CiT: Curation in Training for Effective Vision-Language Data

Hu Xu Saining Xie Po-Yao Huang Licheng Yu
Russell Howes Gargi Ghosh Luke Zettlemoyer Christoph Feichtenhofer
Meta AI, FAIR
https://github.com/facebookresearch/CiT

Abstract

Large vision-language models are generally applicable to many downstream tasks, but come at an exorbitant training cost that only large institutions can afford. This paper trades generality for efficiency and presents Curation in Training (CiT), a simple and efficient vision-text learning algorithm that couples a data objective into training. CiT automatically yields quality data to speed-up contrastive image-text training and alleviates the need for an offline data filtering pipeline, allowing broad data sources (including raw image-text pairs from the web). CiT contains two loops: an outer loop curating the training data and an inner loop consuming the curated training data. The text encoder connects the two loops. Given metadata for tasks of interest, e.g., class names, and a large pool of image-text pairs, CiT alternatively selects relevant training data from the pool by measuring the similarity of their text embeddings and embeddings of the metadata. In our experiments, we observe that CiT can speed up training by over an order of magnitude, especially if the raw data size is large.

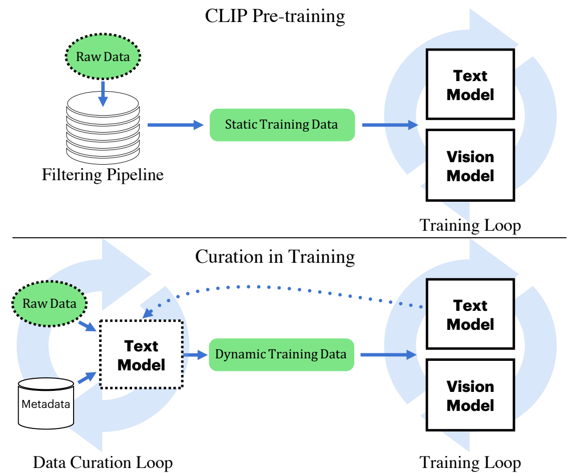

Figure 1: A conceptual illustration of CLIP/LiT training *vs*. CiT. Vanilla CLIP training uses static data from offline human filtering (*e.g*. cleaned YFCC15M or WIT400M [21]) and optimizes the model. Instead, our CiT incorporates dynamic data curation into training in two loops: (*i*) an outer curation loop improving data (for downstream tasks) given the current model; (*ii*) an inner loop optimizing the model given the curated data. The trained text model connects the loops by providing embeddings for curation.

1. Introduction

Vision-language models have demonstrated success for fine-tuning and zero-shot transfer to downstream tasks[21, 12, 26] by training on a general-purpose large-scale dataset instead of a small task-level dataset. While general, large-scale pre-training is computationally expensive (*e.g*. CoCa[36] trains on 2048 TPUs for 5 days) and typically performed on a *pre-filtered* dataset (*e.g*. WIT400M [21] used by CLIP [21] is created by searching for image-text pairs with text containing a set of 500,000 queries from WordNet (includes ImageNet taxonomy) and Wikipedia, and [24] uses this model to create the LAION dataset).

Such filtering pipelines usually involve manual labor-intensive efforts to remove data that is unlikely useful for downstream tasks [12, 21]. Recent effort has been made to curate data for high-quality image-text pairs (such as CC3M[25], CC12M[3], YFCC15M[29, 21], WIT400M[21] and LAION[24, 23]). Nevertheless, research is typically *tied* to the static datasets or model weights (if the data is not released) and is not able to access or change the data pipelines or model architectures. Further, work is *limited* by the prohibitive cost of training on these large image-text datasets (*e.g*. the CLIP model is trained on WIT400M for 12 days using 256 GPUs).

In this work, our goal is to empower training with the capability of adjusting the data distribution. Our intention is to dynamically curate the data during training and our key idea is to use the learned text representation of vision-language models to measure relevance of the data w.r.t. the task of interest. Given metadata (from downstream tasks *e.g*. a class name such as "chicken"), we measure its embedding similarity to the training data. This similarity can guide us for the decision of including this data into our training process. For example a caption containing the word "giraffe" will have higher embedding similarity to "chicken" than a caption such as "throwback Thursday".

Driven by this idea, we presents a simple algorithm that incorporates data Curation in Training (CiT), aiming at improving both data efficiency and model performance. CiT works as follows. Given a large source of image-text pairs and metadata (*e.g*. a list of class names used in this paper), CiT alternatively performs curation of the data and training on that curated data. As shown in Figure 1, CiT contains two loops: an outer loop to curate data given the current model and an inner loop trains the model given the curated data. Similar as Locked image Tuning (LiT [38]), CiT uses pre-trained image and text encoders and freezes the image one. The text model connects the two loops by serving curated data to inner loop for training which in turn learns good representations for the outer loop for curation.

CiT can speed up training by multiple orders of magnitude, especially if the raw data size is large; *e.g*. when trained on LAION-400M data, CiT reaches similar ImageNet zero-shot[1] accuracy as OpenCLIP [31], while being 37.7× faster in training. Since CiT changes the training data distribution that focuses on one or more tasks of interest, it can even handle image-text pairs from any (noisy) source with unknown distribution. Our experiments reveal that vanilla CLIP/LiT training fails on *raw* random image-text pairs crawled from the web, while CiT trains easily.

2. Related Work

Vision-Language Learning. Contrastive learning was initially popular in vision self-supervision[32, 4, 11] and later adopted for cross-modal learning[21, 19, 18, 33, 16]. CLIP[21] populates the idea of contrastive learning from image-text pairs (used before *e.g*. in ConVIRT[40]) at scale and shows a strong performance of zero-shot transfer to image classification and retrieval tasks. SLIP[20] combines image self-supervision and language supervision. LiT[38] shows that when a good pre-trained vision encoder is adopted, it is better to lock (freeze) the well pre-trained vision encoder to protect vision representations from being corrupted by noisy language supervision. Flamingo also use pre-trained models for various tasks[1].

Vision-Language Data. Large-scale vision-language learning is typically coupled to a data pipeline to yield high-quality data for efficient training[26, 37, 12]. For example, CC3M[25] heavily filters web crawled pairs and only keeps 0.1% of the raw data. Both CC3M and CC12M[3] leverage Google Cloud APIs with models predicting a large number of classes (on the order of 10^5)[25] to filter out mismatched image-text pairs. YFCC100M[29] is curated from Yahoo Flicker using text fields (such as title, description, etc.). This ensures certain data quality but limits the scale. Later YFCC100M is further cleaned as YFCC15M to contain English-only image-text pairs by [21]. Due to the limited scale, CLIP further curates a WebImageText dataset (WIT400M) by formulating queries from Wikipedia and WordNet synsets (including ImageNet class names) then searches image-text pairs with texts containing those queries. Florence[37] curates a dataset with the extra multi-label signals to improve supervision. ALIGN[12] relaxes CC12M filtering to show that training on 1.8B noisy pairs can achieve CLIP-level performance. FLAVA[26] combines existing human annotated datasets of smaller scale for high-quality image-text pairs. Different to related research, CiT improves data *within* the training algorithm, and not as a pre-filtering. We demonstrate that such approach allows us to effectively learn from *raw* image-text pairs.

Related Areas. Our work is related to research in other domains. In NLP, there are existing works on domain-adaptive finetuning and retrieval [34, 39, 9, 15, 14, 35]. In machine learning research, subset selection [30, 13] cast data selection as a discrete bi-level optimization problem.

3. Method

In CLIP pre-training, the objective (contrastive image-text correspondence) operates as a training proxy that approximates downstream tasks (*e.g*. classification accuracy). Our CiT introduces a *data proxy* to fit the *data distribution* to downstream tasks. In this section, we first go through the details of the CiT algorithm in §3.1, training loop in §3.2 and the data proxy for the curation loop in §3.3.

3.1. CiT Algorithm

CiT contains two loops: the curation loop curates data given the current weights of the model and the training loop optimizing the weights given the curated data.

Let $\mathcal{D} = \{(x^i_{\text{img}}, x^i_{\text{txt}})\}^N_{i=1}$, be the set of source of image-text pairs. Then $\mathcal{D}_C \subseteq \mathcal{D}$ is the actual *training data* we aim to curate from the source. We define two functions: (i) *Curation*$(\mathcal{D}; \Theta)$, and (ii) *Training*$(\Theta; \mathcal{D}_T)$, for curation and training loops, respectively. Importantly, the weights of the learned model Θ connects the two loops and serves the curation loop with the updated representations from the training loop. CiT uses a sequential setup that alternatively performs curation for every s pairs of training.

CiT is shown in Algorithm 1. It takes 3 inputs: a data source $\mathcal{D}$, the pre-trained weights Θ and a training budget b, which can be training time, resources consumed, *etc*. We simply use steps of weight updates as the training cost in this paper. Line 1 initializes the training budget. Line 2 determines if current training exceeds that training budget.

[1]Zero-shot refers to not seeing any training examples of the target dataset. We note that our approach uses extra information of the downstream task, such as class names; however, this metadata is easy to acquire and can be of various forms as shown in experiments.

Algorithm 1: CiT (see Suppl. for pseudo code)

Input: $\mathcal{D}$: data source
Θ: model's pre-trained weights
b: training budget

1 $c \leftarrow 0$
2 **while** $c < b$ **do**
3 $\quad \mathcal{D}_T \leftarrow \mathit{Curation}(\mathcal{D}; \Theta)$
4 $\quad \Theta, n \leftarrow \mathit{Training}(\Theta; \mathcal{D}_T)$
5 $\quad c \leftarrow c + n$
6 **end**

Algorithm 2: Curation

Input : Θ: model's current weights
$\mathcal{D}$: data source
Constant: $m(\cdot;\cdot)$: model architecture
$\mathcal{T}_{\text{meta}}$: metadata for tasks of interests
s: number of expected pairs

1 $\mathbf{x}_{\text{meta}} \leftarrow m(\mathcal{T}_{\text{meta}}; \Theta)$
2 $\mathcal{D}_C \leftarrow \varnothing$
3 **while** $|\mathcal{D}_C| < s$ *and* $\mathcal{D}_{raw} \subset \mathcal{D}$ **do**
4 $\quad \mathcal{D}_{\text{raw,txt}} \leftarrow \{x^i_{\text{txt}} | (x^i_{\text{img}}, x^i_{\text{txt}}) \in \mathcal{D}_{\text{raw}}\}$
5 $\quad \mathbf{x}_{\text{txt}} \leftarrow m(\mathcal{D}_{\text{raw,txt}}; \Theta)$
6 $\quad f \leftarrow \mathit{DataProxy}(\mathcal{D}_{\text{raw}}, \mathbf{x}_{\text{txt}}, \mathbf{x}_{\text{meta}})$
7 $\quad \mathcal{D}_C \leftarrow f(\mathcal{D}_{\text{raw}}; \Theta, \mathcal{D}_{\text{raw}}, \mathcal{T}_{\text{meta}}) \cup \mathcal{D}_C$
8 **end**
9 **return** $\mathcal{D}_C$

The main framework of CiT is to alternatively perform curation and training in line 2-4. To recap CLIP pre-training, we first detail the training function next.

3.2. Training

The core of CLIP [21] training is the contrastive cross-modal objective serving as the proxy that approximates downstream tasks (*e.g.* higher classification accuracy). This objective pulls embeddings of positive image-text pairs closer and pushes negative pairs from other examples in a training batch apart; thus it creates a proxy for classification, which has one example per class and the rest of the batch are other classes described by natural language.

The training loop is shown in Algorithm 3, with the training data $\mathcal{D}_C$, delivered from curation. We let $m(\cdot;\cdot)$ denote the image-text model. We use $\text{sim}(\mathbf{x}_{\text{img}}, \mathbf{x}_{\text{txt}}) = \mathbf{x}_{\text{img}}\mathbf{x}^{\top}_{\text{txt}}/(\|\mathbf{x}_{\text{img}}\|\|\mathbf{x}_{\text{txt}}\|)$ in line 3 to compute the image-to-text cosine similarity, divided by a trainable temperature τ. Our CiT training objective has almost the same structure as in CLIP, except that we only use an image-to-text (and no text-to-image) contrastive loss ($\mathcal{L}_{\text{img2txt}}$) in line 4. We ablate this loss versus the averaged bidirectional contrastive loss (used by CLIP) in our experiments. Line 5 updates the model parameters and line 6 counts training cost.

3.3. Curation

CiT also has a data objective that curates data using the (previously updated) model. Encoding the data with an updated model allows for better representation of the data. Akin to the contrastive objective for training, the core function in curation is a *data proxy* (or objective) that selects data based on the metadata (*e.g.* a list of class names).

Algorithm 3: Training

Input : $\mathcal{D}_C$: curated training data
Θ: model's weights
Constant: $m(\cdot;\cdot)$: model architecture

1 **foreach** $\mathcal{D}_{batch} \subset \mathcal{D}_C$ **do**
2 $\quad \mathbf{x}_{\text{img}}, \mathbf{x}_{\text{txt}} \leftarrow m(\mathcal{D}_{\text{batch}}; \Theta)$
3 $\quad l \leftarrow \text{sim}(\mathbf{x}_{\text{img}}, \mathbf{x}_{\text{txt}})/\tau$
4 $\quad \mathcal{L}_{\text{img2txt}} \leftarrow \mathit{CrossEntropy}(l, \mathit{arange}(|\mathcal{D}_{\text{batch}}|))$
5 $\quad \Theta \leftarrow \mathcal{L}_{\text{img2txt}}(\Theta; \mathcal{D}_{\text{batch}})$
6 $\quad n \leftarrow n + 1$
7 **end**
8 **return** Θ, n

We detail the curation loop in Algorithm 2. It takes the following inputs: model weights Θ, a data source $\mathcal{D}$, the model architecture, the metadata for downstream tasks $\mathcal{T}_{\text{meta}}$ and an expected size of curated data s. $\mathcal{T}_{\text{meta}}$ is a list containing a pre-defined taxonomy; (*e.g.* ImageNet WordNet lemmas or a combination from a group of tasks in our experiments), but could be generalized to other forms of text.

Algorithm 2 first obtains the embeddings for the metadata in line 1. Then it sets up the curated set $\mathcal{D}_C$ for the next round of training and keeps curating data in line 3-7. Line 3 gets the next batch of raw image-text pairs. Line 4 obtains its text part and line 5 computes the text embedding from the current model. Line 6 is the *data proxy*, which approximates the data distribution for the downstream tasks (detailed in the next subsection). Lastly, we merge the newly curated subset into the curated set $\mathcal{D}_C$.

Data Proxy. We use language-based metadata and the text encoder to measure the relevance of training data. This favors efficiency because the text encoders are typically *significantly cheaper* to evaluate (*e.g.* the text encoder only uses ~4.6% of the ViT-L image-encoders' compute).

In $\mathit{DataProxy}(\mathcal{D}_{\text{raw}}, \mathbf{x}_{\text{txt}}, \mathbf{x}_{\text{meta}})$ of Algorithm 2, we first compute the similarities of text embeddings ($\mathbf{x}_{\text{txt}}$) over embeddings of the metadata ($\mathbf{x}_{\text{meta}}$):

$$v^i_{\text{max}} = \max_j(\text{sim}(\mathbf{x}^i_{\text{txt}}, \mathbf{x}^j_{\text{meta}})), \quad (1)$$

where $\text{sim}(\mathbf{x}^i_{\text{txt}}, \mathbf{x}^j_{\text{meta}}) = \mathbf{x}^i_{\text{txt}}\mathbf{x}^{j,\top}_{\text{meta}}/(\|\mathbf{x}^i_{\text{txt}}\|\|\mathbf{x}^j_{\text{meta}}\|)$ is the cosine similarity between embeddings of sample i and metadata j. Here the highest similarity over all metadata v^i_{max} is used to measure the sample quality.

Let $\mathcal{D}_t = \{(x^i_{\text{img}}, x^i_{\text{txt}}) | (x^i_{\text{img}}, x^i_{\text{txt}}) \in \mathcal{D}_{\text{raw}} \text{ and } v^i_{\text{max}} > t\}$ denote a subset, where all samples have a maximum similarity above a curation threshold t. Given the best possible match to metadata, we use a mixed strategy to determine if a sample shall be used:

$$\begin{cases} \mathcal{D}_t & \text{if } \frac{|\mathcal{D}_t|}{|\mathcal{D}_{\text{raw}}|} > \gamma, \\ \arg\operatorname{topk}_i(v^i_{\text{max}}, k = \gamma|\mathcal{D}_{\text{raw}}|), & \text{otherwise}, \end{cases} \quad (2)$$

where $\frac{|\mathcal{D}_t|}{|\mathcal{D}_{\text{raw}}|}$ is the *ratio of curation* with γ being a pre-defined *minimal* ratio of curation. If enough samples meet the threshold t, $\mathcal{D}_t$ is used. Otherwise, we use a *minimal ratio* γ of samples, that represent the top-k matching ones (with $k = \gamma|\mathcal{D}_{\text{raw}}|$) in terms of similarity across metadata.

The threshold t is crucial for CiT to balance the tradeoff between data quality and quantity. A higher t leads to high data quality, but can lead a lower ratio of curation. We adopt this mixed strategy because line 3 in Algorithm 2 could become a near infinite loop if the ratio of curation is low and not enough data that meets t can be found. This could happen because the threshold is set too high, or the data source has low metadata correspondence. The *otherwise* part in equation 2 resolves this by selecting the γ (typically set to around 1% - 5%) best possible matches for training. See supplementary material for PyTorch pseudo code of CiT.

4. Experiments

We use training data from two categories shown below; clean data that involves human-based offline filter pipelines and raw data that has not undergone cleaning.

4.1. Cleaned Training Data

YFCC15M. We use the 15M subset of YFCC100M[29] (filtered by [21]) as the main evaluation dataset as it is widely adopted in existing literatures[21, 20, 38, 22]. It consists of English-only titles, descriptions, and tags. We simply refer to this as YFCC15M in this paper. Except for applying the script from [20] to remove HTML formatting, we do not perform any extra filtering or preprocessing. In contrast, LiT[38] performs extra filtering such as removing titles that start with "DSC", "IMG" and "Picture", or removing them if more than half of them contain digits.

CC12M. Since YFCC15M may lack enough training data, LiT[38] also combines YFCC15M with Conceptual Captions 12M (CC12M) [3], which is filtered and transformed from image & alt-text pairs from web pages. CC12M involves cleaning by supervised models from Google Cloud APIs to match the image's prediction over classes with text.

LAION400M [24] contains 400M English only image-text pairs. It is crawled from [2] and later filtered by a CLIP[21] model. Thus, LAION400M implicitly carries the data filter pipeline of WIT400M on which CLIP has been trained.

[2] https://commoncrawl.org

4.2. Raw Training Data

YFCC100M. We use the raw YFCC100M (the source of YFCC15M) to compare with YFCC15M. Note that YFCC100M is multilingual, whereas YFCC15M is English.

Raw Image-Text Crawl. To challenge CiT with real-world data, we further collect raw (unfiltered) image-text pairs from Common Crawl. We only perform de-duplication and NSFW filtering, but *no* filtering on image-text association. This ended with 1.2B multilingual image-text pairs and 28.56% pairs are English (identified by our language identification system but this is not used in CiT). As such, ~343M image-text pairs are English, which is slightly less than WIT400M or LAION400M, and much more noisy.

4.3. Implementation and Training

Our training recipe uses a global batch size of 16,384, which is trained in 16 Nvidia V100 32GB GPUs. Our vision encoder corresponds to ViT [7] of various sizes and the text encoder defaults to $\text{BERT}_{\text{base}}$-SimCSE [6, 8] with a maximum token length of 32, similar to LiT [38]. Unless specified, we set a budget of training to be within $b = 5000$ steps (81M image-text pairs). We report hyper-parameters and an extra low-cost single-GPU setting in supplement.

We use pre-trained vision and text encoders and join them via two randomly initialized projection layers. Following LiT, we freeze the vision encoder and make the text encoder and two projection layers trainable. One can either use the text representation *before*, or *after* the projection layer for computing cosine similarity during curation. We ablate these two choices in §4.6.

4.4. Evaluation

We evaluate zero-shot (0-shot) transfer *accuracy* of CiT on **26** benchmarks, following [21, 20]. In our ablation studies, we use YFCC15M as the main data source for training and ImageNet-1K (IN-1K) as the downstream task. We use prompts from SLIP for all 26 tasks and additionally use the extra 2 prompts from LiT[38] for ImageNet for a fair comparison with LiT. Following CLIP, we perform prompt ensembling by averaging the class embeddings for each class across the prompt templates. For classification, cosine similarity is computed between an image embedding and the averaged class embeddings and the class with the highest cosine similarity is CiT's prediction. We perform validation every 500 training steps and stop training if the accuracy does not increase over the previous validation. The corresponding total training time (including curation and training) is reported along with the validation accuracy. We estimate the training time of baselines by re-running them under the same setup as CiT (*i.e.* 16 GPUs) and maximize the GPU usage for best throughput. More results are in the supplementary material.

Curation	Acc
online	**61.4**±0.2
offline	57.5
no	53.8

(a) Curation effect

# of steps	Acc
50	61.0
100	61.4
200	**61.5**
300	61.1

(b) Curation freq.

Feature of Curation	Acc
pooled encoder	**61.4**
projection output	60.7
w/ prompts	61.4

(c) Curation feature

Threshold t	Acc
0.5	60.9
$t = 0.55$	**61.4**
0.6	61.1
0.7	59.7

(d) Threshold t

Text Variants	Acc
BERT max len. 32	**61.4**
BERT max len. 77	61.2
w/o YFCC tag	59.1
w/o YFCC tag aug.	60.8
BERT first 6 layers	60.2

(e) Text variants

Table 1: **Ablations**. We use MoCo-v3/BERT$_{\text{base}}$-SimCSE, YFCC15M as data source and report IN-1K Accuracy(3 runs for default).

4.5. Choice of Pre-trained Models

We first study the effects of encoders. We utilize publicly available pre-trained encoders, following LiT[38].

As *vision encoder*, we consider (1) ViT-B/16 [7] (patch size of 16×16 pixels) with pre-trained weights from self-supervised MoCo-v3 [5], DINO[2] and MAE [10], all trained on IN-1K but without any labels. To be consistent with LiT[38], we also consider (2) supervised ViT(AugReg)[28] B/32, B/16, and L/16 trained on ImageNet-21K[3]. Finally, we also explore weakly-supervised ViT-B/16, ViT-L/16 and ViT-H/14 SWAG[27].

Vision Model	Pre-train Obj.	Pre-train Data	IN-1K Acc.
MoCo-v3[5]	Contrastive	IN-1K	61.4
DINO[2]	Contrastive	IN-1K	60.3
MAE[10]	Masking	IN-1K	42.4
AugReg[28]	Supervised	IN-21K	**69.4**
SWAG[27]	Weakly-Supervised	IG 3.6B	67.5

Table 2: Ablation study of different vision encoders on ViT-B/16 with text encoder as BERT$_{\text{base}}$-SimCSE on YFCC15M. Pre-training objective matters for CiT training.

Results for different vision encoder weights under the same ViT-B/16 architecture are in Table 2. We notice that the accuracy of MoCo-v3 (61.4%) and DINO (60.3%) pre-training are close, while MAE is worse (42.4%), presumably because the representations learned by instance discrimination (MoCo-v3 and DINO), which learns different embeddings for different images, is closer to zero-shot classification than MAE's training objective. AugReg performs best with 69.4% accuracy, presumably because the supervised pre-training on IN-21K is superior to unsupervised IN-1K pre-training. Finally, SWAG is worse than AugReg, but better than MoCo-v3. In the following experiments of this section, we will show larger variants.

For *text encoder*, we consider self-supervised base models from (1) language models BERT [6]; and contrastive tuned (2) BERT-SimCSE and RoBERTa-SimCSE [8], as shown in Table 3.

Text Model	Pre-training obj.	IN-1K Acc.
BERT$_{\text{base}}$(uncased)[6]	from scratch	57.7
BERT$_{\text{base}}$(uncased)[6]	SimCSE[8]	**61.4**
BERT$_{\text{base}}$(uncased)[6]	BERT NSP[6]	59.9
RoBERTa$_{\text{base}}$[17]	SimCSE[8]	59.7

Table 3: Ablation of text encoders with MoCo-v3 on YFCC15M: contrastive pre-training yields better accuracy.

[3]We follow LiT here, but note that using IN-21K is not strictly a zero-shot setting, because 999 of the 1000 classes in IN-1K are in IN-21K.

We observe similar trends as for vision: SimCSE trained BERT is better than vanilla BERT or RoBERTa, probably because contrastively trained `[CLS]` token by SimCSE can perform better text similarity than BERT's pairwise (a.k.a, next sentence prediction) trained `[CLS]` token or RoBERTa's no training on `[CLS]` token.

4.6. Ablations

We adopt the combination of MoCo-v3 ViT B/16 and BERT-SimCSE as our default setting. We summarize ablation experiments of CiT in Table 1.

Stage of Curation. We first ablate the effects of curation in Table 1a. We see that CiT has a **7.6%** boost compared to *no curation*. We further ablate an *offline* curation before training. This is sub-optimal as the SimCSE purely pre-trained from the text may not learn good representations for semantic-level similarity (discussion in §3.1).

Frequency of Curation. Next, we are interested in how frequently curation needs to be performed. Table 1b varies the number of steps (and therefore pairs s when multiplied with the batch-size) for curation (in Alg. 2). We found that curating too frequent or infrequent yields sub-optimal results, but the change is marginal so we chose 100 steps as default.

Feature for Curation. In Table 1c, we find that using the feature before the projection layer (*e.g.* the direct output of SimCSE) is better than the features from the projection layer. This is probably because the projection layer tends to be more unstable during training (*e.g.* randomly initialized and needs longer training to align with the visual representation), whereas the SimCSE embedding is already pre-trained for text similarity.

Threshold. In Table 1d we ablate the threshold t, which controls the trade-off for data quality and quantity. A lower threshold adds more low-quality data and a higher threshold reduces data quantity, so $t = 0.55$ is a good balance.

Text Variants. We ablate the length of text encoders in Table 1e to understand the memory/text sequence length tradeoff. We find that longer text sequences (77) (we reduce batch size per GPU to half and double the number of GPUs) are slightly worse. We also ablate the effectiveness of YFCC15M tag augmentation, adopted from LiT. Lastly, we are wondering if a shallow (6 layers) BERT-SimCSE is also a good text encoder. We obtain 1.2% worse results.

Pre-train Data	Method	Vision Encoder	Vision Initialization	w/ Labels	Total Time	IN-1K Acc
YFCC15M	LiT[38]	ViT-B/16	DINO[2]	✗	n/a	55.5
	CiT	ViT-B/16	DINO[2]	✗	11 hrs	**60.3**
	CLIP[21]	ViT-B/16	MoCo-v3[5]	✗	48 hrs	54.6
	LiT[38]	ViT-B/16	MoCo-v3[5]	✗	n/a	55.4
	CiT	ViT-B/16	MoCo-v3[5]	✗	5 hrs	**61.4**
	LiT[38]	ViT-B/32	AugReg[28]	IN-21K	64 hrs	59.9*
	CiT	ViT-B/32	AugReg[28]	IN-21K	11 hrs	**63.3**
	LiT[38]	ViT-B/16	AugReg[28]	IN-21K	n/a	55.9
	CiT	ViT-B/16	AugReg[28]	IN-21K	8 hrs	**69.4**
	CiT	ViT-L/16	AugReg[28]	IN-21K	8 hrs	**72.0**
	CiT	ViT-L/16	SWAG[27]	IG hashtags	8 hrs	**73.0**
	CiT	ViT-H/14	SWAG[27]	IG hashtags	11 hrs	**73.7**
YFCC15M+CC12M	LiT[38]	ViT-L/16	AugReg[28]	IN-21K	112 hrs	72.2*
	CiT	ViT-L/16	AugReg[28]	IN-21K	32 hrs	**75.6**
YFCC100M	LiT[38]	ViT-B/32	AugReg[28]	IN-21K	153 hrs	58.9*
	CiT	ViT-B/32	AugReg[28]	IN-21K	64 hrs	**65.6**
	CiT	ViT-B/16	MoCo-v3[5]	✗	48 hrs	**64.6**
	CiT	ViT-B/16	AugReg[28]	IN-21K	66 hrs	**72.2**
	CiT	ViT-L/16	AugReg[28]	IN-21K	66 hrs	**74.8**
	CiT	ViT-L/16	SWAG[27]	IG hashtags	62 hrs	**74.8**
	CiT	ViT-H/14	SWAG[27]	IG hashtags	62 hrs	**75.5**

Table 4: Comparison to LiT on YFCC and CC12M. Under identical vision encoders, CiT achieves +3.2% higher accuracy with YFCC100M than using the human-curated YFCC15M subset and +5.9% accuracy over LiT on YFCC15M. * indicates reproduced results with $\text{BERT}_{\text{base}}$ (uncased) for fair comparison; see supplementary for the implementation differences to original LiT [38]. Total time for training and curation is reported for 16 V100 GPUs and varies depending on quality of embeddings from the vision encoder.

Method	Vision Encoder	Vision Initialization	w/ Labeled Data	Total Time	IN-1K Acc
OpenCLIP	ViT-B/32	scratch	✗	458 hrs	62.9
OpenCLIP	ViT-B/16	scratch	✗	981 hrs	67.1
OpenCLIP	ViT-L/14	scratch	✗	6803 hrs	72.8
CiT	ViT-B/16	MoCo-v3[5]	✗	26 hrs	**67.1**
CiT	ViT-B/16	SWAG[27]	IG hashtags	14 hrs	**70.9**
CiT	ViT-B/16	AugReg[28]	IN-21K	63 hrs	**73.1**
CiT	ViT-L/16	AugReg[28]	IN-21K	27 hrs	**75.8**
CiT	ViT-H/14	SWAG[27]	IG hashtags	26 hrs	**76.4**

Table 5: CiT on LAION400M: CiT reaches OpenCLIP-level accuracy with 37× total training time improvement.

4.7. Comparison to prior work on ImageNet

We compare CiT with existing contrastive cross-modal models in Tables 4 (YFCC and CC12M), 5 (LAION400M) and 6 (raw image-text crawl). We report the pre-training method (CLIP/LiT/CiT), vision encoder and initialization, usage of human-annotated labels, total training time in our setup (16 GPUs), as well as the ImageNet 0-shot accuracy.

YFCC. In Table 4 we report several data points for LiT and CiT training with various vision encoders and initialization. On YFCC15M, CiT outperforms LiT on self-supervised MoCo-v3 vision encoders by +5.9% accuracy and CLIP training by 6.8%. With ViT-B/32 trained with supervised AugReg on IN-21K, CiT yields a +3.4% gain over LiT. CiT is much faster than LiT (by 5.8×) and CLIP (by 9.6×).

On YFCC15M+CC12M data with ViT-L/16 models, CiT outperforms LiT by +3.4% using the same backbone.

On YFCC100M we observe that LiT underperforms compared to YFCC15M (58.9 vs 59.9), due to cleaning [21] of the 15M subset. CiT however can *reverse* the trend. CiT outperforms its counterpart from YFCC15M by 3%+ when using the less curated YFCC100M. This indicates the human cleaning of YFCC100M in [21] is sub-optimal. The performance of CiT on YFCC100M is even **+2.6%** better than LiT on YFCC15M+CC12M. This trend holds for larger image model sizes (ViT-L/H) and stronger initialization (AugReg/SWAG), which lead to better accuracy.

LAION400M. In Table 5 we see that CiT performs better than OpenCLIP on LAION400M, while being substantially faster. For example, CiT with ViT-B/16 MoCo-v3 vision encoder performs as good as OpenCLIP but is 37.7×faster in training. With more advanced initialization and larger ViT-L models, CiT is 283× faster and 3% more accurate, producing 75.8% in 1.1 days with a 16 GPU setup, while OpenCLIP would take ~283 days for an accuracy of 72.8%. Specifically, CiT only uses 26 hours for training, compared to 981 hours for OpenCLIP pre-training. We note that this *extreme* speedup comes with the caveat that CiT *curates data online* with respect to downstream tasks and uses a pre-trained encoder; therefore, this comparison is not 100% fair, but shows a low-cost alternative for the community.

Method	Vision Encoder	Vision Initialization	w/ Labeled Data	Total Time	IN-1K Acc.
OpenCLIP	ViT-B/16	from scratch	✗	n/a	NaN loss
LiT	ViT-B/16	MoCo-v3[5]	✗	n/a	NaN loss
LiT (English filter)	ViT-B/16	MoCo-v3[5]	✗	65 hrs	56.7
CiT	ViT-B/16	MoCo-v3[5]	✗	39 hrs	**68.7**
CiT	ViT-B/16	AugReg[28]	IN-21K	72 hrs	**75.2**
CiT	ViT-L/16	AugReg[28]	IN-21K	105 hrs	**77.9**
CiT	ViT-H/14	SWAG[27]	IG hashtags	43 hrs	**77.4**

Table 6: CiT on Raw Image-Text Crawl: CiT produces strong results when learning from raw data, containing 1.2B image-text pairs. An English language filter, which reduces the data to 343M pairs, is required to stabilize LiT training.

	Time	Food-101	CIFAR10	CIFAR100	CUB	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	MNIST	FER-2013	STL-10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMemes	SST2	ImageNet	Avg
CLIP [21, 20]	27	50.6	66.0	34.5	38.8	51.1	4.0	5.4	21.2	28.5	60.9	53.3	8.4	17.3	90.5	30.2	21.5	6.1	35.1	10.5	53.5	28.5	22.1	10.8	52.4	50.7	37.6	34.2
SLIP [20]	41	59.5	78.6	45.2	38.7	53.4	5.4	5.7	26.1	31.1	71.0	56.6	9.8	19.6	94.4	20.3	28.9	14.5	34.0	11.6	55.4	37.7	26.9	17.5	52.8	51.1	42.8	38.0
CiT-1K-meta	5	45.6	81.0	49.9	30.4	44.9	6.3	8.3	26.8	**80.0**	71.2	25.1	7.3	26.0	95.2	19.1	14.3	6.9	22.2	6.2	**54.1**	34.7	24.7	13.4	50.7	50.1	**61.2**	38.5
CiT-21K-meta	15	51.2	**84.4**	53.5	45.7	**52.3**	7.6	9.0	31.6	69.2	73.8	56.1	10.6	24.5	95.7	**30.1**	**23.4**	**7.9**	28.5	9.2	51.0	39.5	**28.7**	**15.0**	49.3	49.1	57.4	40.6
CiT-multi-meta	11	51.3	81.8	50.5	50.7	51.6	9.5	**14.6**	30.8	75.6	73.3	58.7	10.3	**26.2**	95.6	23.2	19.1	7.8	14.6	**9.4**	50.8	39.7	28.0	14.7	**52.8**	50.0	58.8	40.4
CiT-sep.-meta	11	**59.1**	82.2	**55.2**	**56.6**	50.7	**13.0**	13.1	**32.8**	74.8	**77.6**	**65.9**	**16.9**	13.8	**96.3**	17.1	21.6	7.6	**40.6**	**9.4**	53.5	**42.7**	27.8	14.2	52.2	**50.9**	50.7	**42.2**

Table 7: CiT on 26 zero-shot benchmarks when trained on YFCC15M. We vary metadata from IN-1K, IN-21K, combined (multi) and separate (sep.). All methods use ViT-B/16 and we use MoCo-v3 vision initialization. Larger encoders in Supp.

Raw Image-Text Crawl. We further test CiT on our raw image-text crawl containing 1.2B unfiltered image-text pairs from the web (about 343M pairs have English text). The data contains a large degree of noise. Results are shown in Table 6. To understand the challenge of training on raw image-text pairs, we run CLIP and LiT training on the raw image-text pairs. This yields unstable training that quickly reaches NaN loss for both a CLIP and LiT training. We believe some noisy pairs are unhealthy for training. By using our English filter to clean the text, we can train LiT and it reaches 56.7% IN-1K zero-shot accuracy. Training our CiT (*without* even using an English filter) achieves 68.7% which is **+12.0**% higher. This indicates raw and very noisy image-text pairs lead to poor accuracy, but CiT can overcome this and curate high-quality data for vision-language learning.

Surprisingly, as shown in Table 5, CiT achieves much better performance than OpenCLIP trained on LAION400M. CiT on raw image-text reaches **77.9**%, which is +5.1% better than OpenCLIP ViT-L/14 (*c.f.* Table 5). Note that our source is raw, with multilingual texts, whereas LAION400M is a curated English-only dataset filtered by the CLIP model. The training data used by CiT (*e.g.* 131M for 77.9%) is just around 1/5 of the scale of LAION400M dataset (one epoch), showing the effectiveness of curating training data.

4.8. Comparison across 26 benchmarks

We extend CiT to 26 common 0-shot evaluation tasks for CLIP/SLIP models [20] on the public dataset YFCC15M. We provide more comparisons with further encoders as well as pre-training on LAION400M in the appendix. We evaluate with prompts from CLIP/SLIP. For ImageNet, we drop the extra prompts used by LiT for a fair comparison with the baselines. We use three setups of metadata: (*i*) IN-1K, (*ii*) IN-21K, and (*iii*) multi-task CiT that combines class names from all 26 tasks (*iv*) we run every task separately on a *single* GPU as a low-compute setup (this trains a model for each task with separate metadata). Results are in Table 7 and discussed next.

We first evaluate CiT trained with IN-1K metadata on all 26 tasks. As expected accuracy on ImageNet and Pets is highest among the metadata variants (*i-iv*). Overall, we observe that *CiT 1K meta* already exhibits certain generality to all tasks and can outperform CLIP (34.2 *vs.* 38.5%) and is similar to SLIP, but 8.2× faster (5 *vs.* 41 hours), demonstrating its efficiency.

Next, we explore the WordNet lemma from ImageNet-21K as a relatively general metadata for training CiT. In Table 7, *CiT-21K-meta* improves broadly over IN-1K leading to 40.6% average accuracy, showing that a more general taxonomy works well across tasks.

We combine the taxonomies from all 26 tasks in *CiT-multi-meta*. This allows us to curate training data for all 26 tasks at again almost no extra training cost. We notice that multi-task CiT is on average similarly accurate as IN-21K metadata (40.4% *vs.* 40.6%) and converges faster because CiT is more targeted towards tasks of interest.

Finally, we compare a setup that trains a model for each task with separate metadata. *CiT-sep.-meta* in Table 7 achieves overall the best average accuracy of 42.2% across tasks. This setup uses a restricted 1-GPU setting to save compute and could be boosted further with longer training. We think that this scenario might be quite practical, where some domain data exists (*e.g.* on bird images in CUB) and one wants to build a classification system given a large amount of noisy image-text data from the web.

Step (c)	Text	ImageNet Class	Cosine Sim.
0	title: *“Wollaston Beach”*	beach	0.739
100	title: *“tn_kif_3128”*	Vizsla	0.779
1000	tag: *“beach plumisland parker river national wildlife refuge newburyport massachusetts ocean”*	beach	0.716
2000	desc: *“These guys were nice, told me all about this and other planes of the show, but unfortunately...”*	military aircraft	0.725
3000	title: *“Turtle”*	terrapin	0.725
4000	desc: *“One of the fountains close by the south west entrance to the park”*	fountain	0.734
5000	title: *“butterfly”*	Papillon	0.735
5000	tag: *“ash;explosion;sakurajima;kagoshima;桜島;鹿児島県;volcano;tarumizu;垂水市;japan;eruption;日本”*	volcano	0.645

Table 8: Samples of curated text over training steps (c) from YFCC100M. CiT uses MoCo-v3 initialized vision encoder.

4.9. Further Analysis

Samples of Curated Data. We further investigate samples curated by CiT on YFCC100M dataset in Table 8. We show training steps, a sample text, the related ImageNet metadata, as well as the cosine similarity in CiT’s data proxy. At step $c = 0$ CiT’s *data proxy* tends to select text with similar length as class names and string-matching behavior; the short-term run of CiT (*e.g.* $c = 100$) has some matching issues with many false positives. Later on, CiT starts to select texts of various lengths with similar semantics as the metadata. We do not observe any clearly less useful samples such as file names after $c = 2000$. Interestingly, CiT can even use the English part of mixed language texts from YFCC100M (as in the last example).

Speed/accuracy trade-off. In Figure 2, we show the speed/accuracy tradeoff of CiT *vs*. LiT [38], corresponding to results in Table 4). We see that CiT achieves a win-win scenario compared to LiT on identical AugReg ViT-B/32 vision encoders: a *+3.4%* higher accuracy on ImageNet, and a *5*× faster total training time (including the curation time). on data YFCC15M [21].

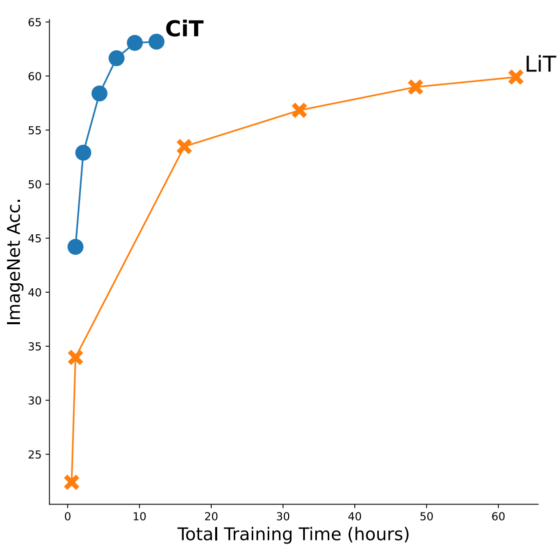

Figure 2: CiT on provides >5× speedup and +3.4% accuracy over LiT[38] on AugReg ViT-B/32 vision encoders. Training data is YFCC15M. Models are evaluated at 6 evenly sampled iterations.

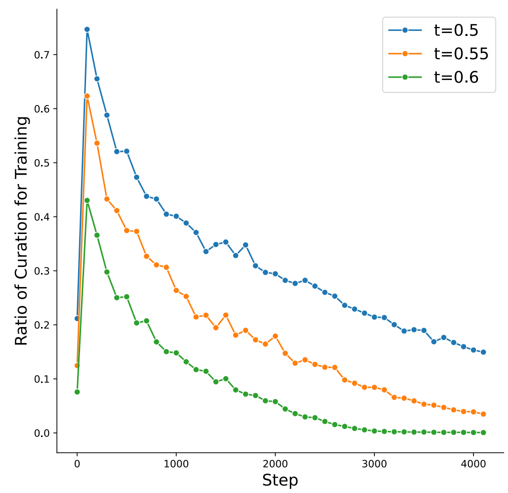

Figure 3: Ratio of curation under different thresholds t. CiT broadly uses data first and curates more towards end of training.

Ratio of Curation. We are interested in the training dynamics of CiT. We use different curation thresholds t and inspect the amount of curated training data. In Figure 3, we see that the ratio of curation which corresponds to the fraction of used training samples from the raw data source, see §3.3, keeps changing over steps for curation/training. Initially, CiT uses more data, *e.g.* for a threshold of $t = 0.5$, it peaks at about 75%. In this phase, the latent space of the text encoder is less aligned with the vision latents. Later on during training, CiT starts to produce embeddings that better represent the downstream task, producing a lower ratio.

5. Conclusion

This paper contributes CiT, a novel learning algorithm for efficient pre-training from noisy image-text data. CiT incorporates a curation process into learning to pull the training data distribution closer to downstream tasks. Our experiments demonstrate both significant accuracy and training time improvements when learning from either public or our own uncurated data from the web. We observe that training on the raw image-text pairs in YFCC can achieve better accuracy over the cleaned version from a hand-crafted filter pipeline. Further, we show that CiT can train with raw image-text pairs crawled from the web, which would lead to instability for vanilla pre-training objectives.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[3] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[8] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, 2021.

[9] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[12] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[13] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8110–8118, 2021.

[14] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

[15] Junlong Li, Zhuosheng Zhang, Hai Zhao, Xi Zhou, and Xiang Zhou. Task-specific objectives of pre-trained language models for dialogue adaptation. *arXiv preprint arXiv:2009.04984*, 2020.

[16] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021.

[17] 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.

[18] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[19] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE international conference on computer vision*, pages 2630–2640, 2019.

[20] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. *arXiv preprint arXiv:2112.12750*, 2021.

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[22] Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. Is a caption worth a thousand images? a controlled study for representation learning. *arXiv preprint arXiv:2207.07635*, 2022.

[23] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, et al. Laion-5b: An open large-scale dataset for training next generation image-text models.

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

[25] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[26] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. *arXiv preprint arXiv:2112.04482*, 2021.

[27] Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens van der Maaten. Revisiting weakly supervised pre-training of visual perception models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 804–814, 2022.

[28] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv preprint arXiv:2106.10270*, 2021.

[29] 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*, 59(2):64–73, 2016.

[30] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR, 2015.

[31] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022.

[32] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[33] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, 2021.

[34] Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. Bert post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, 2019.

[35] Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. Pre-trained transformers for text ranking: Bert and beyond. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 1154–1156, 2021.

[36] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[37] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[38] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022.

[39] Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. Curriculum learning for domain adaptation in neural machine translation. *arXiv preprint arXiv:1905.05816*, 2019.

[40] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.