IS A CAPTION WORTH A THOUSAND IMAGES? A STUDY ON REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The development of CLIP (Radford et al., 2021) has sparked a debate on whether adding language supervision can yield vision models with more transferable representations than traditional image-only methods. Our work studies this question through a *carefully controlled* comparison of two approaches, in terms of their ability to learn representations that generalize to downstream classification tasks. We find that when the pre-training data meets certain criteria—it is sufficiently large and contains descriptive captions with low variability—-image-only methods *do not* match CLIP's performance even when they are trained with more image data. However, contrary to what one might expect, there are practical settings in which these criteria are not met, wherein added supervision through captions is actually *detrimental*. Motivated by our findings, we devise simple data and algorithmic interventions to improve the transfer performance of CLIP-style models.

1 INTRODUCTION

Image-based contrastive learning approaches have shown promise in building models that generalize beyond the data distributions they are trained on (Wu et al., 2018; He et al., 2020; Chen et al., 2020a; Caron et al., 2020; Chen et al., 2020b; Caron et al., 2021). By leveraging large (unlabelled) data sources via self-supervised training, these models learn representations that transfer to diverse image classification tasks—more so than their supervised counterparts (Ericsson et al., 2021).

Recently, Radford et al. (2021) showed that a different approach—contrastive learning with language supervision—can yield models (CLIP) with remarkable transfer capabilities. This development has garnered significant interest in the vision and natural language processing communities alike, leading to a debate on the utility of multi-modality in visual representation learning (Zhai et al., 2022; Devillers et al., 2021; Fang et al., 2022). Our work focuses on a specific question within this debate:

Does added language supervision lead to more transferable visual representations than using images alone?

It might seem like the answer to this question is obvious. After all, CLIP utilized caption information unavailable to traditional image-based approaches and showed substantial gains over them (Radford et al., 2021). However, CLIP is drastically different from these approaches in many ways, from training data to fine-grained implementation choices, which makes it difficult to isolate the contribution of language supervision (see Section 5). Further, recent studies on CLIP's zero-shot classification and robustness properties cast doubt on whether adding language supervision is always beneficial (Fang et al., 2022). Resolving the aforementioned debate thus requires a carefully controlled comparison of the two approaches in which the *only* difference is the form of supervision.

Our contributions. We devise a methodology to assess the utility of language supervision in CLIP[1] from a visual representation learning standpoint. To do so, we recognize that CLIP pre-training and popular image-based methods share the same underlying primitive of contrastive learning. Specifically, Radford et al. (2021)'s approach is strikingly similar to SimCLR (Chen et al., 2020a). The only irreducible difference between them is whether supervision is provided to the model via image augmentations or image-caption matching (see Figure 1)—which is precisely the

[1]We use CLIP to refer to models trained with Radford et al. (2021)'s approach, not their pre-trained model.

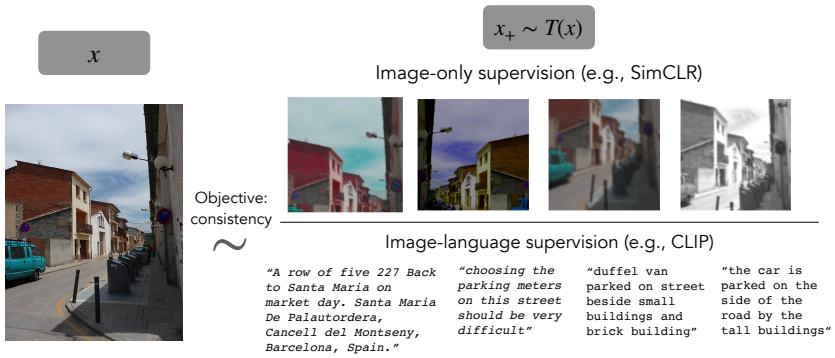


Figure 1: A conceptual view of contrastive image-only and image-language pre-training. Both methods rely on the same self-supervised objective: aligning the representations of positive examples (x, x_+) while distinguishing them from negative ones (x_n). The transformation $T(\cdot)$ used to obtain $x_+ \sim T(x)$ (augmented image or caption) encodes the equivalences the model must satisfy.

quantity we want to study. Thus, we can disentangle the effect of language supervision on visual representations by comparing matched versions of SimCLR and CLIP (trained from scratch). Our focus, in particular, is on how well the learned representations transfer to varied image classification tasks. We find that the picture is nuanced and depends on three properties of the pre-training data:

1. When the *scale* of the dataset is sufficiently large, CLIP's visual representations indeed transfer better than their matched image-only SimCLR counterparts. In fact, this gap is not bridged by training SimCLR with more (image) data, suggesting that a caption can be worth more than *any* number of images. However, in the low-data regime, language supervision actually hurts model performance both in and out-of-distribution.
2. The *descriptiveness* (Kreiss et al., 2021) of captions—i.e., the extent to which they refer to what is contained in an image—directly determines how well CLIP models transfer. In fact, we find that a single descriptive image-caption pair (e.g., from COCO (Lin et al., 2014)) is worth five less descriptive, uncurated captions (e.g., from YFCC (Thomee et al., 2016)).
3. The *variability* of captions (e.g. stylistic or lexical) within a dataset can impair CLIP's performance. We find that a modification to standard CLIP training—performing text augmentations by sampling from a pool of captions for each image—can alleviate this drop.

These properties have inter-twined effects on CLIP's performance: e.g., dataset scale can, to some extent, compensate for less-descriptive and/or varied captions. Guided by our findings, we devise simple datasets interventions that can lead to more-transferrable CLIP models: (i) filtering out low-quality captions with a text-based classifier, and (ii) applying data augmentation to captions by paraphrasing them using pre-trained language models.

2 An apples-to-apples comparison

Prior works have studied image-only and image-language pre-training methods in isolation (Wu et al., 2018; He et al., 2020; Chen et al., 2020a; Caron et al., 2020; Chen et al., 2020b;b; Chen & He, 2021; Caron et al., 2021; Radford et al., 2021) and side-by-side (Desai & Johnson, 2021; Devillers et al., 2021; Fang et al., 2022). Yet, they provide incomplete (and often contradictory) answers to our motivating question of the value of language supervision relative to using images alone (Section 5). Crucially, this is due to various confounders such as: (i) bespoke algorithmic optimizations within the two methods, and (ii) differing pre-training datasets. In this section, we outline a series of steps that we take to mitigate these confounders and compare the two methods on equal footing.

2.1 Finding common ground

Our approach for studying the value of language supervision is guided by the following insight: CLIP pre-training is strikingly similar to the popular image-only SimCLR method (Chen et al.,

2020a)[2]. Both methods rely on the same algorithmic primitive of *contrastive learning*, which we illustrate in Figure 1. Specifically, the (CLIP/SimCLR) model is trained cross-entropy based objective, which for a given pair (x, x_+) of positive examples with with associated negatives $\mathcal{N}$ is:

$$\ell = -\log \frac{\exp(\text{sim}(z, z_+)/\tau)}{\sum_{n \in \mathcal{N} \cup \{z_+\}} \exp(\text{sim}(z, z_n)/\tau)}, \quad \text{where } z = g(\phi(x)) \text{ and } z_{+/n} = g'(\phi'(x_{+/n})), \quad (1)$$

sim is cosine similarity, ϕ/ϕ' are encoders, and g/g' are projection heads. Positive examples x_+ are obtained through a transformation of the image x, i.e., $x_+ \sim T(x)$—such as image augmentations (e.g., rotations or crops) in SimCLR and captions in CLIP. Observe that this difference in $T(\cdot)$ between CLIP and SimCLR corresponds exactly to whether the model is trained with language, which is the quantity we want to study. Thus, to isolate the role of added language supervision, we can compare the downstream performance of matched CLIP and SimCLR models. To this end, we must take some steps to make their implementatations consistent:

- *Datasets:* Typically, CLIP and SimCLR are trained on different datasets, as the former requires image-caption pairs, while the latter can leverage any image data. To control for the effect of the data distribution, we pre-train both models *from scratch* on the same data.
- *Architecture:* We use the ResNet-50 (He et al., 2016) architecture as the image encoder for both methods, and a Transformer (Vaswani et al., 2017) as the text encoder in CLIP. We also extensively tune hyperparameters for both methods (Appendix A.3).
- *Augmentations:* Both methods apply data augmentations to the image x itself at each training step. However, the augmentations used in SimCLR (`resize`, `crop`, `flip`, `jitter`, `blur`, `grayscale`) are far more sophisticated than those in CLIP (`resize` and `crop`). We remove this confounder by using SimCLR augmentations unless otherwise specified.
- *Transformation stochasticity:* The two methods differ in how they obtain x_+, not just due to the choice of $T(x)$ but also the generative process itself. In SimCLR , x_+ is a new random draw from $T(x)$ in every batch, while for CLIP, it is a single fixed caption. Perfectly matching them requires training CLIP by sampling a fresh caption x_+ for each image at each iteration. We will refer to this stochastic version of CLIP as CLIP_S.

Mismatches. Despite our efforts to match CLIP with SimCLR, some inconsistencies remain–partly due to their differing input modalities. In particular, CLIP (and CLIP_S):

(i) Processes $T(x)$ using a text transformer rather than SimCLR's ResNet-50.

(ii) Does not share weights between the encoders processing x and $T(x)$ because they correspond to different modalities, unlike SimCLR.

(iii) Uses a linear projection head g/g' instead of SimCLR's MLP, which we allow as Radford et al. (2021) showed that this choice does not affect CLIP's performance.

(iv) Only uses other examples in the batch from the *same* modality as negatives. Thus CLIP has half the number of negatives compared to SimCLR, which also uses transformed versions of other examples in the batch (i.e. both $\hat{x}$ and $\hat{x}_+$) as negatives.

We now assess how the representations learned by our matched CLIP and SimCLR models compare. In particular, we measure how well their representations transfer to the downstream tasks from Kornblith et al. (2019). Akin to (Radford et al., 2021), we focus on the *fixed-feature* setting, where we freeze the weights of a given model and then train a linear probe using task data (see Appendix A).

2.2 A CASE STUDY

We begin by comparing CLIP and SimCLR models trained on the MS-COCO dataset (Lin et al., 2014) (henceforth referred to as COCO), which contains ∼120K images with multi-object labels. Each image has five human-provided captions, collected post-hoc by Chen et al. (2015) using Mechanical Turk. Annotators were given detailed instructions on how to caption an image such as to describe *only* the important parts of the image and not to use proper names. We use COCO as our

[2]Other image-based methods (He et al., 2020; Chen et al., 2020b; Chen & He, 2021; Caron et al., 2021) have optimizations that are not present in CLIP.

Table 1: Linear probe accuracy for COCO pre-trained models in-distribution and on the transfer suite from Kornblith et al. (2019) (μ_{Tx} denotes average transfer accuracy).

	COCO	Aircraft	Birdsnap	Ctech101	Ctech256	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food-101	Pets	SUN937	μ_{Tx}
Supervised	**90.6**	31.6	11.8	65.8	53.7	21.7	74.8	46.7	55.9	63.4	47.1	45.9	44.5	47.2 $\pm$ 0.2
SimCLR	89.0	40.6	18.5	71.5	58.6	31.5	82.1	57.3	61.7	77.4	58.7	57.3	51.9	56.0 $\pm$ 0.2
CLIP	88.4	41.4	17.6	73.2	60.4	35.8	83.6	60.8	65.7	80.5	60.9	57.0	50.8	57.5 $\pm$ 0.1
CLIP_S (5)	89.8	**46.4**	**20.0**	**78.4**	**65.6**	**41.5**	**84.6**	**62.5**	**66.7**	**83.9**	**65.3**	**61.2**	**54.9**	**61.3 $\pm$ 0.2**

starting point for two reasons. First, we can assess the utility of language supervision in the ideal setting where the captions are of fairly high quality due to the careful curation process. Second, we can approximate CLIP_S[3] by sampling from the available set of five captions per image.

Captions (often) help on COCO. In Table 1, we compare various COCO pre-trained models (supervised, SimCLR, CLIP/CLIP_S) in terms of the accuracy of a linear probe on: (i) COCO classification (in distribution), and (ii) transfer tasks. Note that to contrast image-only and image-language supervision, the "right" comparison is between SimCLR and CLIP_S: they are matched (to the best of our abilities) in terms of dataset, architecture, augmentations and stochasticity. We find that:

- As expected, the supervised model outperforms self-supervised ones in distribution, but the trend flips when we consider transfer tasks.
- Our matched stochastic variant of CLIP, CLIP_S, outperforms SimCLR both in- ($\sim 1\%$) and out-of-distribution ($\sim 6\%$).
- Vanilla CLIP (which does not contain stochastic transformations like SimCLR/CLIP_S) still performs better than SimCLR on downstream transfer tasks: albeit by a smaller margin ($\sim 2\%$). Further, CLIP is worse in-distribution than SimCLR. In Section 3.3, we take a closer look at exactly *why* stochasticity in CLIP has such a drastic impact on performance.
- Finally, matching image augmentations (applied to x) turns out to be crucial to assessing the merit of added language supervision. Using typical CLIP augmentations (only `resize` and `crop`) during training lowers CLIP's average transfer accuracy by 10% (Appendix Table 4) and SimCLR's by 50%. This highlights the importance of controlling for potential confounders (Section 2.1) to fairly evaluate the utility of added language supervision.

3 THE IMPACT OF PRE-TRAINING DATA

Our analysis of COCO shows that language supervision can be beneficial over using images alone. That being said, the datasets that CLIP is typically trained on differ, both in scale and quality, from COCO. For instance, COCO captions were collected post-hoc under controlled settings, which is markedly different from the automatic scraping procedure used to gather data at scale. Thus, we shift our focus to two frequently-used (Ilharco et al., 2021) CLIP training datasets:

ConceptualCaptions (Sharma et al., 2018) (CC) contains $\sim$3.3M images harvested from web, with their ALT-text attributes as captions. The data was filtered for text quality—e.g., well-formed captions that mention at least one object found via the Google Cloud Vision API. Furthermore, all proper nouns in the captions were hypernymized (e.g., "Justin Timberlake" becomes "pop artist").

Yahoo Flickr Creative Commons (Thomee et al., 2016) (YFCC): This dataset has $\sim$ 99.2M images from Flickr, along with their posted titles as captions with no post-processing.

[3]We henceforth overload notation and use CLIP_S to denote: (i) the idealized stochastic version of CLIP, which samples from infinite captions per image, and (ii) our approximation of it with a finite set of captions.

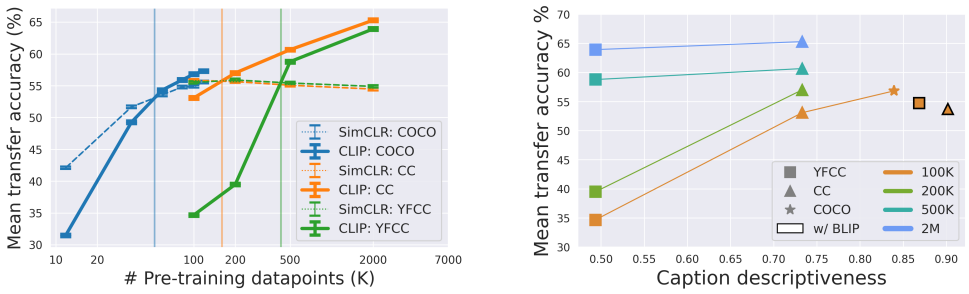


Figure 2: (*left*) Effect of pre-training dataset size on models' performance. While added language supervision consistently improves transfer accuracy in the medium to large data regime over using images alone, it is actually detrimental on small corpora. (Due to compute constraints, we train models for fewer epochs (100 instead of 200) on datasets of size 2M.) (*right*) Relationship between CLIP's transfer performance and the average *descriptiveness* of dataset captions, measured using a pre-trained BLIP caption-scoring model (Li et al., 2022). Additionally, we also visualize the performance of CLIP models trained on 100K YFCC/CC samples using "more descriptive" BLIP-generated captions: cf. the *orange square/triangle outlined in black*.

Do captions still help? We start by comparing the transfer performance of CLIP and SimCLR on 100K subsets of COCO/CC/YFCC in Figure 2(*left*). We observe that SimCLR's transfer capabilities do not vary much across pre-training datasets, while CLIP's performance is highly sensitive to them. With 100K samples from CC/YFCC, using CLIP is *worse* than image-only pre-training via SimCLR—unlike what we see for COCO.

The sensitivity of CLIP to pre-training data. Inspecting dataset samples (Figure 3) yields a possible explanation for this sensitivity. The three datasets differ not just in scale and image diversity, but also the extent to which captions: (i) *describe* visually salient aspects of the image, and (ii) *vary* across images (e.g., in style and wording). For instance, COCO captions are homogenous and descriptive, while YFCC ones vary and are often complementary to the image. We now study the effect these dataset properties—scale, descriptiveness, and variability—have on CLIP's performance.

3.1 Scale matters

A major appeal of contrastive learning methods is that they can leverage the vast amounts of unlabeled data available on the Internet. Thus, it is natural to ask how different forms of contrastive supervision benefit from added pre-training data. We may expect image-only methods to perform worse for smaller datasets as they are less likely to encounter (augmented) images which are similar. We might further expect image-language models to perform more favorably in this setting since they receive richer supervision. To test whether this is the case, we compare CLIP and SimCLR models trained on datasets of varying sizes: 10-100K samples for COCO, and 100K-2M for CC/YFCC.

Our results in Figure 2(*left*) deviate from our earlier expectations. First, beyond a certain point, SimCLR's transfer performance improves only marginally with additional data. While surprising, similar effects have been noted previously (Tian et al., 2021; Cole et al., 2022), especially when the data is uncurated (e.g., YFCC) (Tian et al., 2021). Second, in the low-data regime ($<$50K/200K/500K for COCO/CC/YFCC), training with language actually hurts the models' transfer performance. In fact, (data) scale seems to be essential to benefit from language supervision. With sufficient data, CLIP outperforms SimCLR on all three datasets. This gap remains even if we train SimCLR with extra data, indicating that captions can be worth more than *any* number of images.

3.2 The importance of descriptive captions

Prior work in linguistics and accessibility has drawn a distinction between image "descriptions" and "captions" (Berger & Dibb, 2003; Chandler, 2007; Hodosh et al., 2013; Bernardi et al., 2016; van Miltenburg, 2020; Kreiss et al., 2021; Dognin et al., 2022; Hutchinson et al., 2022). In particular, Bernardi et al. (2016) define descriptions as texts that "verbalize what can be seen in the image,

Figure 3: Random samples from the COCO, CC and YFCC datasets (also see Appendix Figure 6).

i.e., they refer to the objects, actions, and attributes depicted, mention the scene type, etc.". In contrast, Panofsky (1939) suggest that a typical caption "provides personal, cultural, or historical context for the image." This line of work suggests that COCO captions are more descriptive due to the decontextualization of the image and strict instructions provided to the annotators during the caption generation process (Kreiss et al., 2021). In contrast, Flickr captions (e.g., in CC/YFCC) tend to contain information that is complementary to the image Alikhani et al. (2020) since people tend not to not restate what can already be observed in the photographs they post (Hodosh et al., 2013).

Now to perform well on downstream classification tasks, we ideally want model representations that encode salient image objects. Recall that in contrastively-trained models, the learned representations are determined by the transformation $T(x)$ (captions for CLIP). This suggests a hypothesis: pre-training CLIP with descriptive captions will yield more transferrable (vision) representations.

To test this, we need to quantify the descriptiveness of a caption. Since doing so precisely is infeasible, we approximate descriptiveness using a pre-trained caption-scoring model. Specifically, we leverage the BLIP model (Li et al., 2022) which has shown state-of-the-art performance on image-based text retrieval. We then measure the average score assigned by BLIP to dataset captions matching their corresponding images—see Figure 2(*right*). As expected based on our earlier subjective assessment as well as prior work (Hodosh et al., 2013; Kreiss et al., 2021), we indeed find that the caption descriptiveness of COCO $>$ CC $>$ YFCC (see Appendix B.1 for a discussion of relevant vs. noisy captions.) Furthermore, we see that the descriptiveness of captions in the pre-training data directly correlates with CLIP's transfer performance. In fact, a CLIP model trained on 100K descriptive image-caption pairs from COCO attains performance comparable to one trained on 2x and 5x more samples from CC and YFCC respectively.

To further corroborate our hypothesis, we train CLIP CC and YFCC with "more descriptive" captions by re-captioning the images using BLIP (Li et al., 2022). Indeed, we find that CLIP trained on 100K CC/YFCC samples with BLIP captions no longer performs worse than its COCO counterpart (see Figure 2(*right*)). This indicates that CLIP's sensitivity to the pre-training corpus is not just an artifact of differing image distributions, but due to the presence (or absence) of descriptive captions.

3.3 THE EFFECT OF INTRA-DATASET VARIATIONS IN CAPTIONS

Image captions (Figure 1) seem to vary in how they describe an object (e.g., "duffel van" or "car") and the parts of the image they focus on (e.g., discussing the "street" or "brick"). We now study how these lexical and focus variations in captions CLIP's ability to learn meaningful representations.

A simple setting. As a starting point, we investigate this effect on the COCO dataset using *synthetic* captions—constructed using the available multi-object labels—whereby we can precisely control the intra-dataset captions variations. In an attempt to simulate the variations we observe in Figure 1, we design the captions to (not) be: (i) *consistent:* use a fixed term or random synonyms to describe an object across the dataset (lexical variations); and (ii) *complete:* mention all or a random subset of image objects (focus variations). (See Appendix A.6 for details and Appendix Figure 8 for examples.) Surprisingly, we find that a CLIP model trained with complete and consistent synthetic COCO captions *outperforms* a model trained on human-written captions (cf. row 1 in Figure 4(*left*) to row 3 in Table 1). However, dropping these two conditions causes the transfer performance of the model to drop significantly (cf. rows 1, 2, and 4 in Figure 4(*left*)). These findings suggest that variability in dataset captions can have an adverse effect on the resulting CLIP models.

Consistency	Completeness	Model	COCO	μ_{Tx}
✓	✓	CLIP	88.8	59.2 ± 0.1
✓	✗	CLIP	88.4	57.7 ± 0.2
✓	✗	CLIP_S	89.1	59.3 ± 0.2
✗	✗	CLIP	88.4	56.6 ± 0.2
✗	✗	CLIP_S	89.3	58.9 ± 0.2

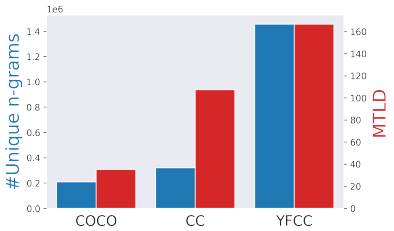


Figure 4: (*left*) CLIP trained on synthetic COCO captions (constructed using multi-object labels): effect of caption variability due to inconsistency (use a single term to describe a given object) and incompleteness (describes all image objects) on transfer performance. (*right*) Quantifying the variability of dataset captions based on the number of unique n-grams (n=1,2,3) (Li et al., 2015; Fung et al., 2020) and the MTLD score (McCarthy & Jarvis, 2010) in practical image-caption datasets.

The effect of stochasticity. We now revisit our stochastic CLIP variant, CLIP_S, in this simple setting. Intuitively, we might expect that sampling from a set of diverse captions per image—which cover possibile lexical and stylistic variations—during training might alleviate the adverse effects of caption variability. Indeed, we find that for synthetic COCO captions, CLIP_S is not as affected by caption inconsistency and/or incompleteness. The $\sim$2% improvement of CLIP_S over CLIP here mirrors the 3.6% gain seen for human-provided captions (cf. Table 1). These findings suggest that one of the reasons why stochasticity significantly boosts CLIP's performance is its role in (caption) variance reduction. We also find that CLIP_S transfers 2% better when trained on human-provided captions as opposed to synthetic ones (unlike CLIP). This indicates that human-written captions do contain useful information that is not present in object labels alone. However, extracting this signal is not straightforward, and may require incorporating multiple captions into CLIP training.

Datasets in practice. We now attempt to characterize caption variability in real-world datasets. Inspired by prior work in natural language processing and linguistics (Appendix A.8), for a set of dataset captions, we measure: (i) the total number of unique n-grams (N=1-3) (Li et al., 2015; Fung et al., 2020) and (ii) measure of textual lexical diversity (MTLD) (McCarthy & Jarvis, 2010). Along both these axes of variability, we see that COCO $<$ CC $<$ YFCC (Figure 4(*right*)). Thus, aside from the lower descriptiveness of YFCC (and to a lesser extent CC) captions, their variability could be the reason why the resulting CLIP models have worse transfer performance (Figure 2(*left*)). This also explains why scale is essential to benefit from language supervision on CC and YFCC. After all, CLIP would need to be trained on more captions to even encounter the same words twice.

How many captions are enough? We saw above that "text data augmentations" via CLIP_S could reduce the adverse impacts of caption variability. We now analyze how this effect scales with the number of available captions per image on the CC and YFCC datasets. Here, we use the BLIP captioning model to generate multiple captions per image via nucleus sampling (Holtzman et al., 2020). This procedure is intended to serve as a proxy for the manual caption annotation or automated scraping procedures that might be used for data collection in practice. We observe in Figure 5(*left*), that CLIP_S improves as the number of available captions per image increases (plateauing around 10). However, scaling up the overall number of image-caption pairs appears to be far more effective than incorporating more captions per image (at least those obtained via BLIP) from the perspective of improving transfer performance (see Figure 5(*right*)). Note that the exact trade-offs and their costs are context dependent and vary based on the exact procedure used for caption collection.

4 Making Existing Captions Work

So far, we identified three properties of the pre-training data that influence CLIP's transfer performance: (i) scale, (ii) caption descriptiveness, and (ii) caption variability. Our analysis shows that one way to improve CLIP's performance, especially on uncurated data sources, is to simply pre-train with more data. Alternatively, for a fixed data scale, we may be able to obtain better CLIP models if we improve *what* captions describe and *how* they describe an image. We now focus on the latter and put forth simple dataset interventions to improve transfer capabilities of CLIP-style models.

Method	N_C	Source	CC (100K)	YFCC (100K)
SimCLR	0	-	55.9 ± 0.2	55.5 ± 0.2
CLIP	1	Human	53.1 ± 0.2	34.7 ± 0.2
CLIP	1	BLIP	53.7 ± 0.2	54.8 ± 0.2
CLIP_S	2	BLIP	56.1 ± 0.2	56.9 ± 0.2
CLIP_S	5	BLIP	57.8 ± 0.2	58.8 ± 0.2
CLIP_S	10	BLIP	-	59.1 ± 0.2

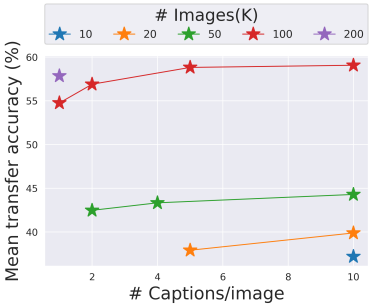


Figure 5: A closer look at CLIP_S. (*left*) Sensitivity of its transfer performance to the number of captions per image (N_C) used during training. For CC and YFCC, we use the BLIP captioning model to generate multiple diverse captions per image. Note: due to compute constraints, we run CC pre-training for a subset of N_C values. (*right*) Performance trade-offs between pre-training using more image-caption pairs vs. more captions per image on the YFCC dataset.

Data pre-processing: Given the importance of caption descriptiveness, we might consider pre-processing scraped data to select for samples with this property. The CC data collection procedure (Sharma et al., 2018) partially demonstrates the effectiveness of this approach, as pre-training CLIP on CC samples leads to better transfer performance than a comparable number of "raw" YFCC ones. However, due to its reliance on the Google Vision API, this procedure can be quite expensive, with costs scaling with the size of the scraped data. Recent works have taken a different approach, using pre-trained image-language models (like CLIP) to filter data (Schuhmann et al., 2021). However, since we are interested in building such models in the first place, we avoid taking this route.

Instead, we focus on understanding how far we can get by simply discarding low quality captions, agnostic to the images. We take inspiration from the filtering pipelines used to build large language models (Brown et al., 2020). Here, raw Internet data is cleaned by selecting samples that are "similar" to known high-quality datasets (e.g., Wikipedia). Taking a similar approach, we train a linear classifier on a bag-of-n-grams sentence embeddings (Joulin et al., 2017) to distinguish validation set CC/YFCC captions from COCO ones. This classifier is then used to filter CC/YFCC, only retaining samples that are predicted as being COCO-like. This simple procedure does end up selecting for captions that are more focused on objects and their descriptions, as opposed to describing contextual properties such as dates or urls—see Appendix A.9. For a given pre-training data budget, we see moderate gains ($\sim 2\%$) from using this heuristic to filter datasets—see Table 2 (*left*).

Mitigating caption variability: As we saw in Section 3.3, models trained with CLIP_S are less impacted by caption variability. However, typical image-captioning datasets (such as CC and YFCC) only have one caption per image. We thus devise a methodology to augment these captions by leveraging recent open-source large language models (Wang & Komatsuzaki, 2021). Concretely, we provide GPT-J with 4 (caption, paraphrase) pairs as *in-context* (Brown et al., 2020) examples. We then prompt it to paraphrase a given target caption. By sampling from GPT-J, we can obtain multiple (in our case, five) paraphrases for every such caption (examples in Appendix Figure 10). In Table 2 (*right*), we see that feeding these captions into CLIP_S results in a considerable performance boost over CLIP (trained with a single caption/image). For instance, for COCO, CLIP_S trained on our generated captions bridges more than half of the performance gap between vanilla CLIP and CLIP_S trained with five human-provided captions.

5 Related Work

Representation learning. Building models with *general* representations that transfer to downstream tasks has been a long-standing goal in ML (Donahue et al., 2014; Razavian et al., 2014; Chatfield et al., 2014; Agrawal et al., 2014; Yosinski et al., 2014). Our work is in line with prior studies aimed at characterizing the effect of design choices made during training (Azizpour et al., 2015; Huh et al., 2016; Chu et al., 2016; Kornblith et al., 2019; Zhai et al., 2019; Locatello et al., 2020), e.g. model architecture, datasets and loss functions, on learned representations.

Table 2: Improving CLIP's transfer performance through simple datasets interventions. (*left*) Applying a simple bag-of-words classifier to identify data subsets with "high quality" captions (results with "descriptive" BLIP captions shown for comparison). (*right*) Using in-context learning with GPT-J to obtain (five) diverse captions for images (via paraphrasing) and then training CLIP_S.

Dataset	Method	Preproc.	μ_{Tx}
CC (100K)	SimCLR	-	55.9 ± 0.3
	CLIP	-	53.1 ± 0.2
	CLIP	Filter	54.2 ± 0.2
YFCC (500K)	SimCLR	-	55.4 ± 0.2
	CLIP	-	58.8 ± 0.2
	CLIP	BLIP	61.8 ± 0.2
	CLIP	Filter	60.4 ± 0.2

Dataset	Method	Caption	μ_{Tx}
COCO (120K)	SimCLR	-	56.0 ± 0.2
	CLIP	Human	57.5 ± 0.1
	CLIP_S	Human	61.3 ± 0.2
	CLIP_S	GPT-J	58.9 ± 0.3
CC (200K)	SimCLR	-	55.3 ± 0.2
	CLIP	Human	57.0 ± 0.3
	CLIP_S	GPT-J	58.8 ± 0.3

The utility of language in vision. There is a long line of work on leveraging language to improve vision models (Quattoni et al., 2007; Srivastava & Salakhutdinov, 2012; Frome et al., 2013; Baltrušaitis et al., 2018; Guo et al., 2019). Recent studies have sought to investigate how integral language is to the performance of such multi-modal models. Fang et al. (2022) study a different property of CLIP—zero-shot robustness, rather than transfer learning—and show that it is comparable to that of a supervised classifier trained on the same YFCC images. Therefore, they conclude that data distribution is more important than language supervision. In concurrent work, (Nguyen et al., 2022) study the sensitivity of CLIP's zero-shot robustness to the pre-training dataset. However, unlike our work, they do not: (i) contrast CLIP against image-only methods trained on the same corpora, and (ii) attempt to explain what properties of the data are responsible for CLIP's sensitivity. Ruan et al. (2022) argue theoretically that the robustness of linear probes on CLIP's representations stems from pretraining with a large and diverse set of images and domain-agnostic augmentations $T(x)$. Most similar to our work are the studies by (Desai & Johnson, 2021) and Devillers et al. (2021), which study the role of language supervision on transfer performance in the context of VirTex (a CLIP precursor) and CLIP respectively. Notably, the two works draw opposite conclusions as to the utility of language compared to purely image-based approaches. This difference stems from the fact that neither of the works attempt to directly control for algorithmic, architectural, and data-related confounders. Our work performs a substantially more controlled study on the effect of language supervision, allowing us to make more direct claims than these works.

6 Discussion

Our work takes a step towards resolving the debate as to whether multi-modality, and language in particular, can improve visual representation learning. A comparison of CLIP with a matched image-only SimCLR model reveals that neither form of supervision (using images alone or coupled with language) is strictly better than the other. Indeed, there are practical regimes where CLIP's performance cannot be matched using SimCLR with *any* amount of image data and others where language supervision is harmful. This is a direct consequence of CLIP's sensitivity to its pre-training data, especially its scale, descriptiveness, and variability of the captions. Through our analysis, we also discovered algorithmic improvements (CLIP_S) and dataset modifications (filtering and augmenting captions) to better take advantage of language supervision.

Limitations. Our exploration allows us to quantify the utility of language supervision (over using images alone) in a specific setting: transfer learning via probing on certain object recognition tasks (Kornblith et al., 2019). We view expanding the scope of our analysis as a direction for future work. Further, despite the significant steps we took to control the differences between CLIP and SimCLR, there are still some inconsistencies that have not been accounted for (discussed in Section 2). Nevertheless, the differences between our and previous results (e.g, Desai & Johnson, 2021; Devillers et al., 2021) suggest that we successfully pinned down some crucial confounders (architecture, augmentations, stochasticity, datasets, hyperparameters).

ETHICS STATEMENT

Below, we discuss certain ethical concerns pertaining to our work:

- Although we rely on existing open source vision/multi-modal datasets for our analysis, prior work has raised concerns about some of these (or other similarly-sourced ones) being biased (Stock & Cisse, 2017; Yang et al., 2020; Birhane et al., 2021; Paullada et al., 2021) and violating privacy (Prabhu & Birhane, 2020; Yang et al., 2022).
- Our focus is on understanding the extent to which CLIP's representations are influenced by what the captions they are trained on describe. However, we sidestep whether or not this is always desirable. After all, recent studies (Birhane et al., 2021) show that vision-linguistic datasets have various biases and stereotypes, which we might not want our models to learn.
- In Section 4, we use large language models (in particular, GPT-J) to augment dataset captions via in-context learning. These models however are known to have their own limitations that might percolate into the generated captions.

REPRODUCIBILITY STATEMENT

Datasets: All the pre-training/transfer learning datasets we use are open-source. In the supplementary material, we include certain variants of the COCO/CC/YFCC datasets we created as CSV files: namely synthetic COCO captions, filtered CC/YFCC samples, and GPT-J paraphrased captions.

Code and hyperparameters: We discuss implementation details including hyperparameter settings in Appendix A. We also include the code for training models in the supplementary material.

REFERENCES

P. Agrawal, R. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *European Conference on Computer Vision (ECCV)*, 2014.

M. Alikhani, P. Sharma, S. Li, R. Soricut, and M. Stone. Clue: Cross-modal coherence modeling for caption generation. *arXiv preprint arXiv:2005.00908*, 2020.

H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability for a generic convnet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

M. M. Baese-Berk, Shiloh Drake, K. Foster, D. Lee, C. Staggs, and J. Wright. Lexical diversity, lexical sophistication, and predictability for speech in multiple listening conditions. *Frontiers in psychology*, 2021.

T. Baltrušaitis, C. Ahuja, and L. Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

J. Berger and M. Dibb. *Ways of seeing*. 2003.

R. Bernardi, R. Cakici, D. Elliott, A. Erdem, E. Erdem, N. Ikizler-Cinbis, F. Keller, A. Muscat, and B. Plank. Automatic description generation from images: A survey of models, datasets, and evaluation measures. *Journal of Artificial Intelligence Research*, 2016.

A. Birhane, V. U. Prabhu, and E. Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*, 2021.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

M. Caron, H. Touvron, I. Misra, H. J'egou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

D. Chandler. *Semiotics: the basics*. Routledge, 2007.

K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020a.

T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b.

X. Chen and K. He. Exploring simple siamese representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell. Best practices for fine-tuning visual classifiers to new domains. In *European Conference on Computer Vision (ECCV)*, 2016.

Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. When does contrastive visual representation learning work? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

K. Desai and J. Johnson. Virtex: Learning visual representations from textual annotations. In *Computer Vision and Pattern Recognition (CVPR)*, 2021.

B. Devillers, B. Choksi, R. Bielawski, and R. VanRullen. Does language help generalization in vision models? In *Computational Natural Language Learning*, 2021.

P. Dognin, I. Melnyk, Y. Mroueh, I. Padhi, M. Rigotti, J. Ross, Y. Schiff, R. A. Young, and B. Belgodere. Image captioning as an assistive technology: Lessons learned from vizwiz 2020 challenge. *Journal of Artificial Intelligence Research*, 2022.

J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2014.

L. Ericsson, H. Gouk, and T. M. Hospedales. How well do self-supervised models transfer? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

W. Falcon and the PyTorch Lightning team. Pytorch lightning, 2019. URL "`https://github.com/Lightning-AI/lightning`".

A. Fang, G. Ilharco, M. Wortsman, Y. Wan, V. Shankar, A. Dave, and L. Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). *arXiv preprint arXiv:2205.01397*, 2022.

A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

P. Fung, Y. Chen, Z. Lin, and A. Madotto. Deeper conversational ai. In *Advances in Neural Information Processing Systems (NeurIPS) Tutorial*, 2020.

W. Guo, J. Wang, and S. Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 2019.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research (JAIR)*, 2013.

A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2020.

M. Huh, P. Agrawal, and A. A. Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

B. Hutchinson, J. Baldridge, and V. Prabhakaran. Underspecification in scene description-to-depiction tasks. *arXiv preprint arXiv:2210.05815*, 2022.

G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt. Openclip, 2021. URL `https://doi.org/10.5281/zenodo.5143773`.

S. Jarvis. Capturing the diversity in lexical diversity. *Language Learning*, 2013.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *European Association for Computational Linguistics (EACL)*, 2017.

S. Kornblith, J. Shlens, and Q. V. Le. Do better imagenet models transfer better? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

E. Kreiss, N. D. Goodman, and C. Potts. Concadia: Tackling image accessibility with descriptive texts and context. *arXiv preprint arXiv:2104.08376*, 2021.

R. Krishna, Y. Zhu, O. Groth, J. Johnson, Hata K, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 2017.

J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. *North American Association for Computational Linguistics (NAACL)*, 2015.

J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022.

T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll'ar, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

F. Locatello, S. Bauer, M. Lucic, G. R"atsch, S. Gelly, B. Sch"olkopf, and O. Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation. *Journal of Machine Learning Research (JMLR)*, 2020.

P. McCarthy and S. Jarvis. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 2010.

T. Nguyen, G. Ilharco, M. Wortsman, S. Oh, and L. Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

E. Panofsky. Studies in iconology: Humanistic themes in the study of art history. 1939.

A. Paullada, I. D. Raji, E. M. Bender, E. Denton, and A. Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2021.

V. U. Prabhu and A. Birhane. Large image datasets: A pyrrhic win for computer vision? *arXiv preprint arXiv:2006.16923*, 2020.

A. Quattoni, M. Collins, and T. Darrell. Learning visual representations using images with captions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014.

Y. Ruan, Y. Dubois, and C. J. Maddison. Optimal representations for covariate shift. In *International Conference on Learning Representations (ICLR)*, 2022.

C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki. LAION-400M: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Association for Computational Linguistics (ACL)*, 2018.

N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

P. Stock and M. Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. *arXiv preprint arXiv:1711.11443*, 2017.

B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li. Yfcc100m: The new data in multimedia research. In *Communications of the Association for Computing Machinery (ACM)*, 2016.

Y. Tian, O. J. Henaff, and A. van den Oord. Divide and contrast: Self-supervised learning from uncurated data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

E. van Miltenburg. On the use of human reference data for evaluating automatic image descriptions. *arXiv preprint arXiv:2006.08792*, 2020.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

B. Wang and A. Komatsuzaki. GPT-J-6B: A 6 billion parameter autoregressive language model. `https://github.com/kingoflolz/mesh-transformer-jax`, 2021.

Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

K. Yang, J. Yau, L. Fei-Fei, J. Deng, and O. Russakovsky. A study of face obfuscation in imagenet. In *International Conference on Machine Learning (ICML)*, 2022.

L. Yang, K. Qinami, L. Fei-Fei, J. Deng, and O. Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Conference on fairness, accountability, and transparency (FaccT)*, 2020.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

X. Zhai, J. Puigcerver, A. Kolesnikov, P. Ruyssen, C. Riquelme, M. Lucic, J. Djolonga, A. S. Pinto, M. Neumann, A. Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer. LiT: Zero-shot transfer with locked-image text tuning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

A SETUP AND EXPERIMENTAL DETAILS

A.1 DATASETS

In Appendix Figure 6, we present (random) samples from the MS-COCO (Lin et al., 2014), Conceptual Captions (Sharma et al., 2018) and YFCC datasets (Thomee et al., 2016). We use the 2017 version of COCO, which contains five human-written captions along with multi-object image labels for each image.

- "A table topped with plates and glasses with eating utensils.."
- "a fork is laying on a small white plate"
- "dirty dishes on a table, and a bottle of something."
- "a table top with some dishes on top of it",
- "A table full of dirty dishes is pictured in this image."

- "An All Nippon Airways 777 sitting at a gate on the tarmac."
- "a large air plane on a run way"
- "A jumbo jet being serviced at an airport."
- "A large blue and white jetliner sitting on top of a tarmac.",
- "A large airliner preparing for departure at an airport."

- "A man jumping a horse over an obstacle."
- "A person jumping a horse over an object."
- "An equestrian competitor and his horse jumping over a stile"
- "A horse and jockey jump over bush hurdles .",
- "A rider and horse jump over a wooden brush obstacle."

- "A couple of zebra standing on top of a dirt field."
- "Some zebras walking around in a field looking around"
- "Some very cute zebras in a big dusty field."
- "A small zebra standing next to a bigger zebra.",
- "The baby Zebras stripes are much closer together than an adults."

Figure 6: Dataset examples: MS-COCO (Lin et al., 2014)

Licenses. These datasets were obtained by scraping images from online hosting services (e.g. Flickr). Thus, the ownership of the images lies with the respective individuals that uploaded them. Nevertheless, as per their terms of agreement, the images can be used for research purposes.

“paratroopers load onto a helicopter.”

“Close up hands of woman typing text message on smart phone in a cafe.”

“woman in a bathrobe is smiling to camera in the forest”

“Girls in old time dresses selling flowers are pictured taking a rest of a bench.”

“A shrimp has pairs of legs.”

Figure 6: Dataset examples: Conceptual Captions (Sharma et al., 2018)

“Kenneth Phan #7 A Day in the Life of DC is a photo project meant to capture a flavor of the region through the eyes of the participants. Participants submitted twelve photos taken on May 30, 2009. Photos by Kenneth Phan”

“Pombas New York - USA 27 de Setembro 2013”

“Um, the girls that live at the house I lived in 13 years ago are huge @foursquare fans! #amazing @ 600 euclid 4sq.commPKNZv (posted via FlickSquare)”

“squares Created for dA Users Gallery Challenge #43 – Winter Stock 1 Model with thanks to Reine-Haru”

“Stripes and Squares Love the contrast of the light on the keyboard, stripes and squares”

Figure 6: Dataset examples: YFCC (Thomee et al., 2016)

Random examples of "full" captions:

- "A light bulb. A brown bulletin board. The bookcases are full. Books and papers on the floor. The carpet is brown. Books on the shelf. A flexible desk lamp. Magazines on the bench. Corded telephone mounted on wall. a book on a book shelf. Books on the desk. Sticker on the box is white and red. The white cord of the phone."
- "Bright light on the ceiling. A short brown wallpaper. Desktop monitor and keyboard. A set of items on the floor. The bookcases are full. Books and papers on the floor. The carpet is brown. Large clock on crowded bookshelf. A black table lamp not lighted. Black desk chair. Picture frame with three pictures. Sticker on the box is white and red. The wall phone is white. The white cord of the phone."
- "A brown bulletin board. Monitor on the table. Pile of books on floor. The brown wallpaper. Black padded piano bench. The carpet is brown. Books on the shelf. Magazines under a black psp. Corded telephone mounted on wall. White and black clock. Chair by the desk. Picture frame with three pictures. Sticker on the box is white and red. The wall phone is white. The white cord of the phone."

Random examples of "quadrant" captions:

- "A green plastic object on the table. Magazines under a black psp. Cardboard box with red and white sticker. Magazines on the bench."
- "Magazines on the bench."
- "Green package on top of cardboard box. Magazines sitting on top of stool. Magazines under a black psp. Magazines on the bench."

Random examples of "full" captions:

- "A blue double decker bus. a window on a building. a window on a building. Yellow and blue door. White clouds in blue sky. Cross on top of the building. a window on a bus. Tire has blue rim. Rectangular window with curve on sides. Number 35 on the back of the bus."
- "White clouds in blue sky. Bus. Building has a window. Varta printed in blue. Stickers on the window. White clouds in blue sky. Cross on top of the building. a window on a bus. Rectangular window with curve on sides. The bus number is 35. Number 220 on the back of the bus."
- "Building has a window. Building has a window. a window on a building. A parked double decker bus. Varta printed in blue. Folding door of the bus. White clouds in blue sky. Cross on top of the building. a window on a bus. Rectangular window with curve on sides. The bus number is 35. 220 printed in gray."

Random examples of "quadrant" captions:

- "David printed in black."
- "Logo on the side of the bus. Windows with papers on it. Stickers on the window.."
- "Building has a window. Building has a window. David printed in black. Windows with papers on it."

Figure 6: Dataset examples: VisualGenome (Krishna et al., 2017)

Like most large-scale datasets, COCO, CC and YFCC have not been extensively vetted, and may contain identifying information or offensive content. Characterizing the pervasiveness of these issues is an important and active area of research. That being said, we do not redistribute the data, our work is unlikely to significantly further the risks from these datasets.

A.2 MODELS

We rely on existing open source implementations of CLIP (Ilharco et al., 2021) and SimCLR (Falcon & the PyTorch Lightning team, 2019) for all our experiments, with a ResNet-50 image encoder (feature dimension=2048), and a linear and MLP projection head respectively. We use the transformer architecture from Radford et al. (2021) for encoding captions in CLIP. Unless otherwise specified, we use five captions per image to train CLIP_S. For downstream transfer, we train a linear probe using task data.

Supervised baseline on COCO. We trained a ResNet-50 classifier from scratch on the COCO dataset. The classifier was trained to predict the presence/absence of each of the 80 object classes per image (i.e., 80 binary classification tasks) as COCO has multi-object labels per image. We then evaluate the accuracy of the model by aggregating (in a class balanced manner) the correctness of each of these binary predictions.

A.3 HYPERPARAMETERS

We ran an extensive hyperparameter grid for CLIP and SimCLR on MS-COCO and used the same configuration in the rest of our experiments. These defaults are stated in Appendix Table 3.

Model	Batch Size	Epochs	Warmup	lr	wd
Supervised	1024	200	10	10^{-3}	10^{-6}
SimCLR	1024	200	10	10^{-2}	10^{-6}
CLIP	1024	200	10	10^{-3}	0.1

Table 3: Default hyperparameters for model training.

We use the Adam optimizer with a cosine lr schedule for all the models. All other hyperparameters are defaults from standard implementations of SimCLR [4] and CLIP.[5]

Exceptions. We train CLIP/SimCLR on CC/YFCC-2M for 100 epochs due to computational restrictions. For corpora smaller than 100K (Figure 2), we scale up the number of epochs to keep the number of iterations roughly comparable.

Data augmentations. The PyTorch pseudo-code for the default SimCLR and CLIP data from prior work augmentation are as follows:

$$\begin{aligned} T_{SimCLR} \;=\; \{&\texttt{RandomResizedCrop}(\texttt{size} = 224),\\ &\texttt{RandomHorizontalFlip}(\texttt{p} = 0.5)\\ &\texttt{RandomApply}(\texttt{ColorJitter}(0.8, 0.8, 0.8, 0.2), \texttt{p} = 0.8)\\ &\texttt{RandomGrayscale}(\texttt{p} = 0.2),\\ &\texttt{GaussianBlur}(\texttt{kernel_size} = 23, \texttt{p} = 0.5)\} \end{aligned}$$

$$\begin{aligned} T_{CLIP} \;=\; \{&\texttt{RandomResizedCrop}(\texttt{size} = 224,\\ &\texttt{scale} = (0.9, 1.0),\\ &\texttt{interpolation} = \texttt{BICUBIC})\} \end{aligned}$$

Note that for our experiments, unless otherwise specified, we use the standard SimCLR set for the Supervised/SimCLR/CLIP models.

[4] https://pytorch-lightning-bolts.readthedocs.io/en/latest/self_supervised_models.html

[5] https://github.com/mlfoundations/open_clip

Linear probe. We train the probe using cross-entropy loss on CLIP/SimCLR features of dimensionality 2048. In cases where the downstream task data is imbalanced, we re-weight the loss to account for it. We also evaluate class balanced accuracy at test time. For each downstream task, we train the probe for 250 epochs using an SGD optimizer. We use a batch size of 256, weight decay of 10^{-6} and momentum 0.9. We perform a grid search for the best learning rate (using the validation set), considering values between $3{\times}10^{-2}$ and 10. We also consider 3 random seeds.

Confidence intervals. We report 95% confidence intervals obtained via bootstrapping over the test set, as well as the three random seeds used for the linear probe. Due to space constraints, we do not always report them in the main paper, but include a detailed table for all our experiments with confidence intervals in the Appendix.

A.4 COMPUTE

We train each of our models of 4 NVIDIA A100 GPUs. Training both CLIP and SimCLR models takes on the order of 8-10 hours for a pre-training corpus of size $\sim$100K.

A.5 BLIP RECAPTIONING

To generate BLIP captions for images from the CC and YFCC datasets, we use the BLIP captioning model (Li et al., 2022). In particular, we use the provided[6] ViT-Base with nucleus sampling (top_p of 0.9, repetition penalty of 1.1, and text length range [5, 40]), varying the random seed to generate multiple captions per image. (Random) image-BLIP caption pairs are shown in Appendix Figure 7.

Dataset: CC

- "portrait of a young boy sitting in the leaves in a park - stock image."
- "toddler boy in a coat sitting on leaves with arms up to the air, smiling and laughing - stock photo."
- "An image of a little boy sitting on the leaves in a park - stock image."

Dataset: CC

- "The men are walking on a dirt ground with equipment in the background."
- "military soldiers in uniforms carrying weapons and soldiers on their back in a desert"
- "Officials and soldiers stand in the desert, looking at a vehicle with missiles."

Dataset: YFCC

- "Signs of various silhouettes of people dancing, standing, and laying on the street."
- "lot of bronze colored women holding their arms up with their hands together in front of a metal wall art"
- "Sculptures on a wall of various silhouettes and dance positions."

Dataset: YFCC

- "I love the white swan in the foreground with the water behind him."
- "an image group of white birds in a green area near some water"
- "the swan is standing on the green grass near the water"

Figure 7: Random images from CC and YFCC alongside BLIP captions.

[6]checkpoint https://storage.googleapis.com/sfr-vision-language-research/BLIP/models/model_base_caption_capfilt_large.pth from https://github.com/salesforce/BLIP

A.6 SYNTHETIC COCO CAPTIONS

We construct these captions for MS-COCO using the available multi-object labels (see Figure 8 for examples). A synthetic caption is complete (incomplete) if it describes all (a random subset) of objects in the image. It is consistent (inconsistent) if it describes a given object using a single consistent term throughout the dataset (one from a set of manually curated synonyms) and whether we use a fixed template (one of a set of templates). In every case, we randomly order the objects that we describe.

As an example of this procedure, consider two images X_1 with labels ["plate", "cup", "cup", "refrigerator"] and X_2 with labels ["refrigerator", "cup", "potted plant", "tennis racket"].

- A consistent and complete caption would describe *all* the image objects using a single descriptor/object in random order. For instance:
  Caption(X_1) = "An image of a plate, two cups, and a refrigerator".
  Caption(X_2) = "An image of a refrigerator, potted plant, tennis racket and cup".
- On the other hand, consistent and incomplete captions would still use a single descriptor/object, but might (randomly) omit certain image objects. For instance,
  Caption(X_1) = "An image of a plate and cup".
  Caption(X_2) = "An image of a refrigerator, potted plant and tennis racket".
- Finally, an inconsistent caption uses one of a set of descriptors/object in the image. For instance, a cup might be described (randomly) as a "cup" or "glass". We also randomly vary the template. Concretely, inconsistent and incomplete captions might look like:
  Caption(X_1) = "An image of a plate and glass".
  Caption(X_2) = "An photo of a fridge, tennis racket and a cup".

For inconsistent captions, the set of templates we consider is: "A photo of {}", , "There are {}", ""{} together", "I see {}", "Shown here are {}", "You can see {}". The set of synonyms per COCO object are:

```
person:  ['human', 'man', 'woman', 'individual', 'person']
bicycle:  ['bike', 'two-wheeler', 'cycle']
car:  ['auto', 'car', 'motorcar', 'automobile']
motorcycle:  ['bike', 'two-wheeler', 'motorbike', 'moped',
'scooter']
airplane:  ['plane', 'airplane', 'aeroplane', 'air plane', 'jet',
'airplane']
bus:  ['vehicle', 'coach']
train:  ['locomotive', 'engine', 'carriage', 'wagon']
truck:  ['hand truck', 'truck', 'motortruck', 'motor truck']
boat:  ['ship', 'boat', 'vessel', 'watercraft']
traffic light:  ['light', 'stoplight', 'traffic signal', 'signal',
'stop light']
fire hydrant:  ['hydrant', 'firehydrant']
stop sign:  ['stop', 'road sign', 'sign']
parking meter:  ['meter', 'parking', 'pay station', 'parking
kiosk']
bench:  ['seat', 'park bench', 'seat']
bird:  ['bird', 'birdie']
cat:  ['cat', 'kitten']
dog:  ['puppy', 'pooch', 'canine']
horse:  ['horse', 'stallion']
sheep:  ['sheep']
cow:  ['cow']
elephant:  ['elephant']
bear:  ['bear']
zebra:  ['zebra']
giraffe:  ['giraffe']
backpack:  ['knapsack', 'backpack', 'rucksack', 'haversack',
'packsack']
```

```
umbrella:  ['umbrella']
handbag:  ['bag', 'handbag', 'purse']
tie:  ['tie', 'necktie']
suitcase:  ['bag', 'traveling bag', 'suitcase']
frisbee:  ['Frisbee']
skis:  ['ski']
snowboard:  ['snowboard']
sports ball:  ['ball', 'basketball', 'football']
kite:  ['kite']
baseball bat:  ['bat']
baseball glove:  ['glove']
skateboard:  ['skateboard', 'skate board']
surfboard:  ['surfboard', 'surf board']
tennis racket:  ['tennis racket', 'racket']
bottle:  ['bottle', 'jar']
wine glass:  ['glass']
cup:  ['mug', 'cup', 'shot glass']
fork:  ['fork']
knife:  ['knife']
spoon:  ['spoon']
bowl:  ['bowl', 'plate']
banana:  ['banana']
apple:  ['apple', 'granny smith']
sandwich:  ['sandwich', 'burger']
orange:  ['Orange', 'mandarin', 'clementine']
broccoli:  ['broccoli']
carrot:  ['carrot']
hot dog:  ['hot dog']
pizza:  ['pie', 'pizza']
donut:  ['donut', 'doughnut']
cake:  ['pastry', 'dessert', 'cake']
chair:  ['chair']
couch:  ['couch', 'lounge', 'sofa']
potted plant:  ['plant', 'houseplant']
bed:  ['bed']
dining table:  ['table']
toilet:  ['crapper', 'toilette', 'potty', 'lavatory', 'lav',
'can', 'bathroom', 'privy', 'toilet']
tv:  ['television receiver', 'telly', 'television', 'TV', 'tv
set', 'tv']
laptop:  ['laptop', 'laptop computer', 'notebook']
mouse:  ['computer mouse', 'mouse']
remote:  ['remote control', 'remote']
keyboard:  ['keyboard']
cell phone:  ['phone', 'mobile', 'cell phone', 'cell']
microwave:  ['microwave oven', 'microwave']
oven:  ['oven']
toaster:  ['toaster']
sink:  ['basin', 'sink']
refrigerator:  ['icebox', 'refrigerator', 'fridge']
book:  ['book', 'novel', 'textbook', 'story book']
clock:  ['clock', 'watch', 'wall clock']
vase:  ['vase', 'jar']
scissors:  ['scissors']
teddy bear:  ['stuffed toy', 'toy']
hair drier:  ['dryer', 'blow dryer']
toothbrush:  ['toothbrush']
```

A.7 VISUAL GENOME CAPTIONS

In our experiments, we also consider training CLIP models on the Visual Genome dataset Krishna et al. (2017). Here, we construct image captions using the available region descriptions. Concretely, for each image, we construct:

- 10 captions by randomly subsampling *all* the available region descriptions and concatenating them.
- 10 captions by randomly subsampling the available region descriptions in the *first quadrant alone* and concatenating them. We do so in order to further understand the effect of incompleteness (i.e., describing only a part of the image) in captions on CLIP's performance.

Examples of the generated captions are shown in Appendix Figure 6

Using the two sets of captions per image ("full" or "quadrant") we train CLIP (using a single caption per image) and CLIP_S (stochastically sampling one of the ten captions) models. We then measure the transfer performance of the models in Appendix Table 9.

A.8 MEASURING CAPTION VARIABILITY

In Section 3.3, we attempt to quantify the variability of captions corresponding to a given real-world dataset. To do so, we look to prior work in linguistics and natural language processing (discussed below) which has sought to study similar quantities. In our setting, we treat the entire set of dataset captions as a single document so that we can measure variability across them.

Measure of Lexical Diversity (MTLD). In linguistics research, the lexical diversity of a given text—the range of words used within it—has been long studied (see (Jarvis, 2013) for a discussion). As discussed in (Baese-Berk et al., 2021), "...samples with (lexical) greater diversity, may also include less repetition, more switches among topics, and use of multiple lexical items to refer to the same concept". A classical measure of lexical diversity is the *type-token ratio (TTR):* the ratio of unique words with respect to the total number of words in the text. However, this metric suffers from certain drawbacks including sensitivity to corpus length. To mitigate these drawbacks, McCarthy & Jarvis (2010) proposed the notion of *measure of lexical diversity* or MTLD. Intuitively, this measure captures the average length of words (within the given text) for which the TTR remains constant. A higher MTLD score indicates that the document contains less repititive (more variable) tokens. We use a standard implementation[7] to measure MTLD over the entire set of captions.

Unique n-grams. In the context of conversational agents in natural language processing, the number of unique n-grams is often used to assess the diversity of the generated text Li et al. (2015); Fung et al. (2020). We count the sum of unique 1, 2, 3-grams for the set of captions in a given dataset.

A.9 FILTERING CAPTIONS

In Section 4, we introduce a methodology to filter poor quality captions from a given source dataset. Using the fastText library,[8] we train a linear classifier bag-of-n-grams sentence embeddings (n=2) to distinguish a subset of source captions from those in the COCO validation set. We then use the classifier to filter the source dataset (CC/YFCC), only selecting the ones that are (mis)classified as being COCO like.

In Appendix Figure 9, we present a (random) subset of filtered examples from the YFCC dataset. Compared to random YFCC samples (cf. Appendix Figure 6), the ones in Appendix Figure 9 have much shorter captions—often without attributes such as dates, urls and hashtags. This difference is more apparent if we consider, for instance, the top-30 most frequent 1-grams mentioned in YFCC captions before and after filtering.

Before filtering: `-, new, photo, day, taken, one, &, photos, see, like, 2013, view, 2012, park, first, 2011, around, old, part, 2010,\,`

[7] https://github.com/jennafrens/lexical_diversity

[8] https://github.com/facebookresearch/fastText

Complete and Consistent:

- "A photo of four bowls, a oven, seven cups, a refrigerator, two persons, a spoon, two cakes"

Incomplete and consistent:

- "A photo of a person, six cups, three bowls, two cakes, a oven."

Incomplete and Inconsistent:

- "A photo of a kitchen, two women, two shot glasses."
- "I see a oven, a kitchen, two mugs, a kitchen, a man."

Complete and Consistent:

- "A photo of a person, a tennis racket, a sports ball, a car."

Incomplete and consistent:

- "A photo of a car, a person, a tennis racket."

Incomplete and Inconsistent:

- "a sports ball, a motorcar together."
- "There is a woman."

Figure 8: Random image samples from MS-COCO alongside our synthetic captions.

`city, @, street, national, time, please, york, state, :, house, center, san, may, visit, go, research, near, use, back, get, 2008, 2, south, great, lake, two, central, north, little`

After filtering: `street, looking, two, water, people, one, beach, white, man, view, road, train, near, black, sign, front, blue, side, red, old, snow, small, tree, bridge, next, playing, river, top, day, little, room, walking, light, back, three, station, around, sitting, big, window, car, table, outside, dog, green, park, food, taken, ready, picture`

A.10 Augmenting captions with GPT-J

We propose a methodology to augment captions contained in existing datasets by using a pre-trained language model (in our case GPT-J-6b, referred to as GPT-J) to paraphrase them (Section 4). To this end, we rely on in-context learning, wherein we provide GPT-J with some (four) paired caption-paraphrase examples (using the five human-provided COCO captions) as the context. We then ask GPT-J to paraphrase a given target caption. For instance, a query to the model might look like:

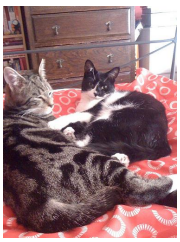
```
Paraphrase the sentence below
Input:  A little boy standing next to a dog in a field.
Output:  A dog parked filled with people and a bunch of
different dogs.


Paraphrase the sentence below
Input:  Some people are on the sandy beach flying kites.
Output:  a sunny day at the beach with colorful kites in the
sky


Paraphrase the sentence below
```

"Orc/Troll There's a face only a mother could love."

"Pedal Board 9 Back Camera."



"Kittens Morrissey and Marr relax on the bed."

"Park in Santa Clara Complete with Che graffiti."

Figure 9: Filtered image-caption pairs from the YFCC dataset (see Section 4).

```
Input:  A living room filled with furniture and a table.
Output:  A living room with a nice couch and a coffee table.

Paraphrase the sentence below
Input:  A couple of people on a surfboard in the ocean.
Output:  A dog is lying on the surfboard as it surfs on a
wave.

Paraphrase the sentence below
Input:  a handful of snowmen   make this with a glove .
Output:
```

We use temperature sampling to generate multiple diverse captions for a given image-caption pair from the dataset. Examples for the CC dataset are shown in Appendix Figure 10.

CC caption: "young beautiful businesswoman holding the documents and crossing the road in financial district ."
GPT-J paraphrases:

- "A women in business attire crossing the road in financial district."
- "A young beautiful businesswoman is crossing the road."
- "A young woman crossing the road in financial district."

CC caption: "Island with waterfalls and rainbows floating above clouds in a dreamy sky – stock photo #"
GPT-J paraphrases:

- "An island with waterfalls and rainbows floating above clouds in a dreamy sky."
- "A rainbow in the sky with a cloud and a rainbow in the water"
- "A rainbow floating in a dreamy sky and a waterfall"

CC caption: "Young friends eating pizza in the park"
GPT-J paraphrases:

- "The young friends are eating pizza in the park"
- "A group of young people eating pizza in the park"
- "Some kids are eating pizza in the park"

CC caption: "Actors attend the spring fashion show during fashion week."
GPT-J paraphrases:

- "A fashion show that's attended by actors."
- "People attending a spring fashion show during fashion week."
- "The spring fashion show was held during fashion week."

Figure 10: Caption paraphrases generated using in-context learning with GPT-J.

B ADDITIONAL EXPERIMENTS

In Appendix Tables 4-13, we report per-task performance for all our experiments. In Appendix Table 4, we also illustrate the performance of SimCLR/CLIP models trained using the simpler data augmentations typically used for CLIP training (cf. Appendix A.3). One can see that both models perform worse with this modification—with the performance of CLIP dropping by 10% and that of SimCLR by 50%.

For COCO, we also consider a variant of SimCLR, which we refer to as SimCLR$_{+lab}$, that factors in label information in the transformation $T(x)$. Specifically, for a given image x, x_+ is a data augmented version of another COCO image which has at least one object in common with x. We see that factoring label information does improve SimCLR's performance considerably, putting it between vanilla CLIP and CLIP$_\text{S}$. However, note for typical pre-training datasets such as CC and YFCC, we do not have access to such "expert" object labels. Instead, we can take advantage of captions to improve the equivalences learned by the model.

Model	SUP	SimCLR$_-$	SimCLR	SimCLR$_{+lab}$	CLIP$_-$	CLIP	CLIP$_\text{S}$
COCO	90.5 ± 1.5	60.4 ± 2.4	88.9 ± 1.6	89.3 ± 1.5	84.9 ± 1.9	88.4 ± 1.7	89.8 ± 1.6
Aircraft	31.6 ± 0.9	2.3 ± 0.3	40.6 ± 1.0	47.0 ± 1.0	30.3 ± 1.0	41.4 ± 1.0	46.4 ± 1.0
Birdsnap	11.8 ± 0.4	0.7 ± 0.1	18.5 ± 0.5	20.8 ± 0.5	14.0 ± 0.4	17.6 ± 0.5	20.0 ± 0.5
Cal101	65.8 ± 0.7	3.8 ± 0.3	71.5 ± 0.7	80.4 ± 0.6	53.6 ± 0.8	73.2 ± 0.7	78.4 ± 0.6
Cal256	53.7 ± 0.5	3.1 ± 0.2	58.6 ± 0.4	65.7 ± 0.4	41.5 ± 0.5	60.4 ± 0.5	65.6 ± 0.5
Cars	21.7 ± 0.5	1.2 ± 0.1	31.4 ± 0.6	39.3 ± 0.7	23.4 ± 0.5	35.8 ± 0.6	41.5 ± 0.6
CIFAR-10	74.8 ± 0.5	23.2 ± 0.5	82.1 ± 0.4	81.5 ± 0.5	74.0 ± 0.5	83.6 ± 0.4	84.6 ± 0.4
CIFAR-100	46.7 ± 0.6	6.0 ± 0.3	57.3 ± 0.6	56.8 ± 0.6	50.4 ± 0.6	60.8 ± 0.6	62.5 ± 0.6
DTD	55.9 ± 1.4	6.2 ± 0.6	61.7 ± 1.3	60.3 ± 1.3	48.2 ± 1.4	65.7 ± 1.3	66.7 ± 1.3
Flowers	63.5 ± 0.7	4.6 ± 0.3	77.4 ± 0.6	81.4 ± 0.6	68.2 ± 0.7	80.5 ± 0.6	84.0 ± 0.6
Food	47.1 ± 0.4	4.0 ± 0.1	58.7 ± 0.3	56.4 ± 0.4	51.8 ± 0.4	60.9 ± 0.4	65.3 ± 0.4
Pets	45.9 ± 1.0	6.3 ± 0.5	57.3 ± 0.9	63.0 ± 0.9	44.6 ± 0.9	57.0 ± 0.9	61.2 ± 0.9
SUN	44.5 ± 0.4	1.3 ± 0.1	51.9 ± 0.4	52.2 ± 0.4	37.6 ± 0.4	50.8 ± 0.4	54.9 ± 0.4
μ_{Tx}	47.2 ± 0.2	5.2 ± 0.1	56.0 ± 0.2	58.7 ± 0.2	44.8 ± 0.2	57.5 ± 0.1	61.3 ± 0.2

Table 4: Extended comparison of transfer performance of supervised, SimCLR and CLIP pre-trained models. Here SimCLR$_-$ and CLIP$_-$ denote models trained with the default CLIP data augmentation transforms instead of the SimCLR ones (cf. Appendix A.3). SimCLR$_{+lab}$ refers to SimCLR models trained by picking x_+ to be a different image with the same label as x.

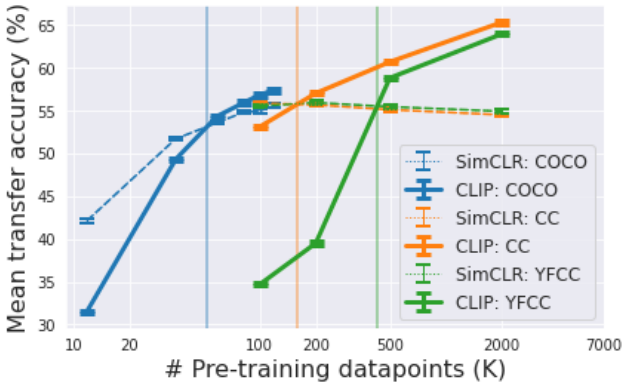


Figure 11: Transfer performance of SimCLR, CLIP and CLIP_S as we vary the number of COCO samples used for pre-training.

Model	CLIP	CLIP	CLIP_S	CLIP	CLIP_S
Complete	✓	✗	✗	✗	✗
Consistent	✓	✗	✗	✓	✓
COCO	88.8 ± 1.7	88.4 ± 1.7	89.3 ± 1.6	88.3 ± 1.7	89.2 ± 1.5
Aircraft	46.6 ± 1.0	44.5 ± 1.0	46.6 ± 1.0	45.6 ± 1.0	45.8 ± 1.0
Birdsnap	18.9 ± 0.5	17.2 ± 0.5	18.6 ± 0.5	18.5 ± 0.5	19.1 ± 0.5
Cal101	77.3 ± 0.6	75.3 ± 0.7	76.8 ± 0.6	76.1 ± 0.7	76.0 ± 0.6
Cal256	63.3 ± 0.5	59.9 ± 0.5	63.0 ± 0.4	61.4 ± 0.5	63.6 ± 0.5
Cars	42.4 ± 0.6	41.6 ± 0.6	42.7 ± 0.7	41.2 ± 0.6	42.8 ± 0.6
CIFAR-10	83.3 ± 0.4	82.4 ± 0.4	82.9 ± 0.4	83.7 ± 0.4	83.2 ± 0.4
CIFAR-100	60.5 ± 0.6	59.0 ± 0.6	58.9 ± 0.5	59.9 ± 0.5	60.1 ± 0.6
DTD	64.3 ± 1.3	63.7 ± 1.3	66.1 ± 1.3	63.4 ± 1.2	65.2 ± 1.2
Flowers	82.1 ± 0.5	78.3 ± 0.6	79.5 ± 0.6	79.3 ± 0.6	80.6 ± 0.6
Food	61.4 ± 0.3	57.6 ± 0.4	60.9 ± 0.4	59.0 ± 0.3	61.9 ± 0.4
Pets	60.0 ± 0.9	57.1 ± 1.0	58.8 ± 0.9	59.8 ± 0.9	60.6 ± 1.0
SUN	52.1 ± 0.4	49.6 ± 0.4	53.1 ± 0.4	50.6 ± 0.4	52.7 ± 0.4
μ_{Tx}	59.2 ± 0.1	56.6 ± 0.2	58.9 ± 0.2	57.7 ± 0.2	59.3 ± 0.2

Table 5: The impact of intra-dataset variations in captions on CLIP's transfer performance. Here, we use synthetic captions for pre-training, constructed using COCO multi-object image labels. We vary whether these captions are consistent (i.e., do they use a single term to describe a given object?) and complete (i.e., do they describe all image objects?). We also consider a variant of CLIP, CLIP_S, which uses multiple captions per image.

	Aircraft	Birdsnap	Ctech101	Ctech256	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food-101	Pets	SUN937
SimCLR	40.6	18.5	71.5	58.6	31.5	82.1	57.3	61.7	77.4	58.7	57.3	51.9
CLIP	41.4	17.6	73.2	60.4	35.8	83.6	60.8	65.7	80.5	60.9	57.0	50.8
CLIP + SimCLR loss	40.4	17.8	**81.5**	**61.6**	**36.3**	**84.3**	**62.1**	**67.1**	79.2	**62.0**	**58.1**	**53.2**

Table 6: Effect of incorporating SimCLR loss into CLIP on downstream transfer performance.

Model	SimCLR				CLIP			
Dataset size	100K	200K	500K	2M	100K	200K	500K	2M
Aircraft	40.5 ± 1.0	40.3 ± 1.0	39.3 ± 0.9	37.9 ± 1.0	35.5 ± 1.0	39.9 ± 1.0	41.6 ± 1.0	45.1 ± 1.0
Birdsnap	20.2 ± 0.5	20.7 ± 0.5	20.6 ± 0.5	20.6 ± 0.5	15.1 ± 0.5	17.5 ± 0.5	19.8 ± 0.5	24.0 ± 0.6
Cal101	70.7 ± 0.7	70.3 ± 0.7	70.3 ± 0.7	69.0 ± 0.7	67.7 ± 0.8	73.5 ± 0.7	79.0 ± 0.6	84.8 ± 0.6
Cal256	57.7 ± 0.5	57.3 ± 0.5	57.3 ± 0.5	56.7 ± 0.5	54.4 ± 0.4	60.0 ± 0.5	65.9 ± 0.4	73.9 ± 0.4
Cars	33.3 ± 0.6	31.2 ± 0.6	29.6 ± 0.6	27.5 ± 0.6	29.8 ± 0.6	33.8 ± 0.6	37.7 ± 0.6	42.6 ± 0.7
CIFAR-10	81.0 ± 0.5	80.4 ± 0.5	79.3 ± 0.5	79.8 ± 0.5	82.5 ± 0.4	83.9 ± 0.4	85.6 ± 0.4	86.8 ± 0.4
CIFAR-100	58.1 ± 0.6	57.4 ± 0.6	56.4 ± 0.5	56.4 ± 0.6	59.7 ± 0.6	63.2 ± 0.5	64.8 ± 0.5	67.8 ± 0.6
DTD	62.8 ± 1.3	63.9 ± 1.2	64.5 ± 1.2	64.3 ± 1.3	63.7 ± 1.3	67.6 ± 1.3	70.3 ± 1.3	74.7 ± 1.2
Flowers	80.8 ± 0.6	80.3 ± 0.6	80.2 ± 0.6	79.4 ± 0.6	76.5 ± 0.6	80.8 ± 0.6	85.0 ± 0.5	88.8 ± 0.5
Food	57.6 ± 0.3	58.3 ± 0.4	57.0 ± 0.4	56.7 ± 0.4	56.6 ± 0.4	59.4 ± 0.4	62.7 ± 0.3	68.1 ± 0.3
Pets	58.2 ± 0.9	57.9 ± 1.0	56.8 ± 0.9	55.9 ± 0.9	49.7 ± 1.0	53.5 ± 0.9	60.2 ± 0.9	65.2 ± 0.9
SUN	49.4 ± 0.4	49.8 ± 0.4	49.7 ± 0.4	49.6 ± 0.4	45.9 ± 0.4	50.9 ± 0.4	55.3 ± 0.4	61.8 ± 0.4
μ_{Tx}	55.9 ± 0.2	55.3 ± 0.2	55.1 ± 0.2	54.5 ± 0.2	53.1 ± 0.2	57.0 ± 0.2	60.7 ± 0.2	65.3 ± 0.2

Table 7: Transfer performance of SimCLR and CLIP models after pre-training on CC subsets.

Model	SimCLR				CLIP			
Dataset size	100K	200K	500K	2M	100K	200K	500K	2M
Aircraft	39.5 ± 0.9	39.3 ± 1.0	38.0 ± 0.9	36.3 ± 0.9	17.0 ± 0.7	21.2 ± 0.8	41.5 ± 0.9	43.0 ± 0.9
Birdsnap	19.2 ± 0.5	18.9 ± 0.5	19.7 ± 0.5	19.0 ± 0.5	8.3 ± 0.4	10.4 ± 0.4	19.8 ± 0.5	26.2 ± 0.6
Cal101	71.0 ± 0.7	71.1 ± 0.7	70.3 ± 0.7	68.4 ± 0.7	42.7 ± 0.7	51.4 ± 0.7	75.2 ± 0.7	82.1 ± 0.6
Cal256	56.9 ± 0.5	58.5 ± 0.5	58.6 ± 0.5	57.7 ± 0.5	32.9 ± 0.4	38.2 ± 0.5	62.4 ± 0.5	70.5 ± 0.4
Cars	33.1 ± 0.6	29.8 ± 0.6	28.1 ± 0.6	26.8 ± 0.6	11.8 ± 0.4	15.5 ± 0.5	36.1 ± 0.7	37.4 ± 0.6
CIFAR-10	80.4 ± 0.5	80.6 ± 0.4	80.2 ± 0.5	79.7 ± 0.5	71.1 ± 0.5	72.9 ± 0.5	83.5 ± 0.4	86.0 ± 0.4
CIFAR-100	56.8 ± 0.5	58.2 ± 0.5	56.8 ± 0.5	57.2 ± 0.6	46.6 ± 0.6	47.7 ± 0.6	62.3 ± 0.6	66.2 ± 0.5
DTD	64.8 ± 1.2	67.0 ± 1.2	67.3 ± 1.2	67.0 ± 1.3	41.9 ± 1.3	49.9 ± 1.3	69.1 ± 1.2	74.3 ± 1.1
Flowers	80.9 ± 0.6	81.2 ± 0.6	80.5 ± 0.6	80.5 ± 0.6	47.6 ± 0.7	54.9 ± 0.8	83.4 ± 0.5	89.4 ± 0.4
Food	57.4 ± 0.4	57.9 ± 0.4	56.9 ± 0.4	57.4 ± 0.4	36.6 ± 0.4	43.0 ± 0.3	61.7 ± 0.3	67.4 ± 0.4
Pets	54.8 ± 1.0	55.4 ± 1.0	55.6 ± 0.9	55.6 ± 0.9	30.4 ± 0.9	34.0 ± 0.9	55.7 ± 1.0	61.9 ± 0.9
SUN	51.4 ± 0.4	52.9 ± 0.4	53.1 ± 0.4	53.2 ± 0.4	29.2 ± 0.4	34.7 ± 0.4	54.6 ± 0.4	62.8 ± 0.4
μ_{Tx}	55.5 ± 0.2	55.9 ± 0.2	55.4 ± 0.2	54.9 ± 0.2	34.7 ± 0.2	39.5 ± 0.2	58.8 ± 0.2	63.9 ± 0.2

Table 8: Transfer performance of SimCLR and CLIP models after pre-training on YFCC subsets.

Model	Captions	Aircraft	Birdsnap	Ctech101	Ctech256	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food-101	Pets	SUN937
CLIP	quadrant	29.9	11.3	64.0	46.3	21.0	78.6	53.5	56.4	61.4	47.8	39.4	39.9
CLIP_S (10)	quadrant	42.8	16.9	79.8	59.2	38.5	81.0	57.4	62.8	77.6	56.0	54.5	50.0
CLIP	full	42.1	15.9	75.9	57.9	36.2	82.5	59.4	62.2	76.7	56.4	52.3	48.8
CLIP_S (10)	full	44.2	18.2	82.4	62.5	39.4	83.0	59.7	65.4	81.2	60.3	56.1	54.6

Table 9: Linear probe accuracy for CLIP models trained on VisualGenome (Krishna et al., 2017).

Method	Size	Epochs	Aircraft	Birdsnap	Ctech101	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food-101	Pets	SUN937
BYOL ((Tian et al., 2021))	100M	1000	47.5	31.3	84.0	44.3	85.0	63.9	75.2	93.4	67.9	71.1	63.4
MoCLR ((Tian et al., 2021))	100M	1000	45.6	29.4	85.6	41.1	87.8	69.9	75.8	92.9	67.7	67.7	63.4
CLIP	2M	100	43.0	26.2	82.1	37.4	86.0	66.2	74.3	89.4	67.4	61.9	62.8

Table 10: Comparison of our results to (Tian et al., 2021).

Model	CLIP				CLIP_S	
Dataset	CC		YFCC		CC	YFCC
Dataset size	100K	500K	100K	500K	100K	100K
Aircraft	35.5 ± 1.0	41.6 ± 1.0	35.4 ± 0.9	42.8 ± 0.9	40.1 ± 1.0	41.3 ± 0.9
Birdsnap	15.1 ± 0.5	19.8 ± 0.5	15.9 ± 0.5	20.7 ± 0.6	17.8 ± 0.5	19.2 ± 0.5
Cal101	67.7 ± 0.8	79.1 ± 0.6	67.9 ± 0.7	79.7 ± 0.6	75.1 ± 0.6	75.8 ± 0.6
Cal256	54.4 ± 0.5	65.9 ± 0.5	55.8 ± 0.5	67.6 ± 0.4	61.8 ± 0.5	62.6 ± 0.5
Cars	29.8 ± 0.6	37.7 ± 0.6	29.6 ± 0.6	37.8 ± 0.6	37.3 ± 0.6	38.1 ± 0.6
CIFAR-10	82.5 ± 0.5	85.6 ± 0.4	82.9 ± 0.4	85.6 ± 0.4	83.6 ± 0.4	82.7 ± 0.4
CIFAR-100	59.7 ± 0.6	64.8 ± 0.5	60.9 ± 0.6	65.2 ± 0.5	62.2 ± 0.6	60.9 ± 0.6
DTD	63.7 ± 1.2	70.2 ± 1.2	64.1 ± 1.3	71.0 ± 1.3	67.7 ± 1.3	68.7 ± 1.2
Flowers	76.5 ± 0.6	85.0 ± 0.5	77.7 ± 0.6	86.2 ± 0.5	81.4 ± 0.6	83.7 ± 0.6
Food	56.6 ± 0.4	62.7 ± 0.3	57.3 ± 0.4	64.0 ± 0.3	59.6 ± 0.4	61.3 ± 0.3
Pets	49.7 ± 1.0	60.2 ± 0.9	49.6 ± 0.9	61.6 ± 0.9	54.7 ± 0.9	56.6 ± 0.9
SUN	45.9 ± 0.4	55.3 ± 0.4	47.7 ± 0.4	57.1 ± 0.4	52.5 ± 0.4	54.9 ± 0.4
μ_{Tx}	53.7 ± 0.2	60.7 ± 0.2	54.8 ± 0.2	61.8 ± 0.2	57.8 ± 0.2	58.8 ± 0.2

Table 11: Effect of using BLIP captions for CC/YFCC images in CLIP training.

Dataset	CC	YFCC
Dataset size	100K	500K
Aircraft	37.0 ± 1.0	41.0 ± 1.0
Birdsnap	15.5 ± 0.5	21.1 ± 0.5
Cal101	71.1 ± 0.7	78.2 ± 0.7
Cal256	55.9 ± 0.5	64.9 ± 0.4
Cars	30.9 ± 0.6	35.2 ± 0.6
CIFAR-10	82.9 ± 0.5	85.1 ± 0.4
CIFAR-100	59.3 ± 0.6	63.4 ± 0.6
DTD	63.8 ± 1.3	71.8 ± 1.2
Flowers	76.3 ± 0.7	84.3 ± 0.5
Food	57.4 ± 0.4	64.1 ± 0.3
Pets	52.7 ± 0.9	59.3 ± 0.9
SUN	47.4 ± 0.4	56.4 ± 0.4
μ_{Tx}	54.2 ± 0.2	60.4 ± 0.2

Table 12: Effect of caption filtering on CLIP's transfer performance.

Dataset	CC	COCO
Dataset size	200K	120K
Aircraft	41.9 ± 0.9	44.7 ± 1.0
Birdsnap	18.8 ± 0.5	18.6 ± 0.5
Cal101	77.4 ± 0.7	75.9 ± 0.6
Cal256	63.5 ± 0.4	62.8 ± 0.4
Cars	38.2 ± 0.6	40.8 ± 0.6
CIFAR-10	84.0 ± 0.4	84.1 ± 0.4
CIFAR-100	62.5 ± 0.6	61.3 ± 0.6
DTD	68.1 ± 1.2	65.3 ± 1.3
Flowers	82.4 ± 0.6	81.9 ± 0.6
Food	60.4 ± 0.4	62.0 ± 0.4
Pets	56.0 ± 1.0	59.6 ± 1.0
SUN	53.1 ± 0.4	51.9 ± 0.4
μ_{Tx}	58.8 ± 0.3	58.9 ± 0.3

Table 13: Training CLIP_S models using additional captions generated via GPT-J paraphrasing.

B.1 DESCRIPTIVE VS. NOISY CAPTIONS

In Section 3.2, we study the effect of the average descriptiveness of dataset captions on CLIP's transfer performance. We find that the CC and YFCC datasets tend to have captions with lower descriptiveness than the manually-sourced COCO captions. A natural question to ask is whether these captions are just irrelevant noise. After all, they have been collected via automated scraping with little or no post-processing. In prior work, Alikhani et al. (2020) provide a taxonomy of relevant, non-noisy captions: Visible ("presents information that is intended to recognizably characterize what is depicted in the image"), Subjective ("describes the speaker's reaction to, or evaluation of, what is depicted in the image"), Action ("describes an extended, dynamic process of which the moment captured in the image is a representative snapshot"), Story ("providing a free-standing description of the circumstances depicted in the image") and Meta ("allows the reader to draw inferences not just about the scene depicted in the image but about the production and presentation of the image itself"). A noisy or irrelevant caption would be one that does not fall into the aforementioned categories.

They then recruit expert annotators to categorize CC images based on this taxonomy. It turns out that CC captions are actually well-aligned with the corresponding images: only 3% of the captions are irrelevant noise. The remainder of the captions are relevant, but might not always fall into the "Visible" category (their analogue to our notion of "descriptive"). To demonstrate that the same is holds for YFCC, we manually annotate random dataset samples using this taxonomy—see Figure 12.

Figure 12: Random samples from the YFCC dataset with different average descriptiveness levels. We manually categorize the corresponding captions into the taxonomy of (Alikhani et al., 2020).