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 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 Visual representations from Textual 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.

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 meta-data 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 $C = (c_0, c_1, \ldots, c_T, c_{T+1})$ token by token, where $c_0 = [\text{SOS}]$ and $c_{T+1} = [\text{EOS}]$ 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

$$
L(\theta, \phi) = 
\sum_{t=1}^{T+1} \log \left( p(c_t \mid c_{0:t-1}, I; \phi_f, \theta) \right) 
+ \sum_{t=0}^T \log \left( p(c_t \mid c_{t+1:T+1}, I; \phi_b, \theta) \right) 
$$

where $\theta$, $\phi_f$, and $\phi_b$ 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 mas-
is causal – it only depends on past predictions $C_{t−1}$ 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 un-normalized 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 Sentence-Piece [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 $118\times10^3$ 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 ($\approx 1080$ 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 $2 \times 10^{-3}$; 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.
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.

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.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Pretrain Images</th>
<th>Annotations</th>
<th>VOC07</th>
<th>IN-1k</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoCo-IN v1 [24]</td>
<td>1.28M</td>
<td>self-sup.</td>
<td>79.4</td>
<td>60.8</td>
</tr>
<tr>
<td>PCL v1 [57]</td>
<td>1.28M</td>
<td>self-sup.</td>
<td>83.1</td>
<td>61.5</td>
</tr>
<tr>
<td>SwAV (200 ep.) [58]</td>
<td>1.28M</td>
<td>self-sup.</td>
<td>87.9</td>
<td>72.7</td>
</tr>
<tr>
<td>ICMLM&lt;sub&gt;att-tc&lt;/sub&gt; [75] †</td>
<td>118K</td>
<td>captions</td>
<td>87.5</td>
<td>47.9</td>
</tr>
<tr>
<td>VirTex</td>
<td>118K</td>
<td>captions</td>
<td>88.7</td>
<td>53.8</td>
</tr>
</tbody>
</table>

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.

4.2. Ablations

The preceding 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 (a) forward captioning, token classification and masked language modeling.

- **Visual Backbone:** Bigger visual backbones improve downstream performance – both, wider (R-50 w2×) and deeper (R-101).

- **Transformer Size:** Larger transformers (wider and deeper) improve downstream performance.

Figure 5: Ablations. (a) Pretraining Task Ablations: 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.
**4.3. 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-
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 trainval18 split and evaluate on val2018 split. Following MoCo settings, we keep BN parameters frozen for all IN-sup baselines. We train with $2\times$ schedule as COCO, use class resampling and test-time hyperparameters (0.0 score threshold and 500 detections per image) same as [31].

**PASCAL VOC Detection:** We train Faster R-CNN [111] models with ResNet-50-C4 backbones on trainval2007+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 pretrained 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 $7 \times 7$ grid of image features through $A = 8$ heads at each time-step for predicting a token. We average these $7 \times 7$ attention weights over all the heads, and overlay them on $224 \times 224$ 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 Bannani, 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


[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.


[73] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson,
[106] Peter Anderson, Basura Fernando, Mark Johnson, and