Annotation Bootstrapping: A Self-Reinforcing Approach to Visual Pre-Training

Anonymous Authors[1]

Abstract

Despite the abundance of unlabeled images in the wild, scalable visual pre-training on raw image data remains a challenge. Generic recipes like pixel reconstruction struggle to efficiently capture detailed semantics, while methods optimizing for consistency between augmented image views rely on inductive biases not present in uncurated data like web crawls or video frames. How can we learn more effectively from broad unlabeled image datasets? We study annotation bootstrapping, an approach that learns to associate images to semantic annotations, and uses unlabeled data to bootstrap the model's understanding by making predictions about the semantics of nearby crops of an image. A key strength is that it decouples specification (what semantic concepts are interesting?) from prediction (how do these concepts occur in natural image data?). We show that annotation bootstrapping allows us to guide pre-training with a curated unlabeled dataset or a weakly-supervised dataset, while learning from all uncurated image data via the bootstrapping loss. Our experiments demonstrate improved pre-training on unlabeled images in the wild, including video data like EpicKitchens, scene data like COCO, and web-crawl data like CC12M.

1. Introduction

The ability to pre-train on large uncurated text corpora has propelled much recent progress in language modeling. Even though unlabeled images are similarly plentiful and easy to collect — from the Internet, embodied agents, videos, and beyond — learning from this data has proven a challenge.

The difficulty is that unlabeled images pose a raw signal with redundancy, low information density, and noise. With no explicit supervision, methods often turn to carefully designed objectives that teach models to obey useful inductive biases using unlabeled data, such as invariance of representations between augmented views of an image. This line of work has achieved much empirical success, but these methods rely on biases tailored to curated datasets like Imagenet and downstream metrics like object classification. It is unclear how they may be generalized towards more uncurated sources of images like web crawls or videos, or towards other downstream tasks like visual question-answering or embodied action prediction.

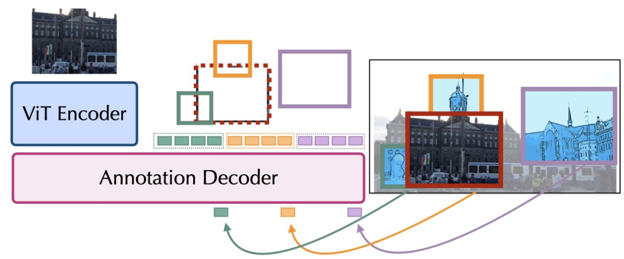

Figure 1. In annotation bootstrapping, we train a model to predict semantic annotations associated with different sub-crops of an image. Our key idea is that we can learn this from *unlabeled* images, even though they provide no supervision of their own. We instead learn by *bootstrapping:* using our model's predictions on one view to generate a learning signal for a different view.

In this paper, we study a pre-training approach that uses unlabeled image data to improve a model's understanding of visual semantic concepts. Consider Figure 1 as intuition: once the model can recognize the dome in the orange frame or the fountain in the green frame, we can ask the model to predict these semantics from the red frame:

What object is above the red frame? A: A dome with a bell.
What's in the bottom left of the red frame? A: A fountain or statue.

The central mechanism here is the bootstrap: that a model's semantic understanding of one image view can generate a training signal *for the same model* to improve its understanding of a different view. Iterating this process can lead to self-improvement; as a model improves its semantic understanding on one subview of an image, it creates supervisory signals to improve the model's understanding of other parts of the same image.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

We concretize this intuition as an objective that we call *annotation bootstrapping*. We use a weakly-labeled or curated dataset to learn semantic associations between images and annotations; we then learn from unlabeled data by training the model to make predictions about the semantics of different sub-crops of an image. In effect, pre-training occurs in two threads: a loss on the curated or labeled data focused on specification (what semantic concepts are interesting?), and a bootstrapping loss on the uncurated unlabeled data focused on prediction (how do these concepts co-occur in natural image data?).

We may propagate the semantics of many common losses using annotation bootstrapping. For instance, bootstrapping atop the CLIP loss yields a self-supervised process that, in effect, predicts the captions associated with one view of an image from another. In experiments using CC12M, we found that this bootstrapping yielded significantly better representations over other methods combining weak supervision and self-supervised losses like SimCLR or DINO.

We may also bootstrap from a self-supervised base loss, like those that optimize for crop-consistency. We show that we can train with crop-consistency on a curated unlabeled dataset (where the crop-consistency inductive bias fits), and then bootstrap these image semantics to a different unlabeled dataset where these inductive biases do not. In our experiments, we show that annotation bootstrapping improves training on COCO (Lin et al., 2014) and Epic-Kitchens (Damen et al., 2020), whose images are not object-centric and where standard self-supervised methods degrade.

Our primary contribution is annotation bootstrapping, a pre-training objective that uses unlabeled images to bootstrap a model's understanding of relationships between images and semantic annotations. Compared to reconstructive and invariance based approaches, this approach offers controllability of learned features through curation or supervision, while ensuring the model may still learn on all unlabeled images available. Our experiments verify the effectiveness of annotation bootstrapping for "in-the-wild" unlabeled datasets like web crawls and video frames where self-supervised objectives typically falter. Annotation bootstrapping offers one approach towards pre-training that can ingest more universal sources of data, and that may train models stronger than the supervision that they are provided.

2. Related Work

Self-supervised learning. Self-supervised methods generally learn in one of two ways: by reconstruction or enforcing representational consistency. Reconstruction-based approaches adopt the "token prediction" ethos from language modeling, and directly predict raw pixels (He et al., 2021) or other low-level features (Xie et al., 2021; Bao et al., 2021) from masked or corrupted inputs, making them simple and easily scalable to large models (El-Nouby et al., 2024; Bai et al., 2024; ChameleonTeam, 2024). However, these objectives yield poor representations for downstream tasks, often require finetuning, and greatly benefit from some data curation (El-Nouby et al., 2024).

Consistency-based approaches use carefully crafted objectives to learn better semantic features, most common being to enforce invariance under random crops and augmentations. Consistency can be optimized directly by contrastively attracting representations of paired views and repelling negative pairs (van den Oord et al., 2018; He et al., 2019; Chen et al., 2020b; Tian et al., 2020), e.g. SimCLR (Chen et al., 2020a). Other approaches implicitly optimize for consistency by iterative self-distillation, e.g. DINO (Caron et al., 2021) or BYOL (Grill et al., 2020). These classes come with different challenges: contrastive methods are stable but require a large batch size to learn effectively (He et al., 2019; Chen et al., 2021); self-distillation methods are more unstable and require careful architectural or objective changes, such as logit sharpening (Caron et al., 2021), k-means clustering (Caron et al., 2019), non-differentiable transports (Caron et al., 2020), or asymmetric predictors (Grill et al., 2020; Xie et al., 2021). Both classes of methods are highly sensitive to the augmentation strategy (Chen et al., 2020a; Chen & Li, 2020) and the choice of data distribution (HaoChen & Ma, 2023; Venkataramanan et al., 2024; Jha et al., 2024). Even at the largest scale, Oquab et al. (2023) find that curation techniques to filter and rebalance collected web data are integral to performance.

Vision-language pre-training. Methods have found success in combining weakly-supervised learning, which learn to associate images and textual captions scraped from the internet (Radford et al., 2021; Jia et al., 2021; Zhai et al., 2023), with the self-supervised objectives above. SLIP (Mu et al., 2021) combines CLIP with a SimCLR objective using an auxiliary head, Li et al. (2021) jointly runs CLIP and SimCLR both on the same representation, and SiLC (Naeem et al., 2023) combines SigLIP with a DINO objective. Combining these losses improves the data efficiency of contrastive vision-language training, and improves performance for more fine-grained tasks like segmentation and prediction (Naeem et al., 2023). However, Fini et al. (2023) and Weers et al. (2023) suggest to the contrary that self-supervised objectives offer only a regularizing effect, and that this gain may be similarly achieved by increasing augmentation in the CLIP objective or by increasing the scale of captioned data (Cherti et al., 2023).

Semi-supervised learning While our paper focuses on unlabeled image pre-training guided by descriptive annotations like free-form text, it is adjacent and inspired by a longer line of semi-supervised approaches learning with partially annotated class labels. Two techniques are common: combining a supervised classification loss with an self-supervised objective on unlabeled data (Pathak et al., 2016; Chen et al., 2020b; Zhai et al., 2019a; Xie et al., 2019), and using the supervised dataset to create pseudo-labels (Lee et al., 2013) for unlabeled images (Xie et al., 2019; Pham et al., 2020). Both pseudo-labeling and our annotation bootstrapping generate target predictions using the model's outputs, but with one important difference: pseudolabeling creates labels for a different student model for the same image, while we use bootstrapping to synthesize supervision for the *same model, but a different image view*.

3. Pre-Training by Annotation Bootstrapping

The core of our approach is to use semantic relationships learned from weakly-labeled or curated unlabeled datasets to define a learning problem over unlabeled images. We connect unlabeled images to this signal through the bootstrap: that our model's predicted relationships on one image view can serve as supervision to train the model on a *different* view. By training to predict grounded semantic concepts, we hope to be able to learn from broader uncurated datasets containing useful visual training signals, but where oft-used inductive biases like crop-consistency do not directly apply.

Pre-training consists of optimizing two threads in parallel: a base loss that associates images to useful semantic concepts ("annotations"), and a self-supervised "bootstrapping" loss to teach models how to predict these relationships for image views that are visually nearby the current one. These two training objectives are synergistic: as a model learns to associate one subview of an image with semantic concepts, it creates a supervisory signal to improve the model's understanding of other parts of the same image.

3.1. Objective

We will define our self-supervised objective supported by some loss that associates an image x with semantic concepts ℓ, which we will refer to as *annotations*. For clarity of exposition, we will assume that this loss is optimized by sampling batches of image-annotation pairs $(x, \ell) \sim \mathcal{D}_a$ and performing noise-contrastive estimation:

$$\max \mathbb{E}_{\{(x_i,\ell_i)\}_{i=1}^n \sim \mathcal{D}_a}[\sum_i \log \frac{\exp(f(x_i, \ell_i))}{\sum_j \exp(f(x_i, \ell_j))}], \quad (1)$$

to estimate the conditional distribution $p(\ell|x)$.

This pattern encompasses many common self-supervised or weakly-supervised learning algorithms of interest. For instance, using text captions as annotations corresponds exactly to the CLIP objective (Radford et al., 2021). Amongst self-supervised methods, using an augmented crop of the same image as annotation: $\ell = \text{augment}(\text{randomcrop}(x))$ recovers the SimCLR objective.

While annotations may take significantly different forms, e.g. text strings (CLIP) vs. corrupted views (SimCLR), what they share in common is that annotations define a natural space to describe and compare images. That is, instead of directly predicting an image x, we may instead predict the annotation distribution of the image $p(\ell|x)$, since two images with similar annotation distributions are notionally similar. Compared to raw pixel prediction or crop-consistency, which prioritize large visually prominent details and obscure subtle semantics, prediction over *annotation distributions* can capture details more uniformly despite their size.

With this in mind, we revisit the generic self-supervised prediction objective used by generative methods: from a partial image view x_1 (for example, by cropping, masking, or noising), we "reconstruct" the neighboring scene with one twist: we do so in the space of annotations, not pixels.

To implement this, we sample a source partial view of an image x_1 by cropping the image to bounding box $\mathbf{bb}_1$, and similarly a target view x_2 with bounding box $\mathbf{bb}_2$. The model is trained to predict the annotation distribution associated with x_2, given x_1 and a description of where x_2 is relative to x_1 (e.g. for crops, we may specify the coordinates of $\mathbf{bb}_2$ in reference to $\mathbf{bb}_1$).

$$\min D_{KL}(p(\ell|x_2) \parallel p_{\text{AB}}(\ell|x_1, \mathbf{bb}_{1\to 2})) \quad (2)$$

This objective trains the model to predict annotations associated with crops nearby the current image (e.g. questions of the form "is there a dog to the right of the image?", "if the image is zoomed out, will there be a playground?", etc.).

We can train this objective even though we have no ground-truth annotations on unlabeled images or their crops. The key to this is the bootstrap: that we can use the predictions of our (currently training) model on x_2 to synthesize a useful target distribution for the same model for $(x_1, \mathbf{bb}_{1\to 2})$.

We term this process *annotation bootstrapping* to be evocative of bootstrapping as it appears in reinforcement learning (RL). In RL, value functions are often learned by supervising the value predictions of a state and action (s, a) with the predicted value of the ensuing state s'. Our pre-training may be interpreted similarly, as we learn about an image view x_1 and an "action" $\mathbf{bb}_{1\to 2}$ by supervising it to match the annotation information from the ensuing image view x_2.

3.2. Practical Implementation

We now describe a practical implementation that can be used with any base loss that learns an annotation distribution $p(\ell|x)$ using Equation 3. For exposition, we will first describe it with CLIP (where annotations are text captions), and then describe the minor changes needed to extend it to

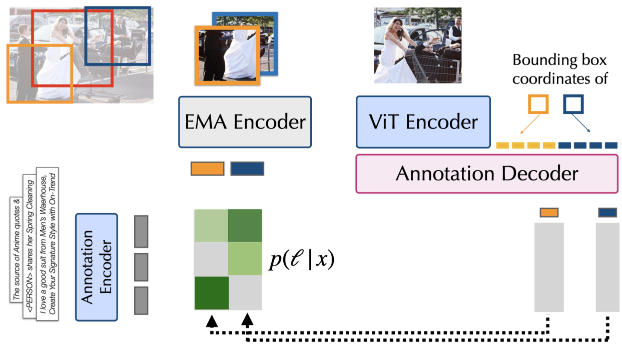

Figure 2. Visualization of bootstrapping objective in our method with a base CLIP loss. The model processes a crop of an image x_1 and a set of tokens demarcating the target bounding box locations ($\mathbf{bb}_{1\to 2}, \mathbf{bb}_{1\to 3}, \dots$) using a standard encoder-decoder architecture. The target supervision is created by running an EMA copy of the model on the target views $\{x_2, x_3, \dots\}$.

self-supervised methods like SimCLR and DINO.

The base CLIP loss trains representations to maximize the cosine similarity between paired images and captions, and minimize those between all other pairs in the batch. The image representation $\phi(x)$ consists of a Vision Transformer (Dosovitskiy et al., 2020), pooled and projected to a shared embedding space; the annotation representation $\psi(\ell)$ is symmetrically implemented (a Transformer, followed by pooling and projection). Training with the InfoNCE objective, for any image x and list of annotations $\{\ell_i\}_{i=1}^n$, these representations define a distribution over annotations:

$$p_\theta(\ell_i|x) \propto \text{softmax}_i \left(t * \phi(x)^\top \psi(\ell_i) \right) \tag{3}$$

where t is a learned scaling constant.

The bootstrapping objective, translated for the base CLIP objective is – given a partial image view x_1 and the relative coordinates of a second view x_2 – to predict the distribution over captions associated with this other view x_2. We mimic the contrastive form of the base predictive distribution:

$$p_{\text{AB}}(\ell_i|x_1, \mathbf{bb}_{12}) \propto \text{softmax}_i \left(t_{\text{AB}} * \phi_{\text{AB}}(x_1, \mathbf{bb}_{12})^\top \psi_{\text{AB}}(\ell_i) \right) \tag{4}$$

The ”image-action” representation is implemented is a standard “S”-size Transformer decoder atop the CLIP image backbone; it takes input a set of tokens describing the coordinates of the desired view x_2 and cross-attends to the visual embeddings. The annotation representation $\psi_{\text{AB}}(\ell)$ is an independent head atop the CLIP text backbone.

During training, we sample unlabeled image data and generate n random crops of each image I: $x_i, \mathbf{bb_i} = \text{RandomCrop}(I)$. We take a set of (unpaired) annotations from the annotation dataset, and for any two views i, j, we train to minimize the KL divergence between the base estimated annotation distribution at x_j and the model's predictions from $x_i, \mathbf{bb}_{i\to j}$:

$$\mathcal{L}_{\text{AB}} = D_{KL}(p_{\text{base}}^{\text{ema}}(\ell|x_j) \parallel p_{\text{AB}}(\ell|x_i, \mathbf{bb}_{i\to j})) \tag{5}$$

where the distributions are defined as in Equations 3 and 4. An important component to ensure the stability of this objective is to use a lagging EMA average of model parameters when computing the target distribution, a well-known deficiency for bootstrapping methods in reinforcement learning. Through token packing and batching, this loss can be computed efficiently across all n^2 pairs of views.

The implementation is summarized in Algo. 1 and Figure 2. When annotations correspond to images (i.e. when the base loss is SimCLR), we do not need a separate text encoder, and instead use an independent head atop the image backbone to represent $\psi_{AB}(\ell_i)$. When annotations belong to a discrete set, (e.g. when the base loss is DINO or a classification task), the annotation representation simplifies into an embedding matrix. In Appendix A, we describe (with pseudocode) the annotation and the image representations for the three instantiations of annotation bootstrapping that we study in our experiments: AB_{CLIP} atop the CLIP loss, $\text{AB}_{\text{SimCLR}}$ atop a SimCLR loss, and AB_{DINO} atop a DINO loss.

3.3. Connections

Soft distillation and pseudo-labeling. The bootstrapping objective, in form, resembles distillation objectives like pseudo-labeling (Iscen et al., 2019; Yang et al., 2023), but induces a very different effect. Distillation transfers knowledge from one model p_θ to another q_θ about an image x; annotation bootstrapping instead transfers knowledge from one images x_2 to another $(x_1, \mathbf{bb}_{1\to 2})$, but for the same model. This distinction is significant, as we are interested in objectives that improve pre-training of the current model, and not those requiring re-training new models from scratch. *Consistency via self-distillation.* The bootstrapping objective also closely relates to self-supervised methods that optimize for consistency via iterative self-distillation. Amongst others, DINO (Caron et al., 2021) and SwAV (Caron et al., 2020) also predict distributions over “prototypes” (cf. annotations) associated with one crop x_2 from a different crop x_1. However, DINO and SwAV seek invariancy, that all crops of an image should emit the same distribution over prototypes. In contrast, annotation bootstrapping optimizes for equivariance; in Figure 2, the orange, blue, and red crops should correspond to different annotation distributions since they capture different semantic details (like the wedding dress, or the shopping cart, or a man in the background); bootstrapping enforces these be predictable from (not the

Algorithm 1 Pseudocode for the bootstrapping objective (visualized in Figure 2)

```
# Inputs: `images': list of images, `annotations`: list of annotations
view1, bbox1 = RandomResizedCrop(images)
view2, bbox2 = RandomResizedCrop(images)

# First, we must compute our (EMA) model's distributions over annotations for view2
target_phi = ema_model.image_head(ema_model.image_backbone(view2))
target_psi = ema_model.annotation_head(ema_model.annotation_backbone(annotations))
target_logits = ema_model.t * target_phi @ target_psi.T

# The bootstrapping loss requires the model to predict these targets
# from view1 and the coordinates of view2
view_tokens = discretize(relative_bbox(bbox1, bbox2))
ab_phi = model.decoder(model.image_backbone(view1), view_tokens)
ab_psi = model.annotation_head(model.annotation_backbone(annotations))
logits = model.ab_t * ab_phi @ ab_psi.T

bootstrapping_loss = CrossEntropy(logits, softmax(target_logits))
```

same as) one another.

Also similar is I-JEPA (Assran et al., 2023); both make predictions in a self-supervised manner about a target view (specified by positional tokens) using an encoder-decoder architecture. The objectives differ in three main ways: the prediction of individual token embeddings for each patch instead of a single pooled output, prediction with an L2 loss in feature space versus a probability divergence between the predicted and target annotation distributions, and most importantly the use of targets generated from running the model on the *full image*, not just the target view. In practice, we find that our approach far exceeds the performance of I-JEPA on these domains; we hypothesize that despite the similarities in architecture and objective, the lack of grounding of the I-JEPA targets makes the model more sensitive to choices of crop, data distribution, and hyperparameters.

4. Experiments

We study the utility of using annotation bootstrapping to pre-train on uncurated datasets of unlabeled images found "in the wild". Our study focuses on the following questions:

1. Can bootstrapping improve pre-training with different base annotation losses like SimCLR, DINO, or CLIP?
2. How does bootstrapping compare to invariance-based or pixel-predictive self-supervised objectives?
3. Can we bootstrap from a curated dataset to learn on a different unlabeled dataset?

As we investigate these questions, we additionally probe the training process to understand how annotation propagation interfaces with the base loss, and the effect of various design decisions in this process. Full experimental details about the method, training, and evaluation are in Appendix B and C. Example code is provided in the supplementary.

Training. We standardize training by running all methods on all datasets using ViT-S/16 vision encoders (and S-sized text encoders in the weakly labeled setting) for $800M$ seen images (each view is counted separately). For ImageNet, this corresponds to approximately 620 epochs of the dataset. All models are trained with AdamW, weight decay, gradient clipping, and using a cosine decay schedule – specific hyperparameters are taken from respective papers when they are provided (see Table C in Appendix B for a full list).

We emphasize that our experimental goal is not to claim state-of-the-art performance on standard unsupervised benchmarks, but rather to evaluate annotation bootstrapping on a wide set of domains and more carefully analyze *bootstrapping in annotation space*, and how it relates to common patterns like crop-consistency and pixel reconstruction.

Evaluation. To avoid overfitting to Imagenet probing performance, we evaluate on a wider set of tasks using the probing strategy introduced by Beyer et al. (2023). In this setup, evaluation tasks (including classification, object detection, visual question answering, captioning, etc) are cast as a sequential modeling problem, and learned using a lightweight decoder that cross-attends with frozen ViT token embeddings. This solution allows us to evaluate a broader set of downstream tasks under a unified interface.

4.1. Pre-training with a self-supervised base loss.

We first evaluate annotation bootstrapping in the fully-unlabeled setting, where we bootstrap from a base SimCLR loss (AB_{SimCLR}) or DINO loss (AB_{DINO}) to make predictions in the induced space of image-image relationships.

When pre-training on unlabeled ImageNet images (Table

Table 1. Bootstrapping annotations enables improvement over several weakly supervised and self-supervised base losses (AB_{CLIP} over CLIP, AB_{SimCLR} over SimCLR, AB_{DINO} over DINO) on CC-12M (Changpinyo et al., 2021), a weakly-curated web crawl dataset with 8.7 million images. The gap is greatest for AB_{CLIP}, where we find that it significantly outperforms other approaches combining self-supervision with weak-supervision. ***Avg. Cls** averages the classification accuracy over the four benchmarks in Beyer et al. (2023): Food101, Oxford IIIT Pets, Resics45, and Sun397.

Pretrain Dataset	Method	ImageNet	Avg Cls*	Clevr/Depth	Clevr/Count
CC12M (No labels)	MAE	60.1	74.5	81.4	88.5
	I-JEPA	60.0	76.0	80.1	**90.2**
	SimCLR	64.9	74.3	78.3	87
	AB_{SimCLR}(Ours)	65.8$_{+0.9}$	79.1$_{+4.8}$	80.0$_{+1.7}$	89.1$_{+2.1}$
	DINO	67.8	79.5	79.5	87.1
	AB_{DINO}(Ours)	**68.7**$_{+0.9}$	**80.1**$_{+0.6}$	**82.1**$_{+2.6}$	89.4$_{+2.3}$
CC12M (w/ Captions)	CLIP	70.0	82.4	73.1	84
	SLIP +SimCLR (Mu et al., 2021)	69.0	81.1	**77.3**	88.7
	SiLC +DINO (Naeem et al., 2023)	71.0	83.6	73.9	86.6
	AB_{CLIP}	**74.6**$_{+4.6}$	**84.0**$_{+1.6}$	**78.0**$_{+4.9}$	**92.9**$_{+8.9}$

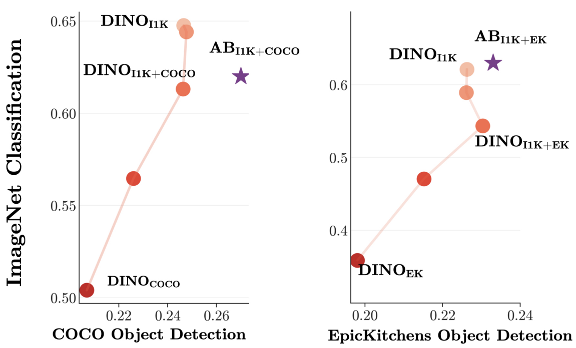

Figure 3. We compare decoupled training of AB_{DINO} on ImageNet and COCO or EpicKitchens to running DINO on a mixture for $p \in \{0, 0.25, 0.5, 0.75, 1.0\}$. AB_{DINO}outperforms all the DINO mixtures, indicating that bootstrapping leads to better performance than self-supervised DINO on any combination of the two datasets.

	Method	Linear	MAP	Decoder
ImageNet (No Labels)	MAE	55.0	60.5	65.0
	I-JEPA	58.5	61.5	64.5
	SimCLR	67.0	68.7	70.0
	AB_{SimCLR}	66.0	69.6	71.0
	DINO	**68.5**	70.0	72.2
	AB_{DINO}	68.0	**71.5**	**73.6**

Table 2. Bootstrapping a self-supervised loss learns better representations than training with the base loss alone on unlabeled ImageNet images, especially for probes that attend to tokens like Multihead Attention Pooling or a Transformer decoder probe.

4.1, additional probes in Table C), a standard well-curated dataset, we find annotation bootstrapping to be synergistic to the base SimCLR / DINO loss, improving performance over running the base losses for a longer period of time. Investigating different probes of the visual representation, the improvement is greatest seen on probes that attend to the encoded tokens (like MAP pooling or a larger decoder), but not those that have been reduced to a single token (e.g. by global average pooling). Perhaps unsurprisingly, we find annotation bootstrapping learns better immediate representations for MAE, a prototypical pixel reconstructive approach), and iJEPA, an example of the bootstrapping objective without semantic grounding. We find these trends to also hold when training fully self-supervised using images from CC-12M (Table 1, a larger and less curated dataset of web-crawl images common for vision-language training (Changpinyo et al., 2021).

We find that crop-consistency methods significantly degrade when they are pre-trained them on scene datasets, specifically on COCO (Lin et al., 2014) and Epic-Kitchens (Damen et al., 2020), treating video data as individual frames following (Venkataramanan et al., 2024). These datasets are a poor fit for the inductive bias underlying consistency methods, not being object-centric, and instead containing many (small) objects, and crop-consistency methods like DINO and SimCLR learn significantly degraded representations relative to more generic methods like MAE (Table 3). On these domains, we find that annotation bootstrapping greatly increases over only running the base loss. However, it is equal or slightly worse than MAE across the board, indicating that while bootstrapping can improve features, it cannot significantly improve upon a base loss whose features do not capture semantic details well.

We test the ability of annotation bootstrapping to decouple the annotation and bootstrapping data distributions, since in theory the former loss may be optimized with a curated dataset to specify semantics, and learning only by bootstrapping on our target unlabeled images. In Table 3, below the line, we find that this decoupled approach leads to significantly better performance for in-domain tasks like object recognition, localization, and action recognition. We analyze this more carefully by sweeping a base DINO algorithm with 5 different data mixture ratios between Imagenet and { Coco, EpicKitchens }. Our results in Figure 3, indicate that annotation bootstrapping learns representations beyond the

Table 3. COCO (Lin et al., 2014) and EpicKitchens (Damen et al., 2020) have different visual semantics from object-centric datasets like ImageNet, causing significant degradation to invariance-based self-supervised methods like SimCLR and DINO. $\text{AB}_{\text{SimCLR}}$ and AB_{DINO} can alleviate these deficiencies. Exploiting the decoupled nature of $\text{AB}_{\text{SimCLR}}$ and AB_{DINO}, we show how pre-training can be improved by learning base features on ImageNet, and bootstrapping learned annotations to COCO and EpicKitchens.

	Method	ImageNet Cls.	COCO Object Detection	COCO Object Cls.
COCO	MAE	**62.3**	**0.31**	**76.5**
	I-JEPA	43.0	0.21	62.5
	SimCLR	56.2	0.24	70.4
	$\text{AB}_{\text{SimCLR}}$	$60.2_{+4.0}$	$0.26_{+0.02}$	$72.3_{+1.9}$
	DINO	56.1	0.24	70.5
	AB_{DINO}	$59.7_{+3.6}$	$0.27_{+0.03}$	$72.6_{+2.1}$
COCO + ImageNet	$\text{AB}_{\text{SimCLR}}$	**68.3**	**0.31**	**79.2**
	AB_{DINO}	65.0	**0.31**	**78.6**

	Method	ImageNet Cls.	EK Action Recog.	EK Object Detect.	EK Object Cls
Epic Kitchens	MAE	**43.5**	20.8	**0.387**	**44.3**
	I-JEPA	38.7	18.5	80.1	39.5
	SimCLR	48.1	20.3	0.299	**44.3**
	$\text{AB}_{\text{SimCLR}}$	$50.6_{+2.5}$	$\mathbf{22.1}_{+1.8}$	$0.354_{+0.055}$	$\mathbf{44.5}_{+0.2}$
	DINO	43.4	18.9	0.295	39.6
	AB_{DINO}	$47.1_{+3.7}$	$19.8_{+0.9}$	$0.328_{+0.033}$	$42.5_{+2.9}$
EK + ImageNet	$\text{AB}_{\text{SimCLR}}$	**68.5**	**23.7**	**0.389**	47.3
	AB_{DINO}	63.0	22.9	0.371	**47.7**

Pareto frontier generated by only running DINO.

Pre-training with a weakly-supervised base loss. We next turn to evaluating annotation propagation in the weakly labeled setting, when the annotations are tokenized strings of text. Recall that in this setting, our approach learns by associating text from images using a base CLIP loss, and bootstrapping by making predictions about the image-text relationships of other crops of an unlabeled image.

On CC12M (Table 1, bottom), we see that weakly supervised methods across the board outperform their unsupervised equivalents; this matches empirical evidence that contrastive language-text methods are more capable of training on lower-quality image data. As discussed by Naeem et al. (2023), we find that combining CLIP with a self-supervised objective like DINO (SiLC) or SimCLR (SLIP) primarily improves fine-grained reasoning on the ClevR benchmark tasks, with only marginal improvement on downstream classification tasks. In contrast, annotation bootstrapping obtains much stronger performance relative to these other approaches on classification and segmentation metrics we evaluated, in particular improving by 4.6% on downstream ImageNet probing performance over the base CLIP representations. We hypothesize that since annotation propagation learns by making predictions about text distributions associated with other crops of an image, it learns features that

Table 4. Combining weakly-labeled supervision with standard self-supervised objectives on COCO degrades performance. AB_{CLIP} is the only method that improves over base CLIP training

Type of Annotations	Method	Object Cls	Detection
COCO Captions	CLIP	25.4	71.9
	SLIP	25.8	75.1
	SILC	21.8	71.2
	AB_{CLIP}	**29.7**	**76.7**
Bounding Boxes	CLIP	31.6	76.4
	SLIP	28.3	76.3
	SILC	29.2	76.4
	AB_{CLIP}	**34.9**	**82.5**

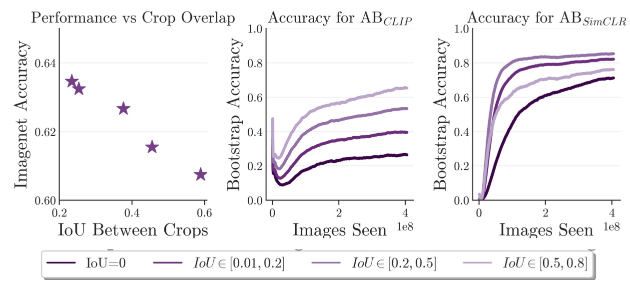

Figure 4. (left) Controlling the difficulty of the bootstrapping prediction problem, we find performance to degrade as the overlap source and target crop grows. (right). Accuracy of annotation decoder in training; the model quickly plateaus to predict nearby crops, but does keeps learning about further crops through training.

are better aligned with the CLIP objective, whereas SLIP and SILC losses may act more orthogonally to the CLIP representation.

We test this hypothesis by comparing different weakly-supervised methods for pre-training on COCO in Table 4, a dataset where we found crop-consistency methods to struggle. We source text descriptions of these images from two annotation sources: captions (Karpathy & Li, 2015) and bounding box descriptions (Lin et al., 2014), both directly present inthe COCO dataset.

In this setting, we notice that AB_{CLIP} is the only method that improves over CLIP, while both SLIP and SiLC counterintuitively *decrease* in performance. Our findings support the hypothesis of Weers et al. (2023), that invariance-based objectives are not necessarily additive upon weakly supervised learning, but instead move the model towards an invariant solution. When crop-consistency matches the inductive biases of the data, adding self-supervision leads to improved performance, but otherwise may degrade performance. In contrast, annotation bootstrapping seems to improve performance over the base CLIP loss, even when inductive biases like consistency do not fit the unlabeled data.

ABLATION	IMAGENET PERFORMANCE
AB_{SimCLR}	63.0
Adding augmentations	62.5 $_{-0.5}$
Removing action tokens	60.8 $_{-2.2}$
No propagation loss	59.4 $_{-3.6}$
No target network	59.4 $_{-3.6}$
No annotation loss	39.0 $_{-24.0}$

Table 5. Ablating different components of AB_{SimCLR}on CC12M.

4.2. Analysis

We now investigate the learning process of annotation bootstrapping, to understand various design decisions in the method, and how the loss evolves through training. These ablatory experiments are run using a budget of 400M views.

How well is the bootstrapping objective optimized through training? In Figure 3 (right), we plot the prediction accuracy for the bootstrapping objective throughout the course of training, clustering by how far the target prediction box is in terms of IoU. Notice that prediction errors increase initially in training as the annotation head is first learned, but decreases uniformly through training. We note that the prediction problem is more challenging for AB_{CLIP}than for AB_{DINO}, reflecting the fact that the base DINO objective is jointly trying to make the predictive distribution more similar across different crops, while CLIP learns a fixed and grounded annotation space.

How does the choice of crops affect the quality of bootstrapping? We next investigate how the choice of bounding boxes affects the performance of the algorithm, by sampling source and target bounding boxes that are closer (or further) apart while keeping the marginal distribution over bounding boxes fixed. In Figure 3, we see that performance increases steadily as the average IoU between the source and target distributions is decreased, meaning that we are makin predictions about image crops that are "further away" from our current view. Combined with Figure 3, these results reflect the folk wisdom that training on the most difficult examples offers the most useful learning signal.

What components of annotation bootstrapping most affect downstream performance?

We ablate different components of the method in Table 4.2. As with other bootstrapping and self-distillation methods, we find that removing the EMA network nullifies all performance gains from the bootstrapping objective. Similarly, removing the base loss, which grounds annotation distributions in a semantically meaningful space, significantly degrades performance. We also perform an ablation replacing the bounding box description tokens with empty mask tokens, thereby forcing the model to predict the *average* annotation distribution across different crops. Doing so turns the bootstrapping objective from one of equivariance to invariance, since all crops are trained to match the same average distribution. In our ablation, we find this invariance to be far this reduces the effectiveness of the propagation objective. Perhaps surprisingly, we see that adding image augmentations to either the source or target views actually hurts performance.

The general heuristic appears to be that one should select challenging target images as possible, without introducing any additional stochasticity into the prediction targets (e.g by adding image augmentations or removing action tokens).

5. Discussion

Our paper introduced annotation bootstrapping, a self-reinforcing approach to pre-training visual representations using unlabeled data. Our method learns by predicting the annotations associated with various sub-crops of an image;. Two qualities make annotation bootstrapping particularly interesting: first, that it cleanly partitions the pre-training process into the specification of image semantics and bootstrapping, allowing us to learn useful details using curated or labeled datasets, while still being able to pre-train on unlabeled images that do not have the same inductive biases as the curated data. As we saw across a number of datasets, annotation propagation learns useful semantic representations beyond those that are learned from common objectives like pixel prediction, CLIP, or models that learn invariances to crops and augmentations.

Our approach is not without limitation; relative to the scale that current CLIP models are being trained on, we were only able to train on relatively small datasets (CC12M only has 8 million images) and with relatively small networks (ViT-S), and at limited training durations. Some of the conclusions in our paper may weaken at larger scales. Second, while the bootstrapping objective does reduce the dependency on inductive biases compared to invariance-based or pixel-predictive approaches, the choice of crops seems to still affect the quality of learned representations. There are are many avenues of further research: how these objectives behave at scale, whether we can use hard-mining or large batch sizes to amplify the signal from the bootstrapping objective, or even whether we we may form an autoregressive version of bootstrapping that makes pre-training look like self-supervised VQA. Our work takes a step towards understanding how we may pre-train on visual data in a self-sufficient bootstrapped manner using vast swaths of unlabeled data. Already at the larger scales of model pre-training today, we are beginning to see methods consume most easily-accessible weakly-labeled data. We must soon answer the question: how will we improve our models when the labeled data runs out?

Impact Statement

The goal of our work is to advance visual pre-training methods capable of ingesting broader uncurated image datasets. Our own contributions are on well-studied and regulated datasets, but we stress that training on large uncurated scrapes of data has the potential for violation of privacy, safety, and perpetuation of biases that may be present on the web. We encourage the community to be careful and cautious of attempts to pre-train these methods at the largest scales, and to carefully analyze the qualia of the pre-trained models before deployment or fine-tuning to real settings.

References

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M. G., LeCun, Y., and Ballas, N. Self-supervised learning from images with a joint-embedding predictive architecture. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Bai, Y., Geng, X., Mangalam, K., Bar, A., Yuille, A. L., Darrell, T., Malik, J., and Efros, A. A. Sequential modeling enables scalable learning for large vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22861–22872, 2024.

Bao, H., Dong, L., and Wei, F. Beit: Bert pre-training of image transformers. *ArXiv*, abs/2106.08254, 2021.

Beaumont, R. img2dataset: Easily turn large sets of image urls to an image dataset. https://github.com/rom1504/img2dataset, 2021.

Beyer, L., Zhai, X., and Kolesnikov, A. Big vision. https://github.com/google-research/big_vision, 2022.

Beyer, L., Wan, B., Madan, G., Pavetic, F., Steiner, A., Kolesnikov, A., Pinto, A. S., Bugliarello, E., Wang, X., Yu, Q., Chen, L.-C., and Zhai, X. A study of autoregressive decoders for multi-tasking in computer vision. *ArXiv*, abs/2303.17376, 2023.

Caron, M., Bojanowski, P., Mairal, J., and Joulin, A. Unsupervised pre-training of image features on non-curated data. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2959–2968, 2019.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *ArXiv*, abs/2006.09882, 2020.

Caron, M., Touvron, H., Misra, I., J'egou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021.

ChameleonTeam. Chameleon: Mixed-modal early-fusion foundation models, 2024.

Changpinyo, S., Sharma, P. K., Ding, N., and Soricut, R. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3557–3567, 2021.

Chen, T. and Li, L. Intriguing properties of contrastive losses. *CoRR*, abs/2011.02803, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020a.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *CoRR*, abs/2006.10029, 2020b.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9620–9629, 2021.

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., and Jitsev, J. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

Damen, D., Doughty, H., Farinella, G. M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., and Wray, M. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:4125–4141, 2020.

Darkhalil, A., Shan, D., Zhu, B., Ma, J., Kar, A., Higgins, R., Fidler, S., Fouhey, D., and Damen, D. Epic-kitchens visor benchmark: Video segmentations and object relations. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020.

El-Nouby, A., Klein, M., Zhai, S., Bautista, M. A., Toshev, A., Shankar, V., Susskind, J. M., and Joulin, A. Scalable pre-training of large autoregressive image models, 2024.

Fini, E., Astolfi, P., Romero-Soriano, A., Verbeek, J., and Drozdzal, M. Improved baselines for vision-language pre-training. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. Featured Certification.

Grill, J.-B., Strub, F., Altch'e, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. Á., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, abs/2006.07733, 2020.

HaoChen, J. Z. and Ma, T. A theoretical study of inductive biases in contrastive learning, 2023.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2019.

He, K., Chen, X., Xie, S., Li, Y., Doll'ar, P., and Girshick, R. B. Masked autoencoders are scalable vision learners. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, 2021.

Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. Label propagation for deep semi-supervised learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5065–5074, 2019.

Jha, A., Blaschko, M. B., Asano, Y. M., and Tuytelaars, T. The common stability mechanism behind most self-supervised learning approaches, 2024.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q. V., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. *ArXiv*, abs/2102.05918, 2021.

Karpathy, A. and Li, F. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 3128–3137. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298932.

Lee, D.-H. et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., and Yan, J. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *ArXiv*, abs/2110.05208, 2021.

Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

Mu, N., Kirillov, A., Wagner, D. A., and Xie, S. Slip: Self-supervision meets language-image pre-training. *ArXiv*, abs/2112.12750, 2021.

Naeem, M. F., Xian, Y., Zhai, X., Hoyer, L., Van Gool, L., and Tombari, F. Silc: Improving vision language pretraining with self-distillation. *arXiv preprint arXiv:2310.13355*, 2023.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. Q., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y. B., Li, S.-W., Misra, I., Rabbat, M. G., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016.

Pham, H., Xie, Q., Dai, Z., and Le, Q. V. Meta pseudo labels. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11552–11563, 2020.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211 – 252, 2014.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

Venkataramanan, S., Rizve, M. N., Carreira, J., Asano, Y. M., and Avrithis, Y. Is imagenet worth 1 video? learning strong image encoders from 1 long unlabelled video. In *International Conference on Learning Representations*, 2024.

Weers, F., Shankar, V., Katharopoulos, A., Yang, Y., and Gunter, T. Masked autoencoding does not help natural language supervision at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23432–23444, June 2023.

Xie, Q., Dai, Z., Hovy, E. H., Luong, M.-T., and Le, Q. V. Unsupervised data augmentation for consistency training. *arXiv: Learning*, 2019.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: a simple framework for masked image modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9643–9653, 2021.

Yang, X., Song, Z., King, I., and Xu, Z. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2023. doi: 10.1109/TKDE.2022.3220219.

Zhai, X., Oliver, A., Kolesnikov, A., and Beyer, L. S4l: Self-supervised semi-supervised learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1476–1485, 2019a.

Zhai, X., Puigcerver, J., Kolesnikov, A., Ruyssen, P., Riquelme, C., Lucic, M., Djolonga, J., Pinto, A. S., Neumann, M., Dosovitskiy, A., Beyer, L., Bachem, O., Tschannen, M., Michalski, M., Bousquet, O., Gelly, S., and Houlsby, N. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv: Computer Vision and Pattern Recognition*, 2019b.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952, 2023.

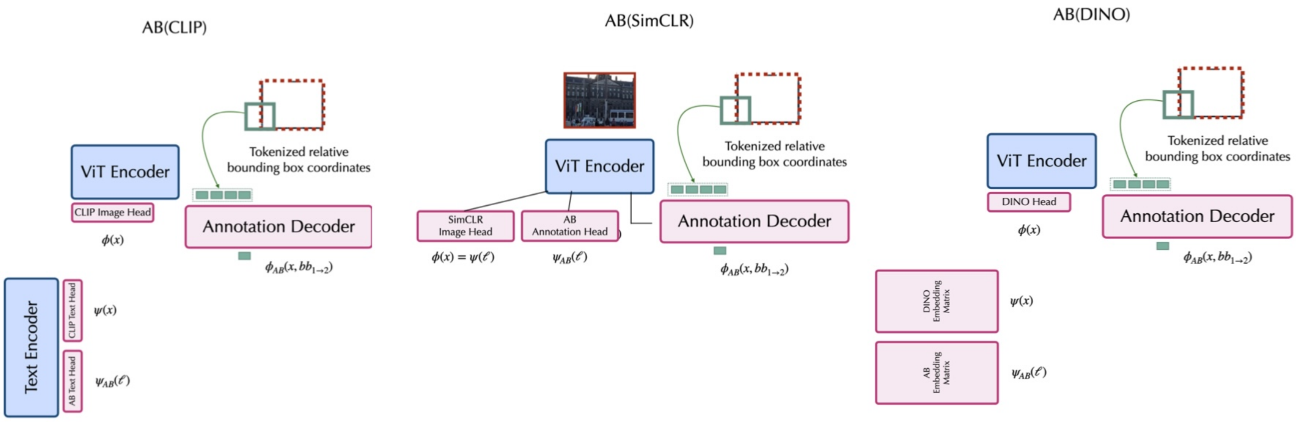

Figure 5. Visualizations of Annotation Bootstrapping for different base learning algorithms, AB_{CLIP}, AB_{SimCLR}, AB_{DINO}. The model architectures are near identical on the visual side: a ViT vision encoder, with a head for the base loss, and a Decoder transformer to predict the annotations associated with other bounding boxes. What differs between the implementations is how annotations are embedded. In CLIP, they are embedded by a separate text encoder; in SimCLR, they are embedded by the same vision backbone; in DINO, they form an embedding matrix. All methods train with the recipe in Algorithm 2.

A. Annotation Bootstrapping Details

Algorithm 2 General Annotation Bootstrapping Pseudocode

```
def loss(annotation_batch, bootstrap_batch, model, ema_model):
  logits = model(annotation_batch['image'], annotation_batch['text'])
  # The annotation loss (CLIP here) associates images and annotations
  # Replace with SimCLR loss or DINO loss for the appropriate variants
  annotation_loss = CrossEntropy(logits, eye(B_a)) + CrossEntropy(logits.T, eye(B_a))

  # The bootstrapping loss uses view1 to predict annotations associated with view2
  view1, bbox1 = RandomResizedCrop(bootstrap_batch['image'])
  view2, bbox2 = RandomResizedCrop(bootstrap_batch['image'])
  target_logits = ema_model(view2, annotations) # B_b x B_a

  # The model is given view1 and the coordinates of view2 wrt view1
  view_tokens = discretize(relative_bbox(bbox1, bbox2))
  logits = model(view1, annotation_batch['text'], target_view=view_tokens) # B_b x B_a

  bootstrapping_loss = CrossEntropy(logits, softmax(target_logits))

  return annotation_loss + bootstrapping_loss
```

B. Training Details

Models. We implement our models and baselines in Jax, using the `bigvision` repository (Beyer et al., 2022) implementation of all transformer components, such as the vision encoder, the text encoder for CLIP, and the annotation decoder that predicts latent representations from encoded imge tokens and bounding box tokens. We were unable to replicate the results from I-JEPA in our internal codebase, so we train this baseline directly using the publicly available code. In Table C, we provide the hyperparameters for all evaluated methods; we obtained hyperparameters from the official code-bases whenever possible; for CLIP, we adopt hyperparameters from Fini et al. (2023), who tune the hyperparameters of CLIP for CC-12M scale training.

Datasets. We evaluate on four datasets representative of the many types of unlabelled images typically available: Imagenet

(Russakovsky et al., 2014), a well-curated, balanced, and image-centric benchmark heavily used by prior work; CC12M (Changpinyo et al., 2021), a dataset of captioned images used for vision-language pre-training that is relatively uncurated and contains a wider range of concepts than Imagenet; COCO (Lin et al., 2014) a dataset of scenes each containing many (potentially small) objects, and Epic-Kitchens (Damen et al., 2020), a video dataset containing many real-world scenes in homes. Note that CC12M is a dataset of links, so links deteriorate due to rot and redirects; the version we collected (Beaumont, 2021) has 8.7 million images.

C. Evaluation

We use the multi-task decoder-based probe from Beyer et al. (2023) for the evaluations in this paper. The probe is defined as a 4-layer transformer decoder with an autoregressive decoding pattern that attends to the final outputs of the Vision Transformer through cross-attention. We choose this architecture so that we can do all of our probing tasks, whether image recognition or bounding box prediction, or classification of the object in a bounding box using a unified framework; this also represents (albeit to a much smaller scale) how vision transformers are being used in VLM models. We adopt all hyperparameters for training this model from Beyer et al. (2023).

When pre-training on Imagenet and CC12M, we probe the model on ImageNet, the Clevr/{Count, Distance} tasks from Zhai et al. (2019b), and then on four tasks used by Beyer et al. (2023): Food101, Oxford IIIT Pets, Resics45, and Sun397.

When pre-training on COCO, we evaluate on small object classification (in which the model is provided the coordinates of a bounding box, and asked to predict the identity of the object within that bounding box), and the corresponding detection task (in which the model must simply identify all bounding boxes corresponding to relevant objects in a scene).

When pre-training on EpicKitchens, we probe the model also on object classification (predicting the label of an object given its bounding box) and object detection (predicting bounding boxes), which we source from the ViSOR annotation set (Darkhalil et al., 2022). We also probe the model's ability to predict the action a human is taking given one frame of context. This problem is not exactly solvable from one frame of context, but the relative performance differences between methods nonetheless informs the quality of the learned representations.

Table 6. Downstream classification metrics beyond ImageNet accuracy when pre-training fully unlabelled on ImageNet/ ***Avg. Cls** averages the classification accuracy over the four benchmarks in Beyer et al. (2023): Food101, Oxford IIIT Pets, Resics45, and Sun397.

Pretrain Dataset	Method	ImageNet	Avg Cls*	Clevr/Depth	Clevr/Count
ImageNet (No Labels)	SimCLR	70.0	80.1	76.9	86.0
	DINO	72.2	82.8	80.0	88.1
	MAE	65.0	77.7	**81.7**	88.6
	I-JEPA	64.5	79.0	81.0	88.8
	AB_{DINO}	**73.6**	**83.7**	81.4	**89.3**

Table 7. Hyperparameters used by all algorithms in our experiments

	MAE	iJEPA	DINO	SimCLR	CLIP	SLIP	SILC	AB
Effective Batch Size (= Batch Size * Views)	8192	4096	10240	8192	8192	8192	9216	8192
Batch Size	8192	4096	1024	4096	8192	CLIP: 4096 SimCLR: 2048	CLIP: 4096 DINO: 512	Annotation batch size: base algo // Bootstrap batch size: base algo //
Number of Views	1	1*	10 (2 global, 8 local)	2	1	2	10 (2 global, 8 local)	4 views for bootstrap batch
Model	ViT-S/16	ViT-S/16	ViT-S/16	ViT-S/16	ViT-S/16 S-size Text decoder	ViT-S/16	ViT-S/16	Follows base loss "S"-sized annotation decoder
Augmentations	RRC(0.2, 1.0), HorizontalFlip	RRC(0.3, 1.0)	Global: RRC(0.4, 1.0), Local: RRC(0.05, 0.04) HorizontalFlip ColorJitter, Random GrayScale	RRC(0.08, 1.0) HorizontalFlip ColorJitter	RRC(0.5, 1.0)	Follows CLIP and SimCLR augmentations	Follows CLIP and DINO augmentations	For unlabeled data RRC(0.05, 1.0) For annotation data follows base loss
Warmup Steps	10,000	40 ImageNet Epochs	10 ImageNet Epochs	10 ImageNet Epochs	1 CC12M Epoch			Follows base loss
LR	2.4e-3	1e-3	1e-3	1e-3	1e-3			Follows base loss
Weight Decay	0.05	$0.04 \rightarrow 0.4$	$0.04 \rightarrow 0.4$	$0.04 \rightarrow 0.4$	0.1			Follows base loss
Gradient Clipping	None	None	1.0	None	None			Follows base loss
EMA	None	$0.004 \rightarrow 0$	$0.004 \rightarrow 0$	None	None	None	0.004	0.004 for AB_{CLIP} $0.004 \rightarrow 0$ for AB_{SimCLR}, AB_{DINO}
Additional Hyperparameters	$b_2 = 0.95$				$b_2 = 0.98$	Loss Ratio = 1.0	Loss Ratio = 1.0	Loss Ratio = 1.0