}^V$, where $y_c = 1$ when c in the set $\mathbb{C}$, otherwise $y_c = 0$.

Compared to previous methods, our approach does not require any preprocessing or manual threshold setting, making it straightforward. At the same time, it also avoids the out-of-vocabulary issue that might be encountered by previous approaches.

Classification Loss. A significant body of research has focused on multi-label classification loss. However, it is important to emphasize that our primary goal is to pretrain vision encoders rather than prioritize multi-label classification accuracy. In a multi-label scenario, a Softmax loss is applied in SuperClass by representing labels in a probabilistic manner, where $\hat{y}_c$ is a normalized weighted label.

$$\ell_{ce} = -\sum_{c=1}^{V} \hat{y}_c \log \frac{e^{x_c}}{\sum_{c'} e^{x_{c'}}} \tag{1}$$

We evaluated several multi-label classification losses, including Softmax loss, BCE loss, soft margin loss, ASL [59], and two-way loss [33]. Surprisingly, the simple Softmax loss yielded the best pretraining results. This may be due to the fact that existing multi-label classification losses assume that labels are precise and exhaustive, aiming to optimize the margin between positive and negative classes. However, the inherent noise in image-text data and the limitations of text in fully capturing an image's content mean that not all objects in an image are always referenced in the accompanying text.

Inverse Document Frequency. Within the subword vocabulary, not all categories contribute semantically equally, as different words carry varying amounts of information. Additionally, the subword dictionary contains many words unrelated to visual content that frequently appear in sentences, which do not provide effective supervisory information. Therefore, words with higher information content should be given greater weight during training. To achieve this, we employ Inverse Document Frequency (IDF) as a measure of information significance. The fewer the number of samples containing a specific word, the stronger its ability to differentiate between samples. We use the IDF statistic of each category (subword) as the weight for the corresponding classification label, assigning different weights w_c to the classification labels c.

$$\hat{y}_c = \frac{w_c y_c}{\sum_{c'} w_{c'} y_{c'}}, \quad w_c = \log\left(\frac{|\mathcal{D}|}{1 + \mathrm{df}(c)}\right) \tag{2}$$

where $|\mathcal{D}|$ denotes the total number of image-text pairs, $\mathrm{df}(c)$ is the document frequency (df) of subword c, in other words, it's the count of texts containing subword c. For greater ease of use, we have implemented an online IDF statistic that is computed during the training process, eliminating the need for pre-training offline statistics. This approach enhances the user-friendliness and portability of our method.

4 Experiment

4.1 Experiment setup

We use a standard subset of the datacomp dataset [21] for pre-training, which contains about 1.3B image-text pairs. A batch size of 16k and 90k are adopted for our classification models and CLIP models. In the ablation section, all experiments are conducted with a batch size of 16k. To make a fair comparsion with the CLIP, we use 90k batch size, adopt the AdamW with a cosine schedule, and set the same learning rate and decay as CLIP.

4.2 Evaluation protocols

In order to better highlight the effectiveness of pretraining method, we concentrate on the properties of the frozen representations.

Linear probing We evaluate the classification accuracy when using the full ImageNet-1k [14] training set to learn a dense projection layer and frozen the parameters of backbone. We follow the linear probing training recipe from MAE [24].

Table 1: Comparison of the Linear probing top-1 accuracy on ImageNet-1K dataset.

Method	**PreTraining data**	**ViT-Base**		**ViT-Large**	
		#Seen Samples	Top-1 (%)	#Seen Samples	Top-1 (%)
contrastive or clustering based					
MoCov3 [10]	IN1K	400M	76.7	400M	77.6
DINO [5]	IN1K	512M	78.2	-	-
iBOT [80]	IN22K	400M	79.5	256M	81.0
DINOv2 [55]	LVD-142M	-	-	2B	84.5
reconstruction based					
BEiT [3]	D250M+IN22K	1B	56.7	1B	73.5
SimMIM [73]	IN1K	1B	56.7	-	-
CAE [8]	D250M	2B	70.4	2B	78.1
MAE [24]	IN1K	2B	68.0	2B	75.8
vision-language pretraining based					
Openai CLIP [57]	WIT-400M	13B	78.5	13B	82.7
Cappa [70]	WebLI-1B	-	-	9B	83.0
OpenCLIP [29]	Datacomp-1B	-	-	13B	83.9
SuperClass	Datacomp-1B	1B	78.7	1B	82.6
SuperClass	Datacomp-1B	13B	**80.2**	13B	**85.0**

Table 2: Performance of frozen visual representations on different classification datasets. 10-shot linear evaluation accuracy on the pre-logit representation. *results from the paper.

Method	ImageNet	Pets	Cars
MAE	44.0±0.1	57.7±0.2	32.5±0.1
DINOv2	77.0±0.1	94.2±0.1	76.8±0.2
CapPa*	70.6±0.2	92.6±0.5	92.2±0.2
OpenCLIP	75.6±0.1	92.2±0.6	**92.7±0.3**
SuperClass	**77.2±0.1**	**94.6±0.1**	92.6±0.1

Table 3: Zero-shot Top-1 acc. and CIDEr are tested on ImageNet-1k dataset and COCO captions, respectively. The zero-shot accuracy of SuperClass is obtained after lock-image tuning [77].

Case	Backbone	0-shot	CIDEr
Openai CLIP	ViT-L/14	75.3	**113.5**
OpenCLIP	ViT-L/14	79.2	-
CLIP,reimpl.	ViT-L/16	79.0	112.6
SuperClass	ViT-L/16	**79.7**	113.0

10-shot classification Following the setting of Cappa [70], we perform 10-shot classification on ImageNet-1k [14], Pets [56] and Cars [34]. For each dataset and model, we run 3 times, and report the mean results and variance.

Locked-image Tuning Locked-image Tuning (LiT) [77] employs contrastive training to align locked image and unlocked text models. Generally, LiT is an efficient way to equip any pretrained vision backbone with zero-shot classification and retrieval capabilities. We follow the setup from LiT[77] and assess the zero-shot classification accuracy on ImageNet-1k [14].

Collaborating with language models Motivated by recent works [47, 11, 1, 67, 2, 71] combining pretrained vision backbones [57, 19, 76] and language models [69, 13], we investigate the amenability of the learned representations to interface with a text decoder. Here, we evaluate two ways to collaborate with large language models. 1) following ClipCap [53], we frozen both pretrained image encoder and pretrained 12-layer GPT2 decoder [58], only train an adapter to connect the image encoder and language model to perform image captioning on COCO captions [9]. 2) following LLaVA [47] setup, we train and finetune a projection layer and a pretrained large language models, Vicuna-7B [13] to solve downstream tasks, including VQAv2(val) [22], GQA [28], VizWiz(val) [23], T-VQA(val) [66], SQA(img) [79], MMBench(en) [48], MME [20], POPE [46], MMMU [25] and SEEDBench [40].

4.3 Main results

Comparison with different types of pretraining methods In Table 1, we compare the models trained by SuperClass with the different types of pretraining methods, including contrastive or clustering based methods [10, 5, 80, 55], reconstruction based [3, 73, 8, 24], and vision-language pretraining based methods [57, 70, 21]. In general, the proposed method achieves best performance among these pretraining methods.

Compared to the current SOTA self-supervised model DINOv2 [55], our method achieves a 0.5% higher accuracy in IN-1K linear probing (85.0 vs 84.5) without a bunch of bells and whistles. Although SuperClass has seen more samples, it operates as a simpler classification framework and does not employ MultiCrop, Masked Image Modeling, or Contrastive learning, as DINOv2 does. Although comparing a self-supervised learning method to a (weakly) supervised learning approach is a system-level comparison, we still observe that a simple classification pretraining approach demonstrates superior performance across many classy benchmarks that shown in Table 2.

Compared to the contrastive counterparts CLIP [57], our method achieves higher linear probing top-1 accuracy on ImageNet-1K dataset with ViT-Base (80.2 vs 78.5) and ViT-Large (85.0 vs 82.7) as backbone. For a fair comparison, we further make a comparison with OpenCLIP [29] which trains a ViT-Large model with a batch size 90k based on Datacomp-1B dataset. Our method consistently outperforms OpenCLIP by a large margin (85.0 vs 83.9). In Table 2, our method surpasses OpenCLIP on IN-1K and Pets by clear margins with improvements of 1.6 and 2.2 points, while being comparable with OpenCLIP on Cars (92.6 v.s 92.7).

Further comparison with CLIP In Table 3, we compare the models trained by SuperClass with the currently widely used CLIP models, including zero-shot classification, and COCO captioning. To verify the effectiveness of the pretraining method, we adapt the standard ViT [18] structure as the visual backbone and added a classification head on top of it. We use the open-source Datacomp-1B [21] dataset and encounter 13B samples in the training process.

For a better comparison with the CLIP models, we select three types: OpenAI's CLIP ViT-L/14 trained on the internal WiT-400M data, and Laion's CLIP ViT-L/14 trained on the open-source Datacom-1B dataset. The checkpoints of these two models have been open-sourced. The checkpint of Laion's openCLIP is downloaded from Hugginface Hub[1]. For a fair comparison, we trained the ViT-L/16 model with a batch size 90k based on the our codebase and Datacomp-1B dataset.

With Lock image Tuning [77], the trained classification model also gains the ability of zero-shot classification. Our method achieves 79.7% Top-1 zero-shot accuracy on ImageNet-1k datatset which is much better than OpenAI CLIP ViT-L/14 and OpenCLIP ViT-L/14. Although maybe they are not directly comparable, this do reflect that the vision model trained by the proposed SuperClass is with strong visual perception capabilities.

Combining the frozen vision encoder with a frozen pretrained 12-layer GPT-2 decoder [58] via a trained adapter, the models are trained on COCO captions [9] and CIDEr socres are reported. We observe that the CIDEr socres of our method are slightly below OpenAI's CLIP, which may be due to the use of different datasets; OpenAI's CLIP utilizes an internal dataset, WiT-400M. However, our approach outperforms our implemented CLIP model with the same settings.

Overall, the models trained by proposed SuperClass demonstrated marginally improved accuracy in both classification capabilities and the vision & language task when compared to the contrastive pretrained CLIP models.

Large multi-modal models Many large multi-modal models integrate pre-trained vision backbones with large language models. We explore how amenable the learned representations are to interfacing with a text decoder. Following the LLaVA setup [47], we combine frozen CLIP models and SuperClass models with the pretrained Vicuna-7B [13] and perform downstream tasks. In Figure 2, we show some results of vision&language downstream tasks. The results demonstrate that SuperClass models could achieve better performance than CLIP models on the majority of datasets. It is worth mentioning that, in comparison to CLIP models, SuperClass models exhibit significantly better performance on VQAv2 [22], T-VQA [66], and MMBench [48], which pertain to OCR and fine-grained recognition tasks, respectively. In addition, the overall accuracy measurement on VizWiz [23] are not stable due to a significant portion of questions being labeled as unanswerable. To ensure the completeness of our findings, we still present the results on this dataset.

Due to space limitations, detailed numerical results are provided in the appendix. Additionally, you can find more experimental results in the appendix.

[1] Laion's CLIP https://huggingface.co/laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K.

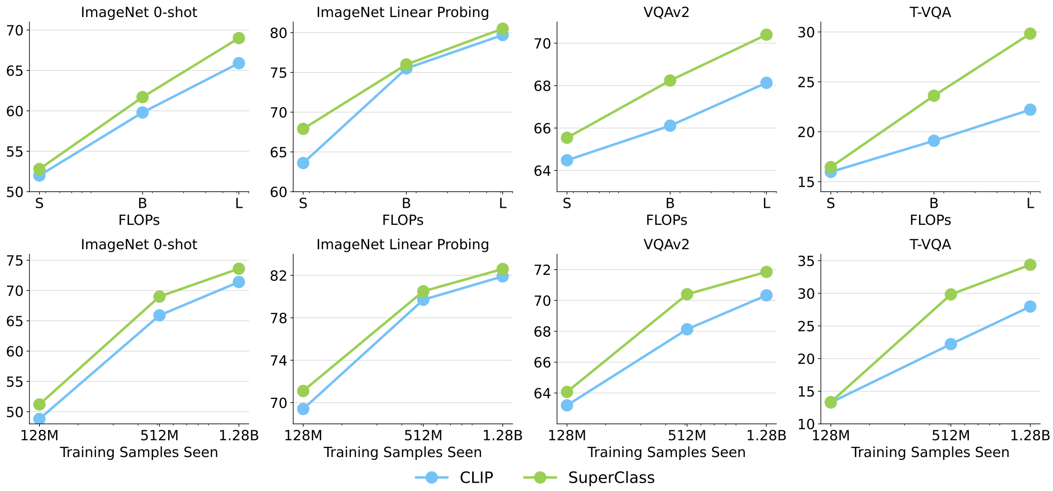

Figure 2: Zero-shot classification accuracy and linear probing accuracy on ImageNet-1k dataset (left two columns); Performance of VQAv2 and T-VQA with LLaVA training recipe (right two columns). **Top row:** We compare the performance of vision backbones—ViT-S/16, B/16, and L/16—pretrained via classification and contrastive methods with the same batch size of 16k and 512 million seen samples, focusing on their size and computational cost. SuperClass demonstrates better scaling on zero-shot classification and VQAv2, T-VQA tasks. **Bottom row:** Comparing SuperClass and CLIP, performance increases with more training examples, mirroring the effects of model scaling. All methods are trained the same batch size of 16k and ViT-L/16 as backbone.

Model scaling results In the top row of Figure 2, we showcase the performance across classification and vision & language tasks for varying model scales. For a fair comparison, both CLIP and SuperClass models undergo training with identical settings, which include a batch size of 16k and 512 million seen samples. As shown in Figure 2, with the model scaling up, we observe a corresponding enhancement in performance, whether it is for classification tasks or the downstream tasks associated with LLaVA. Generally speaking, with the same model size, models pre-trained using SuperClass exhibit superior precision compared to those trained with CLIP. SuperClass demonstrates better scaling on zero-shot classification and VQAv2, T-VQA tasks.

Data Scaling results In the bottom row of Figure 2, we showcase the performance across classification and vision & language tasks for varying seen samples. For a fair comparison, both CLIP and SuperClass models undergo training with identical settings, which include a batch size of 16k and ViT-L/16 as backbone. Figure 2 illustrates that as the number of seen samples grows, there is a noticeable improvement in performance for both classification and downstream tasks linked to LLaVA. Typically, models pre-trained with SuperClass outperform those trained with CLIP in terms of accuracy when given the same amount of seen samples. SuperClass exhibits the same or slightly better scaling behavior compared to CLIP on downstream tasks. In addition, SuperClass does not require a text encoder, it offers better efficiency in training compared to CLIP.

4.4 Ablations

To verify the rationality of the SuperClass, we conduct extensive ablation experiments that pretrain on datacomp-1B [21] and evaluate on several downstream tasks with different settings for SuperClass.

Word-level tokenizer vs. Subword-level tokenizer Table 4 presents the results of word-level tokenizer and subword-level tokenizer on serval daownstream tasks. We use the tokeinzer in openai CLIP as our subword-level tokenizer. We compare it with the word-level tokenizer used in CatLIP [51], which carefully selected approximately 40,000 "gold labels" from the datacomp-1B dataset. Aside from the tokenizer being different, all models are trained under the same settings.

For ViT-S/16, word-level tokenizer achieves better classification accuracy than subword-level tokenizer. A possible reason is that when the model capacity is limited, the filtered clean supervisory

Table 4: Word tokenizer vs. Subword tokenizer. The performance of classification and LLaVA downstream tasks with different tokenizers. SuperClass use a subword-level tokenizer to map text into category labels. All models are trained in the same settings with a batch size 16k and 512M seen samples.

		Classification		Vision & Language Downstream Tasks									
Tokenizer	ViT	LP	ZS	VQAv2	GQA	VizWiz	T-VQA	SQA	MMBench	MME	POPE	MMMU	SEED
Word	S/16	**68.4**	**53.2**	65.28	55.38	43.12	15.84	**65.83**	**49.14**	1228	80.32	**36.6**	**50.54**
Subword	S/16	67.9	52.8	**65.54**	**55.95**	**46.23**	**16.46**	65.64	48.53	**1306**	**81.02**	33.2	50.43
Word	B/16	**76.1**	61.4	67.72	57.12	**46.20**	20.98	**65.79**	**54.63**	1296	81.88	34.4	**53.38**
Subword	B/16	76.0	**61.7**	**68.34**	**57.43**	41.79	**24.41**	65.54	54.03	**1324**	**82.28**	**36.9**	52.88
Word	L/16	80.3	68.2	69.47	57.36	**51.94**	23.87	65.00	56.27	1335	83.28	35.4	53.64
Subword	L/16	**80.5**	**69.0**	**70.40**	**58.16**	51.48	**29.83**	**67.72**	**59.45**	**1373**	**84.04**	**36.3**	**53.74**

Table 5: The performance of classification and LLaVA downstream tasks with different subword-level tokenizers. All models are trained in the same settings with a batch size 16k, 512M seen samples and ViT-L/16 as Backbone.

		Classification		Vision & Language Downstream Tasks									
Tokenizer	Vocab	LP	ZS	VQAv2	GQA	VizWiz	T-VQA	SQA	MMBench	MME	POPE	MMMU	SEED
OpenaiCLIP	49,152	**80.5**	**69.0**	**70.40**	**58.16**	**51.48**	**29.83**	**67.72**	**59.45**	1373	**84.04**	**36.3**	53.74
WordPiece	32,000	**80.5**	68.5	69.95	57.76	49.07	29.33	65.99	56.18	**1375**	83.37	35.1	**54.05**
SentencePiece	32,000	80.2	67.8	69.52	57.95	49.20	28.56	65.05	57.47	1301	82.16	34.8	53.52

information may be more conducive to model convergence. However, with the increasing size of the model, subword-level tokenizer gradually outperforms the word-level tokenizer, whether in classification tasks or vision & language tasks.

Regardless of model size, on most vision & language tasks, subword-level tokenizer tends to perform better than word-level tokenizer. The reason may be that subword-level tokenizer retain a substantial amount of language-related information, although it may not be highly relevant to visual information. This makes the features of the models trained with the subword-level tokenizer more readily integrated with large language models.

Overall, using a subword-level tokenizer exhibits better scaling behavior and is more suitable for use in large multi-modal models.

Different subword-level tokenizers Table 5 presents the results on classification tasks and LLaVA downstream tasks with different subword-level tokenizers. Here, we compare the character-based byte pair encoding tokenizer [64] used in CLIP [57], the WordPiece [72] tokenizer used in BERT [16] and the SentencePiece [36] tokenizer used in LLama [69], they are all subword-level tokenizers. The tokenizer used in openai CLIP obtains best performance on the classification task and LLaVA downstream tasks. Finally, we choose the tokenizer used in CLIP [57] for the training of SuperClass models.

Classification loss Table 6 represents different classification loss on ImageNet-1k dataset. We selected several of the most commonly used multi-label classification losses for experimentation. **Softmax loss** is often used in single-label classification tasks. It is possible apply a softmax loss in a multi-label scenario through describing labels in a probabilistic way. **BCE loss** is a binary cross-entropy (BCE) loss and is often used in multi-label classification tasks as baseline. Asymmetric Loss(**ASL loss**) a improved BCE loss to address positive-negative imbalance. **Soft margin loss** is a

Table 6: The performance on classification tasks with different classification losses. All models are trained in the same settings with a batch size 16k, 512M seen samples and ViT-B/16 as Backbone.

Loss&Acc.	Softmax	BCE	ASL	SoftMargin	Two-way
Linear prob	**75.6**	73.6	73.8	73.5	74.8
Zero-shot	**60.8**	58.5	58.7	58.1	59.7

Table 7: The effect of IDF weight in the loss and removing stopwords.

	Classification		Vision & Language Downstream Tasks									
Tokenizer	LP	ZS	VQAv2	GQA	VizWiz	T-VQA	SQA	MMBench	MME	POPE	MMMU	SEED
SuperClass	**76.0**	**61.7**	**68.34**	**57.43**	41.79	**24.41**	**65.54**	54.03	1324	82.28	**36.9**	52.88
w/o IDF	75.6	60.8	68.08	57.27	47.60	23.73	65.44	**54.55**	1310	**82.58**	34.6	52.53
rm Stopwords	75.7	61.0	68.29	57.41	**47.67**	24.12	65.34	53.86	**1343**	82.50	33.8	**53.12**

margin-based loss for multi-label classification tasks. **Two-way loss** is the current state-of-the-art (SOTA) for multi-label classification tasks.

Surprisingly, the simplest softmax loss outperforms all other multi-label classification losses by a large margin. We believe that existing multi-label classification losses operate under the assumption that labels are both accurate and complete, aiming to optimize the classification margin between positive and negative classes. However, in reality, image-text data contains considerable noise, and a single text passage cannot possibly capture all the contents of an image. Consequently, certain categorical objects present in the image may not be mentioned in the associated text. In the context of image-text pretraining, how to design a better loss function remains a question worthy of exploration.

IDF as class weights Considering that the importance of each category (subword) in the vocabulary is not equal and the information they carry varies, we use IDF as class weights. The Table 7 repsents the results of with and without IDF as class weights. SuperClass without IDF experienced a noticeable decrease in accuracy on classification tasks, the change in precision on LLaVA tasks is not significant.

Removing stopwords? Stop words are commonly used in Text Mining and Natural Language Processing (NLP) to eliminate words that are so widely used that they carry very little useful information. In the previous classification methods, the stopwords are removed. The stopwords are download from NLTK [49]. However, the results in Table 7 shows that the keeping stopwords could help the vision encoder to gain better performance on classification tasks.

5 Limitation and Conclusion

We have conducted a thorough comparison of vision encoders pre-trained with contrastive and classification objectives and determined that models pre-trained with classification surpass CLIP models in both classification and vision & language tasks. Additionally, our approach does not require a text encoder, which leads to higher training efficiency than that of CLIP. Furthermore, our findings suggest that classification as a pre-training task may have beneficial scaling properties as model and data sizes increase, and we encourage future research to delve into this possibility.

While it delivers impressive results on various downstream tasks, it completely ignore word order and object relationships, which implies that we are losing important supervisory information. Addressing this will be the direction of our future research efforts.

To sum up, we have demonstrated that straightforward image classification can serve as an effective pre-training strategy for vision backbones derived from image-text data. Our aim is to stimulate subsequent studies to pay more attention to the benefits of classification as a pre-training task for vision encoders.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: A visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[4] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022.

[5] 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 International Conference on Computer Vision (ICCV)*, 2021.

[6] 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 *CVPR*, 2021.

[7] Jienneg Chen, Qihang Yu, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Vitamin: Designing scalable vision models in the vision-language era. *arXiv preprint arXiv:2404.02132*, 2024.

[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. *International Journal of Computer Vision*, 132(1):208–223, 2024.

[9] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO Captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021.

[11] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

[12] 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.

[13] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, march 2023. *URL https://lmsys. org/blog/2023-03-30-vicuna*, 3(5), 2023.

[14] 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. Ieee, 2009.

[15] Karan Desai and Justin Johnson. VirTex: Learning visual representations from textual annotations. In *CVPR*, 2021.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[17] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Shuyang Gu, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vit-b and vit-l on imagenet. *arXiv preprint arXiv:2212.06138*, 2022.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[19] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

[20] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[21] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023.

[22] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[25] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[26] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *CVPR*, pages 17980–17989, 2022.

[27] Xinyu Huang, Youcai Zhang, Jinyu Ma, Weiwei Tian, Rui Feng, Yuejie Zhang, Yaqian Li, Yandong Guo, and Lei Zhang. Tag2text: Guiding vision-language model via image tagging. *arXiv preprint arXiv:2303.05657*, 2023.

[28] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[29] 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.

[30] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021.

[31] Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 67–84. Springer, 2016.

[32] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. OCR-Free document understanding transformer. In *ECCV*, pages 498–517, 2022.

[33] Takumi Kobayashi. Two-way multi-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7476–7485, 2023.

[34] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[36] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[37] Weicheng Kuo, AJ Piergiovanni, Dahun Kim, Xiyang Luo, Ben Caine, Wei Li, Abhijit Ogale, Luowei Zhou, Andrew Dai, Zhifeng Chen, et al. Mammut: A simple architecture for joint learning for multimodal tasks. *arXiv:2303.16839*, 2023.

[38] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. *arXiv:2210.03347*, 2022.

[39] Ang Li, Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. Learning visual n-grams from web data. In *ICCV*, pages 4183–4192, 2017.

[40] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[41] Gang Li and Yang Li. Spotlight: Mobile UI understanding using vision-language models with a focus. In *ICLR*, 2023.

[42] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, pages 12888–12900, 2022.

[43] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

[44] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023.

[46] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji rong Wen. Evaluating object hallucination in large vision-language models. *ArXiv*, abs/2305.10355, 2023.

[47] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[48] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

[49] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[50] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7086–7096, 2022.

[51] Sachin Mehta, Maxwell Horton, Fartash Faghri, Mohammad Hossein Sekhavat, Mahyar Najibi, Mehrdad Farajtabar, Oncel Tuzel, and Mohammad Rastegari. Catlip: Clip-level visual recognition accuracy with 2.7 x faster pre-training on web-scale image-text data. *arXiv preprint arXiv:2404.15653*, 2024.

[52] M Minderer, A Gritsenko, A Stone, M Neumann, D Weissenborn, A Dosovitskiy, A Mahendran, A Arnab, M Dehghani, Z Shen, et al. Simple open-vocabulary object detection with vision transformers. arxiv 2022. *arXiv preprint arXiv:2205.06230*, 2, 2022.

[53] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[54] Yasuhide Mori, Hironobu Takahashi, and Ryuichi Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *First international workshop on multimedia intelligent storage and retrieval management*, volume 2. Citeseer, 1999.

[55] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[56] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[57] 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 *ICML*, 2021.

[58] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[59] Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 82–91, 2021.

[60] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[61] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *ECCV*, pages 153–170, 2020.

[62] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[63] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[64] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[65] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018.

[66] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[67] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

[68] 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. *Commun. ACM*, 59(2):64–73, 2016.

[69] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[70] Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. *Advances in Neural Information Processing Systems*, 36, 2024.

[71] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *ICLR*, 2022.

[72] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[73] 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.

[74] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022.

[75] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Trans. Machine Learning Research*, 2022.

[76] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv:2303.15343*, 2023.

[77] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022.

[78] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv preprint arXiv:2306.03514*, 2023.

[79] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

[80] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.

Table 8: The performance of classification and LLaVA downstream tasks with different seen samples. "LP" means linear probing and "ZS" means zero-shot classification, these two are tested on ImageNet-1K dataset. The results of vision&language downstream tasks are obtained by combine the frozen vision models and Vicuna-7B [13], following the settings in LLaVA [47].

		Classification		Vision & Language Downstream Tasks									
Method	Data	LP	ZS	VQAv2	GQA	VizWiz	T-VQA	SQA	MMBench	MME	POPE	MMMU	SEED
CLIP	128M	69.4	48.8	63.20	53.98	45.80	13.27	64.70	46.56	1216	78.20	35.1	47.84
SuperClass	128M	**71.1**	**51.2**	**64.07**	**54.96**	**49.21**	**13.33**	**65.44**	**49.14**	**1241**	**80.01**	**35.3**	**49.08**
CLIP	512M	79.7	65.9	68.13	57.32	44.92	22.21	64.45	51.54	1299	82.48	35.3	53.06
SuperClass	512M	**80.5**	**69.0**	**70.40**	**58.16**	**51.48**	**29.83**	**67.72**	**59.45**	**1373**	**84.04**	**36.3**	**53.74**
CLIP	1.28B	81.9	71.4	70.33	58.95	46.71	27.97	64.65	55.49	1351	83.37	35.7	55.09
SuperClass	1.28B	**82.6**	**73.6**	**71.85**	**59.09**	**51.70**	**34.37**	**65.94**	**59.70**	**1392**	**84.41**	**36.8**	**55.51**

Table 9: The performance of classification and LLaVA downstream tasks with different model sizes.

		Classification		Vision & Language Downstream Tasks									
Method	ViT	LP	ZS	VQAv2	GQA	VizWiz	T-VQA	SQA	MMBench	MME	POPE	MMMU	SEED
CLIP	S/16	63.6	52.0	64.48	55.26	44.16	15.98	**65.84**	47.33	1227	80.49	**35.8**	49.72
SuperClass	S/16	**67.9**	**52.8**	**65.54**	**55.95**	**46.23**	**16.46**	65.64	**48.53**	**1306**	**81.02**	33.2	**50.43**
CLIP	B/16	75.5	59.8	66.11	56.28	**49.17**	19.10	64.30	48.62	1289	81.47	35.9	50.76
SuperClass	B/16	**76.0**	**61.7**	**68.34**	**57.43**	41.79	**24.41**	**65.54**	**54.03**	**1324**	**82.28**	**36.9**	**52.88**
CLIP	L/16	79.7	65.9	68.13	57.32	44.92	22.21	64.45	51.54	1299	82.48	35.3	53.06
SuperClass	L/16	**80.5**	**69.0**	**70.40**	**58.16**	**51.48**	**29.83**	**67.72**	**59.45**	**1373**	**84.04**	**36.3**	**53.74**

A Appendix / supplemental material

Broader Impacts

This work presents a new approach to train vision models, which can be applied for image recognition and other vision tasks. The approach demonstrates higher efficiency than the popular one in the community, which can reduce the computational cost and the power cost for computer vision model training.

Data Scaling results In Table 8, we showcase the performance across classification and vision & language tasks for varying seen samples. For a fair comparison, both CLIP and SuperClass models undergo training with identical settings, which include a batch size of 16k and ViT-L/16 as backbone.

Figure 2 illustrates that as the number of seen samples grows, there is a noticeable improvement in performance for both classification and downstream tasks linked to LLaVA. Typically, models pre-trained with SuperClass outperform those trained with CLIP in terms of accuracy when given the same amount of seen samples. SuperClass exhibits the same or slightly better scaling behavior compared to CLIP on downstream tasks. In addition, SuperClass does not require a text encoder, it offers better efficiency in training compared to CLIP.

Model scaling results In Table 9, we showcase the performance across classification and vision & language tasks for varying model scales. For a fair comparison, both CLIP and SuperClass models undergo training with identical settings, which include a batch size of 16k and 512 million seen samples.

Table 10: Performance of frozen visual representations trained via image classification (SuperClass) and constrastively (CLIP). Linear probing and zero-shot classification are both tested on ImageNet-1k dataset. Captioning is conducted on COCO captions and CIDEr is reported in the table. The zero-shot accuracy of SuperClass is obtained after lock-image tuning [77].

Method	Backbone	Data	Seen Samples	Zero-shot	Linear Probing
CLIP	RN-50	Datacomp-1B	1.28B	60.73	70.28
SuperClass	RN-50	Datacomp-1B	1.28B	62.81	71.92
CLIP	ConvNext-tiny	Datacomp-1B	1.28B	59.94	70.35
SuperClass	ConvNext-tiny	Datacomp-1B	1.28B	62.85	72.33

Table 11: The performance of vision & language downstream tasks with different pretrained models.

Method	VQAv2	GQA	VizWiz	T-VQA	SciQA	MME	MMB	PoPE	MMMU
OpenCLIP	74.54	61.03	50.47	38.16	**67.33**	**1434**/269	60.73	85.52	35.9
MAE	63.50	54.58	50.22	11.55	54.75	1175/343	42.44	80.69	35.7
DINOv2	73.32	**61.87**	49.15	14.08	64.90	1336/297	57.90	**86.24**	35.3
SuperClass	**75.24**	60.96	**54.33**	**39.20**	66.09	1371/**322**	**63.14**	85.69	**36.0**

As shown in Figure 2, with the model scaling up, we observe a corresponding enhancement in performance, whether it is for classification tasks or the downstream tasks associated with LLaVA. Generally speaking, with the same model size, models pre-trained using SuperClass exhibit superior precision compared to those trained with CLIP. SuperClass demonstrates better scaling on zero-shot classification and VQAv2, T-VQA tasks.

Superclass with different model types To evaluate the robustness of our proposed method across different model types, we selected two representative convolution-based networks: ResNet50 and ConvNext-Tiny. We compare SuperClass against CLIP for ImageNet zero-shot (LiT) and linear probing classification, as shown in Table 10. All experiments were conducted with a batch size of 16k and 1.28B seen samples. We observe that SuperClass surpasses CLIP in all settings by a clear margin, ranging from 1.64 to 2.91. These results demonstrate that the superiority of SuperClass over CLIP is robust across different model architectures.

VLM downstream tasks with different types of pretraining models Following the LLaVA setup, we combine frozen CLIP models, self-supervised models, and SuperClass models with the pre-trained Vicuna-7B and perform downstream tasks. The experimental results in Table 11 demonstrate that the proposed method could achieve better than self-supervised ViT pre-training methods, like DINOv2, and weakly-supervised methods, like CLIP.

Comparison with other classification based pretraining models We have included the comparison with other classification-based methods, like CatLIP [51] in the subsection Word-level tokenizer vs. Subword-level tokenizer. The word-level tokenizer is used in CatLIP [51], which carefully selected approximately 40,000 "gold labels" from the datacomp-1B dataset. Aside from the tokenizer being different, all models are trained under the same settings. The results of Table 4 show that with the increasing size of the model, the subword-level tokenizer gradually outperforms the word-level tokenizer, whether in classification tasks or vision & language tasks. We also provide the results of finetuning on ImageNet-1k in Table 12. Using the same dataset Datacom-1B for training, the same backbone ViT-L/16 as backbone, the same number 13 Billion of training samples seen, the SuperClass could achieve better performance than CatLIP (87.8 vs 86.5).

Table 12: Comparison of the Fine-tuning top-1 accuracy on ImageNet-1K dataset. *number from the paper.

Method	Pretraining Data	ImageNet-1k Fine-tuning
OpenCLIP ViT-L/14	Datacomp-1B	87.4
CatLIP ViT-L/16*	Datacomp-1B	86.5
Superclass ViT-L/16	Datacomp-1B	87.8

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: refer to abstract and introduction

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Discussed in the conclusion

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification:

   Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Important details are provided in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

Justification: We use public data for model training. The code will be cleaned and made public in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: we provide all the details in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We follow the convention in prior works and report the performance number on the standard benchmarks.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: please refer to the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss in the conclusion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We used public datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.