VirTex: Learning Visual Representations from Textual Annotations

Karan Desai Justin Johnson
University of Michigan
{kdexd,justincj}@umich.edu

Abstract

The de-facto approach to many vision tasks is to start from pretrained visual representations, typically learned via supervised training on ImageNet. Recent methods have explored unsupervised pretraining to scale to vast quantities of unlabeled images. In contrast, we aim to learn high-quality visual representations from fewer images. To this end we revisit supervised pretraining, and seek data-efficient alternatives to classification-based pretraining. We propose VirTex – a pretraining approach using semantically dense captions to learn visual representations. We train convolutional networks from scratch on COCO Captions, and transfer them to downstream recognition tasks including image classification, object detection, and instance segmentation. On all tasks, VirTex yields features that match or exceed those learned on ImageNet – supervised or unsupervised – despite using up to ten times fewer images.

1. Introduction

The prevailing paradigm for learning visual representations is first to *pretrain* a convolutional network [1, 2] to perform image classification on ImageNet [3, 4], then *transfer* the learned features to downstream tasks [5, 6]. This approach has been wildly successful, and has led to significant advances on a wide variety of computer vision problems such as object detection [7], semantic [8] and instance [9] segmentation, image captioning [10–12], and visual question answering [13, 14]. Despite its practical success, this approach is expensive to scale since the pretraining step relies on images annotated by human workers.

For this reason, there has been increasing interest in *unsupervised pretraining* methods that use unlabeled images to learn visual representations which are then transferred to downstream tasks [15–21]. Some recent approaches have begun to match or exceed supervised pretraining on ImageNet [22–26], and have been scaled to hundreds of millions [22, 25, 27, 28] or billions [24] of images.

Continuing to scale unsupervised pretraining to ever-larger sets of unlabeled images is an important scientific

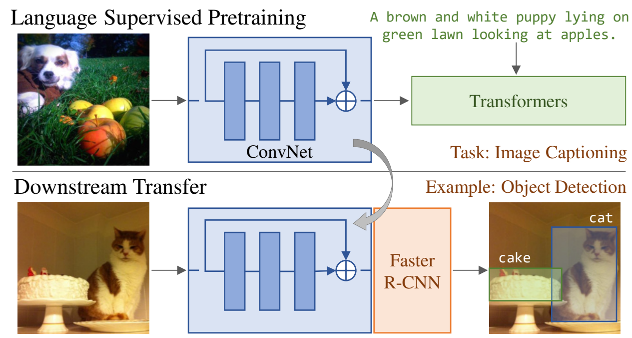


Figure 1: **Learning visual features from language:** First, we jointly train a ConvNet and Transformers using image-caption pairs, for the task of image captioning (top). Then, we transfer the learned ConvNet to several downstream vision tasks, for example object detection (bottom).

goal. But we may also ask whether there are alternate ways of pretraining that learn high-quality visual representations with *fewer* images. To do so, we revisit *supervised* pretraining and seek an alternative to traditional classification pretraining that uses each image more efficiently.

In this paper we present an approach for learning **Vi**sual **r**epresentations from **Tex**tual annotations (VirTex). Our approach is straightforward: first, we jointly train a ConvNet and Transformer [29] *from scratch* to generate natural language captions for images. Then, we transfer the learned features to downstream visual recognition tasks (Figure 1).

We believe that using language supervision is appealing due to its *semantic density*. Figure 2 compares different pretraining tasks for learning visual representations. Captions provide a semantically denser learning signal than unsupervised contrastive methods and supervised classification. Hence, we expect that using textual features to learn visual features may require fewer images than other approaches.

Another benefit of textual annotations is simplified data collection. To collect classification labels, typically human experts first build an ontology of categories [3, 4, 30, 31], then complex crowdsourcing pipelines are used to elicit labels from non-expert users [32, 33]. In contrast, natural language descriptions do not require an explicit ontology and can easily be written by non-expert workers, leading to a

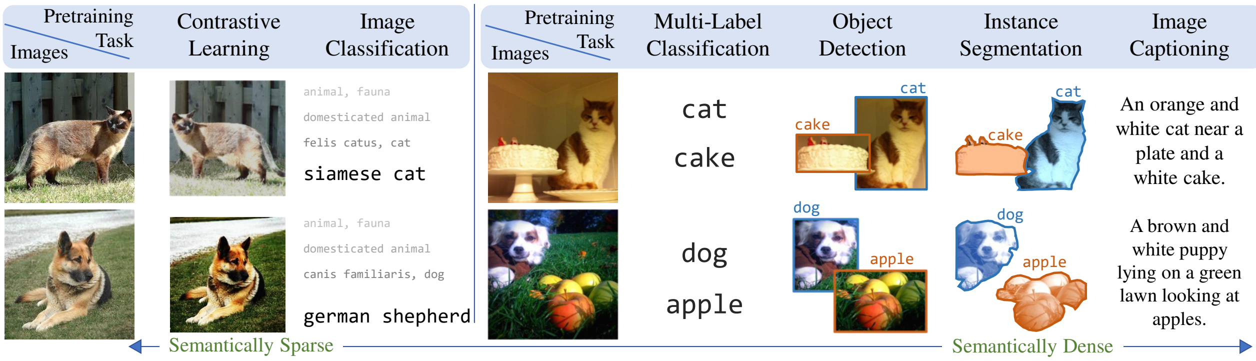


Figure 2: **Comparison of pretraining tasks for learning visual representations:** Contrastive self-supervised learning methods provide a *semantically sparse* learning signal, encouraging different transforms of an image to have similar features. Image classification pairs an image with a single semantic concept, providing moderate semantic density. Multi-label classification, object detection, and instance segmentation increase semantic density by labeling and localizing multiple objects. Captions describe multiple objects, their attributes, relationships, and actions, giving a semantically dense learning signal. In this work, we aim to leverage this semantic density of captions to learn visual representations in a data-efficient manner.

simplified data collection pipeline [34–36]. Large quantities of weakly aligned images and text can also be obtained from internet images [37–39].

Our main contribution is to show that natural language can provide supervision for learning transferable visual representations with better data-efficiency than other approaches. We train models from scratch on the COCO Captions dataset [36], and evaluate the learned features on downstream tasks including image classification, object detection, instance segmentation, and low-shot recognition. On all tasks, VirTex matches or exceeds the performance of existing methods for supervised or unsupervised pretraining on ImageNet, despite using up to 10× fewer images. Our code and pretrained models are available at https://github.com/kdexd/virtex

2. Related Work

Our work is related to recent efforts to move beyond supervised pretraining on ImageNet using alternate data sources or pretraining tasks.

Weakly Supervised Learning scales beyond supervised pretraining with a *quantity over quality* approach, and learns on large numbers of images with noisy labels from web services. *Li et al.* [40] trains visual N-gram models on the YFCC-100M dataset [41], that provides 100M Flickr images with user-provided tags. Recent works [42–44] also use JFT-300M [42] dataset, curated by automatic labeling of images from web signals using Google's internal tooling. Weakly-supervised learning has also been studied on up to 3.5B Instagram images, using hashtags as labels [45, 46]. These approaches learn visual representations with large quantities of images with low-quality labels; in contrast we focus on using fewer images with high-quality annotations.

Self-Supervised Learning focuses on learning visual representations by solving *pretext tasks* defined on unlabeled images. Early works on self-supervised learning proposed hand-crafted pretext tasks, such as context prediction [15], colorization [17, 18], solving jigsaw puzzles [47], predicting rotation [19], inpainting [16], clustering [27], and generative modeling [48]. Recent works are based on contrastive learning [49, 50], encouraging similarity between image features under different random transformations on single input image [24–26, 51, 52]. Other approaches use contrastive losses based on context prediction [20, 23], mutual information maximization [21, 53, 54], predicting masked regions [55], and clustering [56–58].

These methods lack semantic understanding as they rely on low-level visual cues (color, texture), whereas we leverage textual annotations for semantic understanding. Unlike these methods, our approach can leverage additional metadata such as text, when scaled to internet images [37–39].

Vision-and-Language Pretraining attempts to learn joint representations of image-text paired data that can be transferred to multimodal downstream tasks such as visual question answering [13, 14, 59, 60], visual reasoning [61, 62], referring expressions [63], and language-based image retrieval [35]. Inspired by the success of BERT [64] in NLP, several recent methods use Transformers [29] to learn transferable joint representations of images and text [65–72].

These methods employ complex pretraining pipelines: they typically **(1)** start from an ImageNet-pretrained CNN; **(2)** extract region features using an object detector fine-tuned on Visual Genome [73], following [74]; **(3)** optionally start from a pretrained language model, such as BERT [64]; **(4)** combine the models from (2) and (3), and train a multimodal transformer on Conceptual Captions [37]; **(5)** fine-tune the model from (4) on the downstream task. In this pipeline, all vision-and-language tasks are downstream from the initial visual representations learned on ImageNet.

In contrast, we pretrain via image captioning, and put vision tasks downstream from vision-and-language pretraining.

Concurrent Work: Our work is closest to *Sariyildiz et al.* [75] on learning visual representations from captions via image conditioned masked language modeling, with one major difference – we train our entire model from scratch, whereas they rely on pretrained BERT for textual features. Moreover, we evaluate on additional downstream tasks like object detection and instance segmentation. Our work is also closely related to *Stroud et al.* [76] on learning video representations using paired textual metadata, however they solely operate and evaluate their method on video tasks.

3. Method

Given a dataset of image-caption pairs, our goal is to learn visual representations that can be transferred to downstream visual recognition tasks. As shown in Figure 2, captions carry rich semantic information about images, including the presence of objects (*cat*, *plate*, *cake*); attributes of objects (*orange and white cat*); spatial arrangement of objects (*cat near a plate*); and their actions (*looking at apples*). Learned visual representations that capture such rich semantics should be useful for many downstream vision tasks.

To this end, we train *image captioning* models to predict captions from images. As shown in Figure 3, our model has two components: a *visual backbone* and a *textual head*. The visual backbone extracts visual features from an input image I. The textual head accepts these features and predicts a caption $C = (c_0, c_1, \dots, c_T, c_{T+1})$ token by token, where $c_0 =$ `[SOS]` and $c_{T+1} =$ `[EOS]` are fixed special tokens indicating the start and end of sentence. The textual head performs bidirectional captioning (*bicaptioning*): it comprises a *forward model* that predicts tokens left-to-right, and a *backward model* that predicts right-to-left. All model components are randomly initialized, and jointly trained to maximize the log-likelihood of the correct caption tokens

$$\begin{aligned}\mathcal{L}(\theta, \phi) = &\sum_{t=1}^{T+1} \log\Big(p(c_t \mid c_{0:t-1}, I; \phi_f, \theta)\Big) \\ &+ \sum_{t=0}^{T} \log\Big(p(c_t \mid c_{t+1:T+1}, I; \phi_b, \theta)\Big)\end{aligned} \tag{1}$$

where θ, ϕ_f, and ϕ_b are the parameters of the visual backbone, forward, and backward models respectively. After training, we discard the textual head and transfer the visual backbone to downstream visual recognition tasks.

Language Modeling: Our choice of pretraining task is image captioning [10–12] – a well-studied vision-and-language task, so far kept *downstream* from vision-based pretraining. We draw inspiration from recent work in NLP using language modeling as a pretraining task to learn transferable text representations. This involves training massive language models – either unidirectional [77] or bidirectional [78–81], for predicting tokens one by one. However, following BERT [64], many large-scale models [82, 83] instead use *masked language models* (MLMs): some tokens are randomly masked and are predicted by the model.

We performed preliminary experiments with MLMs, but like [64, 84] we observed that MLMs converge more slowly than directional models. We note that MLMs have poor sample efficiency, as they only predict a subset of tokens for each caption, while directional models predict all tokens. Due to computational constraints, we focus on directional models and leave MLMs to future work.

Visual Backbone: The visual backbone is a convolutional network which computes visual features of images. It inputs raw image pixels, and outputs a spatial grid of image features. During pretraining, these features are used to predict captions. In downstream tasks, we either train linear models on features extracted from the visual backbone, or fine-tune the visual backbone end-to-end.

In principle we could use any convolutional network architecture for the visual backbone. In our experiments we use a standard ResNet-50 [2] as the visual backbone to facilitate comparison with our baseline methods (Section 4). It accepts a 224×224 image and produces a 7×7 grid of 2048-dimensional features after the final convolutional layer. During pretraining, we apply a linear projection layer to the visual features before passing them to the textual head to facilitate decoder attention over visual features. This projection layer is not used in downstream tasks.

Textual Head: The textual head receives features from the visual backbone and predicts captions for images. It provides a learning signal to the visual backbone during pretraining. Our overall goal is not to predict high-quality captions, but instead to learn transferable visual features.

The textual head comprises two identical language models which predict captions in forward and backward directions respectively. Following recent advances in language modeling, we use Transformers [29], which use multiheaded self-attention both to propagate information along the sequence of caption tokens, as well as to fuse visual and textual features. We closely follow the transformer decoder architecture from [29], but use GELU [85] rather than ReLU, following [64, 79]. We briefly review the architecture here; refer to [29] for a more complete description.

During training, the forward model receives two inputs: image features from the visual backbone, and a caption describing the image. Image features are a matrix of shape $N_I \times D_I$ giving a D_I-dimensional vector for each of the $N_I = 7 \times 7$ positions in the final layer of the visual backbone. As described earlier, the caption $C = (c_0, c_1, \dots, c_T, c_{T+1})$ is a sequence of $T + 2$ tokens, with $c_0 =$ `[SOS]` and $c_{T+1} =$ `[EOS]`. It is trained to predict $C_{1:T+1}$ token-by-token, starting with c_0. The prediction c_t

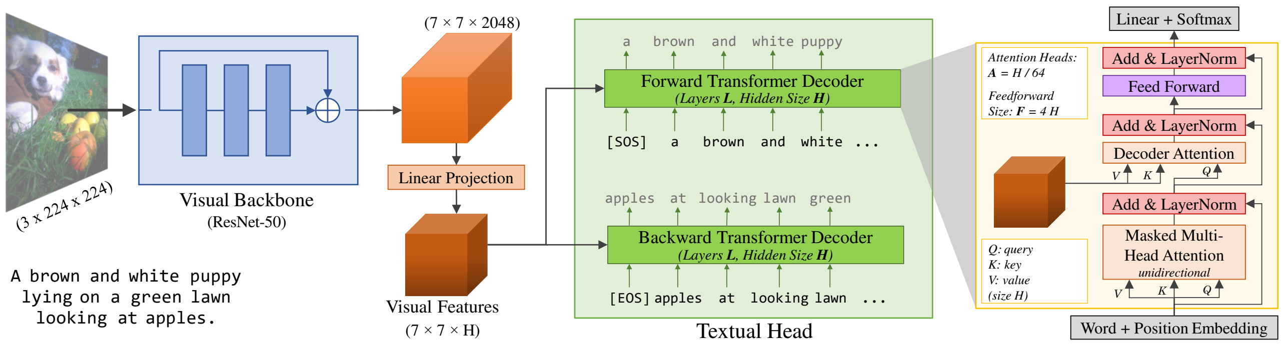


Figure 3: **VirTex pretraining setup:** Our model consists of a *visual backbone* (ResNet-50), and a *textual head* (two unidirectional Transformers). The visual backbone extracts image features, and textual head predicts captions via bidirectional language modeling (*bicaptioning*). The Transformers perform masked multiheaded self-attention over caption features, and multiheaded attention over image features. Our model is trained end-to-end from scratch. After pretraining, the visual backbone is transferred to downstream visual recognition tasks.

is *causal* – it only depends on past predictions $c_{0:t-1}$ and visual features. The backward model is similar; it operates right-to-left – trained to predict $C_{T:0}$, given c_{T+1}.

First, we convert the tokens of C to vectors via learned token and positional embeddings, followed by elementwise sum, layer normalization [86] and dropout [87]. Next, we process these vectors through a sequence of Transformer layers. As shown in Figure 3, each layer performs masked multiheaded self-attention over token vectors, multiheaded attention between token vectors and image vectors, and applies a two-layer fully-connected network to each vector. These three operations are each followed by dropout, wrapped in a residual connection, and followed by layer normalization. Token vectors interact only through self-attention; the masking in this operation maintains causal structure of the final predictions. After the last Transformer layer, we apply a linear layer to each vector to predict unnormalized log-probabilities over the token vocabulary.

The forward and backward models consist of independent Transformer layers. However they share the same token embedding matrix (similar to [77]) which is also reused at the output layers of each model (similar to [88, 89]).

Model Size: Several architectural hyperparameters control the size of our textual head. We can control the *width* of each Transformer layer by varying its *hidden size* H, the number of *attention heads* A used in multiheaded attention, and the *feedforward size* F of the fully-connected network. We follow [64] and always set $A = H/64$ and $F = 4H$; this allows us to control the width of our textual head by varying H. We can also control the *depth* of our textual head by varying the number of transformer layers L.

Tokenization: We tokenize captions with SentencePiece [90] using the BPE algorithm [91]. Prior to tokenization we lowercase and strip accents from captions. We build a vocabulary of 10K tokens, including boundary (`[SOS]`, `[EOS]`) and out-of-vocab (`[UNK]`) tokens. Following [79, 80] we restrict subword merges between letters and punctuation to prevent redundant tokens such as `dog?` and `dog!`. Compared to basic tokenization schemes often used for image captioning that split on whitespace [10, 11], BPE makes fewer linguistic assumptions, exploits subword information, and results in fewer out-of-vocab tokens.

Training Details: We train on the `train2017` split of the COCO Captions dataset [36], which provides $118K$ images with five captions each. During training we apply standard data augmentation: we randomly crop to 20-100% of the original image size, apply color jitter (brightness, contrast, saturation, hue), and normalize using the ImageNet mean color. We also apply random horizontal flips, also interchanging the words 'left' and 'right' in the caption.

We train using SGD with momentum 0.9 [92, 93] and weight decay 10^{-4} wrapped in LookAhead [94] with $\alpha = 0.5$ and 5 steps. Following [64], we do not apply weight decay to layer normalization and bias parameters in Transformers. We perform distributed training across 8 GPUs with batch normalization [95] per GPU, following [22]. We train with a batch size of 256 images (32 per GPU) for 500K iterations (≈1080 epochs). We use linear learning rate warmup [22] for the first 10K iterations followed by cosine decay [96] to zero. We found that the visual backbone required a higher LR than the textual head for faster convergence. The visual backbone uses a max LR of 2×10^{-1}; the textual head uses 10^{-3}. We implement our models using PyTorch [97] with native automatic mixed-precision [98].

We observe that performance on image captioning has a positive but imprecise correlation with performance on downstream visual recognition tasks (Refer Appendix A.4). We thus perform *early stopping* based on the performance of our visual backbone on downstream PASCAL VOC [99] linear classification (see Section 4.1) since it is fast to evaluate and correlates well with our other downstream tasks.

Method	Annotations	Cost (hours)†	VOC07	IN-1k
MoCo-COCO	self-sup.	–	63.3	41.1
Multi-label Clf.	labels	11.1K [30]	86.2	46.2
Instance Segmentation	masks	30.0K [30]	82.3	51.0
VirTex (1 caption)	captions	1.3K [100]	84.2	50.2
VirTex (5 caption)	captions	6.5K [100]	**88.7**	**53.8**

Table 1: **Annotation Cost Efficiency:** We compare downstream performance of various pretraining methods on COCO. VirTex outperforms all other methods trained on the same set of images with best performance vs. cost tradeoff.
†: *For COCO* `train2017` *split, see Appendix A.1 for more details.*

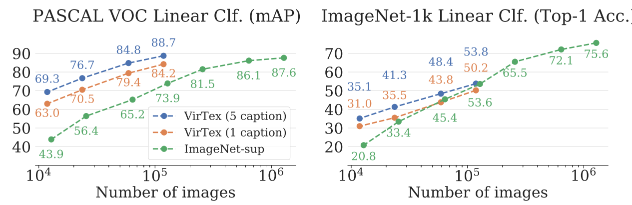


Figure 4: **Data Efficiency:** We compare VirTex and IN-sup models trained using varying amounts of images. VirTex closely matches or significantly outperforms IN-sup on downstream tasks despite using $10\times$ fewer images. *IN-1k: Models using* $\leq 10^5$ *images are mean of 5 trials, std dev.* ≤ 1.0.

4. Experiments

In our experiments, we aim to demonstrate the effectiveness of learning visual features via natural language supervision. As described in Section 3, we jointly train a VirTex model from scratch on the COCO Captions [36] dataset. Here, we evaluate the features learned by visual backbone on six downstream vision tasks. We select these tasks based on two common mechanisms for transfer learning: where the visual backbone is either used as (a) frozen feature extractor, or (b) weight initialization for fine-tuning.

4.1. Image Classification with Linear Models

Our first set of evaluations involve training linear models on frozen visual backbones – we compare VirTex with various pretraining methods to test our two hypotheses:

1. Learning visual features via captions is cheaper than using other types of annotations, like labels and masks.
2. Using semantically dense captions helps with learning effective visual features using fewer training images.

We evaluate on two datasets: PASCAL VOC [99] and ImageNet-1k [4]. We choose these tasks based on their simplicity and evaluation speed. We briefly describe the setup here. Refer Appendix A.1 for more details.

PASCAL VOC: We follow same protocol as SwAV [58] (highly similar to [22, 25]); we train on VOC07 `trainval` split (9K images, 20 classes) and report mAP on `test` split. We train per-class SVMs on 2048-dimensional global average pooled features extracted from the last layer of the visual backbone. For each class, we train SVMs for cost values $C \in \{0.01, 0.1, 1, 10\}$ and select best C by 3-fold cross-validation. Other SVM hyperparameters are same as [22].

ImageNet-1k: We follow similar protocol as MoCo [24] and SwAV [58]: we train on the ILSVRC 2012 `train` split and report top-1 accuracy on `val` split. We train a linear classifier (fully connected layer + softmax) on 2048-dimensional global average pooled features extracted from the last layer of the visual backbone. We train with batch size 256 distributed across 8 GPUs for 100 epochs. We use SGD with momentum 0.9 and weight decay 0. We set the initial LR to 0.3 and decay it to zero by cosine schedule.

Annotation Cost Efficiency: We believe that using captions is appealing due to a simple and cost-efficient collection pipeline. Here, we test our first hypothesis by comparing various pretraining methods on COCO, each drawing supervision from different annotation types (Figure 2):

- **MoCo-COCO (self-supervised):** We train a MoCo-v1 model on COCO images with default hyperparameters.
- **Multi-label Classification (labels):** We use COCO object detection annotations (80 classes), and train a ResNet-50 backbone to predict a K-hot vector with values $1/K$ with a KL-divergence loss, similar to [45].
- **Instance Segmentation (masks):** We use a pretrained Mask R-CNN from Detectron2 model zoo [101], and extract its ResNet-50 backbone for downstream tasks. This model is trained from scratch on COCO, following [102].
- **VirTex (captions):** We train a VirTex model on COCO Captions, with ResNet-50 visual backbone and $L = 1, H = 2048$ textual head. Note that COCO Captions provides five captions per image, which effectively increases image-caption pairs by five-fold. Hence for a fair comparison, we also train an additional VirTex model using only one randomly selected caption per image.

Results are shown in Table 1. We also compare annotation costs in terms of worker hours. For labels and masks, we use estimates reported by COCO [30]. For captions, we estimate the cost based on `nocaps` [100] [1], that follows a similar data collection protocol as COCO. We observe that VirTex outperforms all methods, and has the best performance vs. cost tradeoff, indicating that learning visual features using captions is more cost-efficient than labels or masks.

Data Efficiency: We believe that the semantic density of captions should allow VirTex to learn effective visual features from fewer images than other methods. To test our hypothesis, we compare VirTex and ImageNet-supervised models (**IN-sup**) trained using varying amount of images from COCO Captions and ImageNet-1k respectively.

We train 4 VirTex models using $\{10, 20, 50, 100\}\%$ of COCO Captions (118K images) and 7 ResNet-50 models using $\{1, 2, 5, 10, 20, 50, 100\}\%$ of ImageNet-1k (1.28M

[1] We could not find estimates for COCO Captions in existing literature.

Method	Pretrain Images	Annotations	VOC07	IN-1k
MoCo-IN v1 [24]	1.28M	self-sup.	79.4	60.8
PCL v1 [57]	1.28M	self-sup.	83.1	61.5
SwAV (200 ep.) [58]	1.28M	self-sup.	87.9	72.7
ICMLM$_{\texttt{att-fc}}$ [75] †	118K	captions	87.5	47.9
VirTex	118K	captions	88.7	53.8

Table 2: **Comparison with other methods:** We compare downstream performance of VirTex with recent SSL methods and concurrent work. †*: Uses pretrained BERT-base.*

images). Similar to prior experiments, we also train 4 VirTex models using one randomly selected caption per image. All VirTex models use $L = 1, H = 2048$ textual heads.

We show results in Figure 4. On VOC07, VirTex-100% outperforms IN-sup-100% (mAP **88.7** vs **87.6**), despite using 10× fewer images (118K vs. 1.28M). When using similar amount of images, VirTex consistently outperforms IN-sup (**blue**, **orange** vs **green**), indicating superior data efficiency of VirTex. We also observe that given the same number of captions for training, it is better to spread them over more images – VirTex-50% (1 caption) significantly outperforms VirTex-10% (5 captions) (mAP **79.4** vs **69.3**).

Comparison with IN-sup on ImageNet-1k classification is unfair for VirTex, since IN-sup models are trained for the downstream task, using the downstream dataset. Even so, VirTex-100% outperforms IN-sup-10% (**53.8** vs. **53.6**, 118K vs. 128K images), and consistently outperforms it when both methods use fewer than 100K images.

Comparison with other methods: Here, we compare VirTex with recent pretraining methods that have demonstrated competitive performance on downstream tasks.

- **Self-supervised pretraining:** We choose three recent methods based on their availability and compatibility with our evaluation setup – MoCo [24], PCL [57], and SwAV [58]. We choose models trained with a similar compute budget as ours (8 GPUs, 200 ImageNet epochs).
- **ICMLM (Concurrent Work):** We adapt numbers from *Sariyildiz et al.* [75]; evaluation may slightly differ. This model uses pretrained BERT [64] for textual features.
- **Note on vision-and-language pretraining:** Since we use captions, we also consider methods that learn multimodal representations for downstream vision-and-language tasks [65–72]). As described in Section 2, all these methods use an object detector trained on Visual Genome [73] (with ImageNet-pretrained backbone) to extract visual features, made available by [74]. These features are kept frozen, and do not learn from any textual supervision at all. Our comparison with ImageNet-supervised models subsumes this family of models.

Results are shown in Table 2. VirTex outperforms all methods on VOC07, despite being trained with much fewer images. On ImageNet-1k, comparison between self-supervised models and VirTex is unfair on both ends, as the former observes downstream images during pretraining, while the latter uses annotated images.

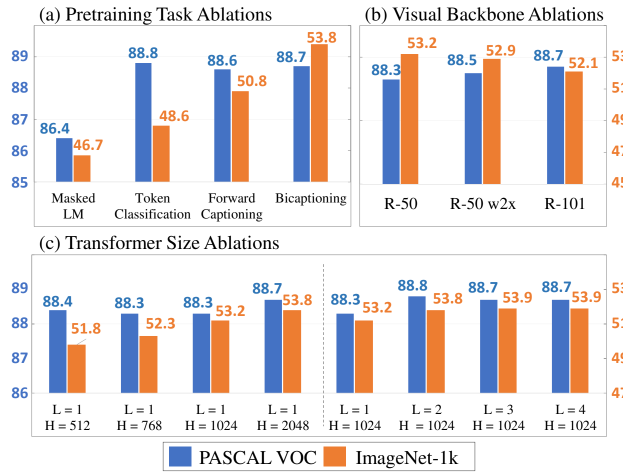


Figure 5: **Ablations. (a) Pretraining Tasks:** Bicaptioning improves over weaker pretraining tasks – forward captioning, token classification and masked language modeling. **(b) Visual Backbone:** Bigger visual backbones improve downstream performance – both, wider (R-50 w2×) and deeper (R-101). **(c) Transformer Size:** Larger transformers (wider and deeper) improve downstream performance.

4.2. Ablations

The preceeding linear classification experiments demonstrate the effectiveness and data-efficiency of VirTex. In this section, we conduct ablation studies to isolate the effects of our pretraining setup and modeling decisions, and uncover performance trends to seed intuition for future work. We evaluate all ablations on PASCAL VOC and ImageNet-1k linear classification, as described in Section 4.1.

Pretraining Task Ablations: We choose bicaptioning task as it gives a dense supervisory signal per caption. To justify this choice, we form three pretraining tasks with *sparser* supervisory signal and compare them with bicaptioning:

- **Forward Captioning:** We remove the backward transformer decoder and only perform left-to-right captioning.
- **Token Classification:** We replace the textual head with a linear layer and perform multi-label classification (Table 1, row 2). We use the set of caption tokens as targets, completely ignoring the linguistic structure of captions.
- **Masked Language Modeling (MLM):** We use a single bidirectional transformer in the textual head, and perform BERT-like masked language modeling. We randomly mask 15% of input tokens, and train the model to predict ground-truth tokens of masked positions.

All textual heads with transformers have $L = 1, H = 2048$.

Results are shown in Figure 5(a). Bicaptioning outperforms forward captioning, indicating that denser supervi-

Method	Pretrain Images	COCO Instance Segmentation						LVIS Instance Segmentation			PASCAL VOC Detection			iNat 18
		AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}_{all}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{mask}_{all}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{bbox}_{all}	AP^{bbox}_{50}	AP^{bbox}_{75}	Top-1
1) Random Init		36.7	56.7	40.0	33.7	53.8	35.9	17.4	27.8	18.4	33.8	60.2	33.1	61.4
2) IN-sup	1.28M	41.1	62.0	44.9	37.2	59.1	40.0	22.6	35.1	23.7	54.3	81.6	59.7	65.2
3) IN-sup-50%	640K	$40.3_{-0.8}$	$61.0_{-1.0}$	$44.0_{-0.9}$	$36.6_{-0.6}$	$58.0_{-1.1}$	$39.3_{-0.7}$	$21.2_{-1.4}$	$33.3_{-1.8}$	$22.3_{-1.4}$	$52.1_{-2.2}$	$80.4_{-1.2}$	$57.0_{-2.7}$	$63.2_{-2.0}$
4) IN-sup-10%	128K	$37.9_{-3.2}$	$58.2_{-3.8}$	$41.1_{-3.8}$	$34.7_{-2.5}$	$55.2_{-3.9}$	$37.1_{-2.9}$	$17.5_{-5.1}$	$28.0_{-7.1}$	$18.4_{-5.3}$	$42.6_{-11.7}$	$72.0_{-9.6}$	$43.8_{-15.9}$	$60.2_{-4.7}$
5) MoCo-IN	1.28M	$40.8_{-0.3}$	$61.6_{-0.4}$	$44.7_{-0.2}$	$36.9_{-0.3}$	$58.4_{-0.7}$	$39.7_{-0.3}$	$22.8_{+0.2}$	$35.4_{+0.3}$	$24.2_{+0.5}$	$56.1_{+1.8}$	$81.5_{-0.1}$	$62.4_{+0.7}$	$63.2_{-1.7}$
6) MoCo-COCO	118K	$38.5_{-0.6}$	$58.5_{-3.5}$	$42.0_{-2.9}$	$35.0_{-2.2}$	$55.6_{-3.5}$	$37.5_{-2.5}$	$20.7_{-1.9}$	$32.3_{-2.8}$	$21.9_{-1.8}$	$47.6_{-6.7}$	$75.4_{-6.2}$	$51.0_{-8.7}$	$60.5_{-4.4}$
7) VirTex	118K	$40.9_{-0.2}$	$61.7_{-0.3}$	$44.8_{-0.1}$	$36.9_{-0.3}$	$58.4_{-0.7}$	$39.7_{-0.3}$	$23.0_{+0.4}$	$35.4_{+0.4}$	$24.3_{+0.6}$	$55.3_{+1.0}$	$81.3_{-0.3}$	$61.0_{+1.3}$	$63.4_{-1.4}$

Table 3: **Fine-tuning Tasks for Transfer:** We compare VirTex with different pretraining methods across four downstream tasks. For each task, all methods use the same architecture. We initialize the ResNet-50 backbone weights from pretraining (except Random Init), which are then fine-tuned end-to-end. Performance gaps with IN-sup are shown on the side. On all tasks, VirTex significantly outperforms all methods that use similar amount of pretraining images. VirTex closely matches or exceeds ImageNet supervised and self-supervised methods, despite using $10\times$ fewer pretraining images.

sory signal from bidirectional modeling is beneficial. Bicaptioning and forward captioning both outperform token classification, demonstrating that learning to model the sequential structure of language improves visual features.

MLM performs quite worse than all three methods, possibly due to poor sample efficiency (discussed in Section 3) It may benefit from longer training schedules, however we leave this for future work due to computational constraints.

Visual Backbone Ablations: Bigger visual backbones tend to show improvements on many vision tasks [2, 9, 103]. We investigate whether VirTex models with bigger visual backbones can improve downstream performance. We train three VirTex models with $L = 1, H = 1024$ textual heads, and different visual backbones: (a) ResNet-50 (default), (b) ResNet-50 w2$\times$ [104] (2$\times$ channel width), and (c) ResNet-101 (2$\times$ depth). We observe that bigger visual backbones better results on VOC07, however the trends are opposite on ImageNet (Figure 5(b)). We believe it to be an optimization issue. See Appendix A.2 for comparison on other tasks.

Transformer Size Ablations: Prior work in language modeling has shown that larger Transformers tend to learn better *textual* features [80–83]. We investigate whether this holds for VirTex: do larger transformers in the textual head cause the visual backbone to learn better *visual* features? As discussed in Section 3, we may scale our textual head by increasing its *width* (hidden size H) or its *depth* (number of layers L). We investigate both, training VirTex models with:

– Fixed $L = 1$, increasing $H \in \{512, 768, 1024, 2048\}$.
– Fixed $H = 1024$, increasing $L \in \{1, 2, 3, 4\}$.

Results are shown in Figure 5(c) – increasing transformer size, both width and depth, generally improves downstream performance. Performance degrades slightly with very deep transformers ($L = 4$), indicating overfitting. We hope that massive transformers with billions of parameters will help when scaling VirTex to large-scale, more noisy image-text paired datasets [37–39] that are larger than COCO Captions.

4.3. Fine-tuning Tasks for Transfer

So far we have evaluated VirTex using features extracted from *frozen* visual backbones. Another common mechanisms for transfer learning is *fine-tuning*, where the entire visual backbone is updated for the downstream task.

We evaluate features learned using VirTex on four downstream tasks with fine-tuning: (a) Instance Segmentation on COCO [30]; (b) Instance Segmentation on LVIS [31]; and (c) Object Detection on PASCAL VOC [99]; (d) Fine-grained Classification on iNaturalist 2018 [107]. In all these experiments, we use the VirTex model with ResNet-50 visual backbone and a textual head with $L = 1, H = 2048$.

Baselines: Our main baselines are ImageNet-supervised (IN-sup) and MoCo. We consider three variants of IN-sup pretrained with $\{10, 50, 100\}\%$ of ImageNet images (Figure 4). Similarly for MoCo, we consider both MoCo-IN (Table 2) and MoCo-COCO (Table 1). We also include Random Init baseline, trained from scratch on downstream task.

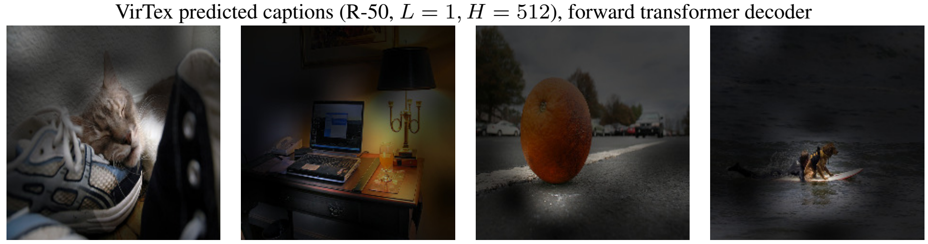
We follow the same evaluation protocol as MoCo [24] for all four tasks. We use Detectron2 [101] for tasks (a,b,c). Our IN-sup-100% results are slightly better than those reported in [24] – we use pretrained ResNet-50 model from `torchvision`, whereas they used the MSRA ResNet-50 model from Detectron [108]. We briefly describe implementation details here, refer Appendix A.3 for full details.

COCO Instance Segmentation: We train Mask R-CNN [9] models with ResNet-50-FPN backbones [109]. We initialize backbone with pretrained weights, train on `train2017` split, and evaluate on `val2017` split. We fine-tune all layers end-to-end with BN layers synchronized across GPUs [110] (*SyncBN*). We also use SyncBN in FPN layers. We train with batch size 16 distributed across 8 GPUs, following 2$\times$ schedule (180K iterations with initial LR 0.02, multiplied by 0.1 at iterations 120K and 160K).

LVIS Instance Segmentation: The LVIS dataset provides instance segmentation labels for a long tail of 1230 entry-

Backbone	Depth	Width	CIDEr	SPICE
R-50	1	512	**93.4**	18.1
R-50	1	768	93.1	**18.5**
R-50	1	1024	92.8	18.4
R-50	1	2048	92.1	18.4
R-50	1	1024	92.8	**18.4**
R-50	2	1024	94.1	**18.4**
R-50	3	1024	**95.5**	18.1
R-50	4	1024	92.9	17.6
R-50 w2×	1	1024	91.7	18.2
R-101	1	1024	**94.3**	**18.6**

VirTex predicted captions (R-50, $L = 1, H = 512$), forward transformer decoder



`a cat laying on a pair of blue` **`shoes`** | `a laptop computer sitting on top of a` **`desk`** | `an orange is sitting on the side of a` **`road`** | `a dog` **`riding`** `on a surfboard in the ocean`

Figure 6: **Image Captioning.** We report image captioning performance (**CIDEr** [105] and **SPICE** [106]) of VirTex models on COCO `val2017`. On the right, we show some predicted captions and attention visualizations on input image for the **`highlighted word`**. VirTex focuses on relevant regions to predict objects (**`shoes`**), background (**`road`**) and actions (**`riding`**).

level object classes and stresses the ability to recognize many object types from few training samples. We train Mask R-CNN models with ResNet-50-FPN backbones on `train_v1.0` and evaluate on `val_v1.0` split. Following MoCo settings, we keep BN parameters frozen for all IN-sup baselines. We train with 2× schedule as COCO, use class resampling and test-time hyperparameters (0.0 score threshold and 300 detections per image) same as [31].

PASCAL VOC Detection: We train Faster R-CNN [111] models with ResNet-50-C4 backbones on `trainval07+12` split, and evaluate on `test2007` split. Like COCO, we fine-tune all models with batch size 2 per GPU (8 GPUs) and SyncBN. We train for 24K iterations, including linear LR warmup for first 100 iterations. We start with LR 0.02 and divide it by 10 at iterations 18K and 22K.

iNaturalist 2018 Fine-grained Classification: The iNaturalist 2018 dataset provides labeled images for 8142 fine-grained categories, with a long-tailed distribution. We fine-tune the pretrained ResNet-50 with a linear layer end-to-end. We train on `train2018` split and evaluate on `val2018` split. We use SGD with momentum 0.9 and weight decay 10^{-4} for 100 epochs with batch size 256 distributed across 8 GPUs. Fine-tuning uses LR 0.025 (and Random Init uses 0.1), which is multiplied by 0.1 at epochs 70 and 90.

Results: We show results in Table 3. VirTex matches or exceeds ImageNet-supervised pretraining and MoCo-IN on all tasks (row **`2,5`** vs. **`7`**) despite using 10× fewer pretraining images. Moreover, VirTex significantly outperforms methods that use similar, or more pretraining images (row **`3,4,6`** vs. **`7`**), indicating its superior data-efficiency. Among all tasks, VirTex shows significant improvements on LVIS, indicating the effectiveness of natural language annotations in capturing the long tail of visual concepts in the real world.

4.4. Image Captioning

Our goal is to learn transferable visual features via textual supervision. To do so, we use image captioning as a pretraining task. Although our goal is not to advance the state-of-the-art in image captioning, in Figure 6 we show quantitative and qualitative results of VirTex models trained from scratch on COCO. All models show modest performance, far from current state-of-the-art methods, that commonly involve some pretraining. However, captioning metrics are known to correlate weakly with human judgement – we surpass human performance on COCO.

We show some predicted captions by VirTex (R-50, $L = 1, H = 512$) model. We apply *beam search* on the forward transformer decoder (5 beams) to decode most likely captions. The *decoder attention module* in this transformer attends over a 7×7 grid of image features through $A = 8$ heads at each time-step for predicting a token. We average these 7×7 attention weights over all the heads, and overlay them on 224×224 input image (via bicubic upsampling).

In Figure 6, we show visualizations for some tokens. We observe that our model attends to relevant image regions for making predictions, indicating that VirTex learns meaningful visual features with good semantic understanding.

5. Conclusion

We have shown that learning visual representations using textual annotations can be competitive to methods based on supervised classification and self-supervised learning on ImageNet. We solely focus on downstream vision tasks – future works can explore other tasks that transfer both the visual backbone and the textual head. Finally, using captions opens a clear pathway to scaling our approach to web-scale image-text pairs, that are orders of magnitude larger, albeit noisier than COCO Captions.

Acknowledgements: We thank Harsh Agrawal, Mohamed El Banani, Richard Higgins, Stefan Lee, Nilesh Kulkarni, Chris Rockwell for feedback on the paper. We thank Georgia Gkioxari, Ishan Misra, Saining Xie for help replicating evaluation setup, and Ross Girshick, Yuxin Wu for Detectron2. We thank Jia Deng for UMich ARC-TS for extra GPUs and cluster management. Finally, we thank the Starbucks outlets in Ann Arbor for free WiFi. This work was partially supported by the Toyota Research Institute (TRI). However, note that this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, 2015.

[5] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *ICML*, 2014.

[6] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *CVPR Workshops*, 2014.

[7] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask R-CNN," in *ICCV*, 2017.

[10] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan, "Show and tell: A neural image caption generator," in *CVPR*, 2015.

[11] Andrej Karpathy and Li Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *CVPR*, 2015.

[12] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.

[13] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh, "VQA: Visual question answering," in *ICCV*, 2015.

[14] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei, "Visual7w: Grounded question answering in images," in *CVPR*, 2016.

[15] Carl Doersch, Abhinav Gupta, and Alexei A Efros, "Unsupervised visual representation learning by context prediction," in *ICCV*, 2015.

[16] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016.

[17] Richard Zhang, Phillip Isola, and Alexei A Efros, "Colorful image colorization," in *ECCV*, 2016.

[18] Richard Zhang, Phillip Isola, and Alexei A Efros, "Split-brain autoencoders: Unsupervised learning by cross-channel prediction," in *CVPR*, 2017.

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.

[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[21] Yonglong Tian, Dilip Krishnan, and Phillip Isola, "Contrastive multiview coding," in *ECCV*, 2020.

[22] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra, "Scaling and benchmarking self-supervised visual representation learning," in *CVPR*, 2019.

[23] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord, "Data-efficient image recognition with contrastive predictive coding," *arXiv preprint arXiv:1905.09272*, 2019.

[24] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

[25] Ishan Misra and Laurens van der Maaten, "Self-supervised learning of pretext-invariant representations," in *CVPR*, 2020.

[26] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[27] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018.

[28] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin, "Unsupervised pre-training of image features on non-curated data," in *ICCV*, 2019.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.

[31] Agrim Gupta, Piotr Dollar, and Ross Girshick, "LVIS: A dataset for large vocabulary instance segmentation," in *CVPR*, 2019.

[32] Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei, "Scalable multi-label annotation," in *SIGCHI*, 2014.

[33] Ranjay A Krishna, Kenji Hata, Stephanie Chen, Joshua Kravitz, David A Shamma, Li Fei-Fei, and Michael S Bernstein, "Embracing error to enable rapid crowdsourcing," in *CHI*, 2016.

[34] Micah Hodosh, Peter Young, and Julia Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *JAIR*, 2013.

[35] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *TACL*, 2014.

[36] 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.

[37] 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.
[38] Junhua Mao, Jiajing Xu, Kevin Jing, and Alan L Yuille, “Training and evaluating multimodal word embeddings with large-scale web annotated images,” in *NIPS*, 2016.
[39] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg, “Im2Text: Describing images using 1 million captioned photographs,” in *NIPS*, 2011.
[40] Ang Li, Allan Jabri, Armand Joulin, and Laurens van der Maaten, “Learning visual n-grams from web data,” in *ICCV*, 2017.
[41] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li, “YFCC100M: The new data in multimedia research,” *Communications of the ACM*, 2016.
[42] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *ICCV*, 2017.
[43] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby, “Large scale learning of general visual representations for transfer,” *arXiv preprint arXiv:1912.11370*, 2019.
[44] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V Le, “Self-training with noisy student improves ImageNet classification,” in *CVPR*, 2020.
[45] Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten, “Exploring the limits of weakly supervised pretraining,” in *ECCV*, 2018.
[46] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan, “Billion-scale semi-supervised learning for image classification,” *arXiv preprint arXiv:1905.00546*, 2019.
[47] Mehdi Noroozi and Paolo Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *ECCV*, 2016.
[48] Jeff Donahue and Karen Simonyan, “Large scale adversarial representation learning,” in *NeurIPS*, 2019.
[49] Michael Gutmann and Aapo Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *AISTATS*, 2010.
[50] Raia Hadsell, Sumit Chopra, and Yann LeCun, “Dimensionality reduction by learning an invariant mapping,” in *CVPR*, 2006.
[51] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang, “Unsupervised embedding learning via invariant and spreading instance feature,” in *CVPR*, 2019.
[52] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin, “Unsupervised feature learning via non-parametric instance-level discrimination,” 2018.
[53] Philip Bachman, R Devon Hjelm, and William Buchwalter, “Learning representations by maximizing mutual information across views,” in *NeurIPS*, 2019.
[54] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio, “Learning deep representations by mutual information estimation and maximization,” *ICLR*, 2019.
[55] Trieu H Trinh, Minh-Thang Luong, and Quoc V Le, “Selfie: Self-supervised pretraining for image embedding,” *arXiv preprint arXiv:1906.02940*, 2019.
[56] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins, “Local aggregation for unsupervised learning of visual embeddings,” in *ICCV*, 2019.
[57] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C.H. Hoi, “Prototypical contrastive learning of unsupervised representations,” *arXiv preprint arXiv:2005.04966*, 2020.
[58] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *NeurIPS*, 2020.
[59] 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 *CVPR*, 2017.
[60] Drew A Hudson and Christopher D Manning, “GQA: A new dataset for real-world visual reasoning and compositional question answering,” in *CVPR*, 2019.
[61] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi, “A corpus for reasoning about natural language grounded in photographs,” in *ACL*, 2019.
[62] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi, “From recognition to cognition: Visual commonsense reasoning,” in *CVPR*, 2019.
[63] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg, “Referitgame: Referring to objects in photographs of natural scenes,” in *EMNLP*, 2014.
[64] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, 2019.
[65] Hao Tan and Mohit Bansal, “LXMERT: Learning cross-modality encoder representations from transformers,” in *EMNLP*, 2019.
[66] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee, “ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks,” in *NeurIPS*, 2019.
[67] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang, “VisualBERT: A simple and performant baseline for vision and language,” *arXiv preprint arXiv:1908.03557*, 2019.
[68] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai, “VL-BERT: Pre-training of generic visual-linguistic representations,” 2020.
[69] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou, “Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training,” *AAAI*, 2020.
[70] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu, “Uniter: Learning universal image-text representations,” *arXiv preprint arXiv:1909.11740*, 2019.
[71] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao, “Unified vision-language pre-training for image captioning and VQA,” *AAAI*, 2020.
[72] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, *et al.*, “Oscar: Object-semantics aligned pre-training for vision-language tasks,” in *ECCV*, 2020.
[73] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson,

Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael S Bernstein, and Li Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *IJCV*, 2017.

[74] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *CVPR*, 2018.

[75] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus, "Learning visual representations with caption annotations," in *ECCV*, 2020.

[76] Jonathan C Stroud, David A Ross, Chen Sun, Jia Deng, Rahul Sukthankar, and Cordelia Schmid, "Learning video representations from textual web supervision," in *ECCV*, 2020.

[77] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, "Deep contextualized word representations," in *NAACL*, 2018.

[78] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *NeurIPS*, 2019.

[79] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever, "Improving language understanding by generative pre-training," 2018.

[80] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, 2019.

[81] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[82] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "RoBERTa: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[83] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro, "Megatron-LM: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv:1909.08053*, 2019.

[84] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," in *ICLR*, 2020.

[85] Dan Hendrycks and Kevin Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[86] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[87] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, 2014.

[88] Hakan Inan, Khashayar Khosravi, and Richard Socher, "Tying word vectors and word classifiers: A loss framework for language modeling," *arXiv preprint arXiv:1611.01462*, 2016.

[89] Ofir Press and Lior Wolf, "Using the output embedding to improve language models," in *EACL*, 2017.

[90] Taku Kudo and John Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *EMNLP: System Demonstrations*, 2018.

[91] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Neural machine translation of rare words with subword units," in *ACL*, 2016.

[92] Boris T Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, 1964.

[93] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, "On the importance of initialization and momentum in deep learning," in *ICML*, 2013.

[94] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton, "Lookahead optimizer: k steps forward, 1 step back," in *NeurIPS*, 2019.

[95] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[96] Ilya Loshchilov and Frank Hutter, "SGDR: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[97] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[98] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, *et al.*, "Mixed precision training," in *ICLR*, 2018.

[99] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman, "The pascal visual object classes (VOC) challenge," *IJCV*, 2009.

[100] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson, "nocaps: novel object captioning at scale," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 8948–8957, 2019.

[101] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.

[102] Kaiming He, Ross Girshick, and Piotr Dollár, "Rethinking ImageNet pre-training," in *ICCV*, 2019.

[103] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017.

[104] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[105] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh, "CIDEr: Consensus-based image description evaluation," in *CVPR*, 2015.

[106] Peter Anderson, Basura Fernando, Mark Johnson, and

Stephen Gould, "SPICE: Semantic propositional image caption evaluation," in *ECCV*, 2016.
[107] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie, "The inaturalist species classification and detection dataset," in *CVPR*, 2018.
[108] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He, "Detectron." https://github.com/facebookresearch/detectron, 2018.
[109] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.
[110] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun, "MegDet: A large mini-batch object detector," in *CVPR*, 2018.
[111] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.