# Improving CLIP Fine-tuning Performance

Yixuan Wei[1], Han Hu[2]*, Zhenda Xie[1], Ze Liu[3], Zheng Zhang[2], Yue Cao[2],
Jianmin Bao[2], Dong Chen[2], Baining Guo[2]
[1]Tsinghua University [2]Microsoft Research Asia [3]USTC
{t-yixuanwei,hanhu,t-zhxie,t-liuze,zhez,yuecao,jianbao,doch,bainguo}@microsoft.com

## Abstract

*CLIP models have demonstrated impressively high zero-shot recognition accuracy, however, their fine-tuning performance on downstream vision tasks is sub-optimal. Contrarily, masked image modeling (MIM) performs exceptionally for fine-tuning on downstream tasks, despite the absence of semantic labels during training. We note that the two tasks have different ingredients: image-level targets versus token-level targets, a cross-entropy loss versus a regression loss, and full-image inputs versus partial-image inputs. To mitigate the differences, we introduce a classical feature map distillation framework, which can simultaneously inherit the semantic capability of CLIP models while constructing a task incorporated key ingredients of MIM. Experiments suggest that the feature map distillation approach significantly boosts the fine-tuning performance of CLIP models on several typical downstream vision tasks. We also observe that the approach yields new CLIP representations which share some diagnostic properties with those of MIM. Furthermore, the feature map distillation approach generalizes to other pre-training models, such as DINO, DeiT and SwinV2-G, reaching a new record of 64.2 mAP on COCO object detection with +1.1 improvement. The code and models are publicly available at* https://github.com/SwinTransformer/Feature-Distillation.

## 1. Introduction

The pre-training and fine-tuning paradigm is instrumental in the success of deep learning methods in computer vision, as evidenced by numerous influential works such as [20, 30, 13, 39]. One common practice is to use model weights pre-trained on ImageNet-1k classification task [10] as the initialization for various downstream vision tasks, such as object detection [13] and semantic segmentation [39]. However, this approach faces two key challenges: the difficulty in scaling up high-quality image classification data, and the limited semantic information contained in category labels, both of which constrain the ability to further improve model performance.

Table 1: Improving the fine-tuning performance of the ViT-B/16 CLIP model [42] via a classical feature distillation framework. The model is distilled on ImageNet-1K dataset [10] with images only for 300 epochs. Clear gains are observed on four evaluation benchmarks. MAE [17] results are also listed in gray for reference.

| Method | IN-1K (%) | | ADE20K | COCO | | NYUv2 |
|---|---|---|---|---|---|---|
| | linear | f.t. | mIoU | $AP_{box}$ | $AP_{mask}$ | RMSE (↓) |
| MAE [17] | 68.0 | 83.6 | 48.1 | 46.5 | 40.9 | 0.383 |
| CLIP [42] | 79.5 | 82.9 | 49.5 | 45.0 | 39.8 | 0.416 |
| **FD**-CLIP | 80.1 | 85.0 | 51.7 | 48.2 | 42.5 | 0.352 |
| Δ | ↑**0.6** | ↑**2.1** | ↑**2.2** | ↑**3.2** | ↑**2.7** | ↓**0.064** |

The recent CLIP [42] alleviates these challenges. It utilizes contrastive learning to learn representations from web-scale noisy vision-language pairs. The learned representations exhibit impressive semantic modeling capabilities, as evidenced by performance on zero-shot image classification and image-text retrieval tasks. Meanwhile, a new self-supervised pre-training method based on masked image modeling (MIM) [2, 58, 17] has also attracted great attention for its excellent fine-tuning performance on various downstream tasks. Without losing generalizability, we mainly discuss MAE [17] in this paper.

When comparing the two pre-training methods, the CLIP model learns richer semantic information reflected by its superior linear probing performance on ImageNet-1K. However, its fine-tuning performance on most other tasks are worse than MAE, as shown in Tab. 1. This observation appears counter-intuitive since models with better semantics are usually considered to have better transferability. This raises a further question: can CLIP be made as successful as, or even surpass, MIM in fine-tuning? To answer this question, we firstly decompose the ingredients of these pre-training methods into three aspects: input ratios, training

*Corresponding Author. The work is done when Yixuan Wei, Zhenda Xie, and Ze Liu are interns at Microsoft Research Asia.

Table 2: Improving the fine-tuning performance of the ViT-L/14 CLIP model [42] on ImageNet-1K classification.

| Method | Res. | Pre-train datasets | IN-1K(%) |
|---|---|---|---|
| WiSE-FT [54] | $336^2$ | WIT-400M [42] | 87.1 |
| DeiT III [52] | $384^2$ | IN-22K [10] | 87.7 |
| ViT [11] | $512^2$ | JFT-300M [50] | 87.8 |
| Scaling [63] | $384^2$ | JFT-3B[63] | 88.5 |
| BeiT [2] | $512^2$ | DALLE [45] & IN-22K | 88.6 |
| CLIP [42] | $224^2$ | WIT-400M | 86.1 |
| **FD**-CLIP | $224^2$ | WIT-400M | **87.7** (+1.6) |
| | $224^2$ | WIT-400M & IN-22K | **88.3** |
| | $336^2$ | WIT-400M & IN-22K | **89.0** |

target granularity and training losses, as listed in Tab. 3. By comparing the differences between CLIP and two typical MIM approaches, we exclude the training losses to be responsible for the inferior fine-tuning performance of CLIP, and speculate that the input ratios (*i.e*. full image *vs*. partial image) and training target granularity (*i.e*. image-level *vs*. token-level) might be key factors. Although narrowing the differences in input ratios is straightforward, changing the granularity of the CLIP training targets from image-level to token-level poses a significant challenge, since existing vision-language training data is more suitable for image-level supervision and lacks fine-grained information.

Knowledge distillation [21] is a widely used technique for transferring information from one model to another, typically for the purpose of model compression. In this paper, we demonstrate that distillation can also perform as a bridge for converting the training target granularity of CLIP models from image-level to token-level, while preserving the semantic information. Specifically, we take the pre-trained CLIP model as the teacher model, use its output feature map as the distillation target, and distill this information into a randomly initialized student model that shares the same architecture and size as the teacher model. This process is illustrated in Fig. 1, which we refer to as *"feature distillation"* to differentiate it from logits distillation [21, 51, 12]. Notably, although the student mimics the teacher model's output, their different optimization paths can lead to different diagnostic properties on the inter-mediate layers, which is thought to be important for fine-tuning.

The flexibility of the distillation framework allows us to introduce proper inductive bias and regularization to shape the optimization path of the student model and enhance the student model performance on downstream tasks. Specifically, we propose several crucial adjustments: 1) Standardization of the teacher feature map. This adjustment amplifies the subtle information contained within the teacher model and stabilizes the output value range; 2) Asymmetric drop path rates for the teacher and student models. This asymmetric regularization enhances the robustness of

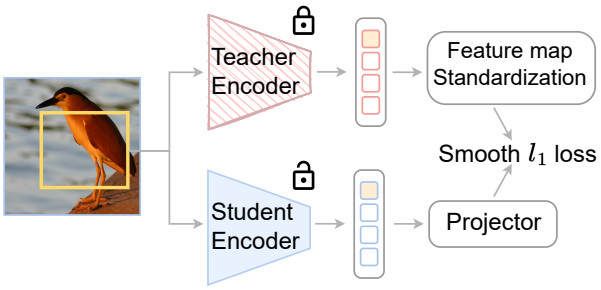



Figure 1: Illustration of the feature map distillation that introducing token-level targets to distill the pre-trained CLIP models. The orange block stands for `[CLS]` token, and the orange box means a random crop of the original image.

student representations and results in accurate and consistent teacher signals. 3) Shared relative position bias. The introduction of this inductive bias further strengthens the translation-invariant property of the student model.

With the feature distillation framework and the above adjustments, we derive a model that preserves the strong semantic information of CLIP while being friendly to downstream task fine-tuning, as shown in Tab. 1. We observe consistent improvements compared to the original CLIP on various tasks: +2.1 accuracy gains on ImageNet-1K image classification [10], +2.2 mIoU gains on ADE20K semantic segmentation [66], +3.2 box AP and +2.7 mask AP on COCO object detection and instance segmentation [36], and reducing RMSE(↓) by 0.064 on NYUv2 depth estimation [49]. The improvement remains when scaling up to the largest CLIP-L/14 model with a +1.6 accuracy gain on ImageNet-1K, as shown in Tab. 2. Moreover, when generalizing our method to other models, like DINO [3], DeiT [51] and the advanced SwinV2-G [37], we still earn clear gains on various downstream tasks, especially reaching a new record of 64.2 mAP on COCO object detection with +1.1 mAP improvement on SwinV2-G.

In addition to these improvements in experimental results, we further introduce several diagnostic tools to analyze the properties of learned visual representations from different models. These analyses provide deeper insights into understanding how feature distillation improves the CLIP model: 1) diversifying different attention heads of the CLIP model in deeper layers; 2) improving the translational invariance of learned representations; and 3) flattening the loss landscapes and reflecting optimization friendliness.

Our contributions are summarized as follows:

- We examine the ingredient differences between CLIP and MIM methods and demonstrate target granularity is vital in the success of MIM in fine-tuning.
- We leverage the classical feature map distillation to convert the training target granularity of CLIP to token-level ones, which enhances its fine-tuning per-

formance and preserves its semantic information.

- We propose several crucial techniques during feature distillation that further enlarge the improvements, including distilling standardized feature maps, asymmetric drop path rates, and shared relative position bias.
- With several diagnostic tools, we find that compared to CLIP, both MIM and **FD**-CLIP possess several properties that are intuitively good, which may provide insights on their superior fine-tuning performance.
- We generalize our method to various pre-training models and observe consistent gains. We also set a new record on COCO object detection, by improving the advanced 3B SwinV2-G model with our framework.

## 2. Related Work

**Representation Learning** There are four notable representation learning approaches in vision area. 1) Image classification (CLS) on supervised datasets [10, 50] has been the standard upstream pre-training task for nearly a decade since AlexNet [30]. The pre-trained weights are applied to numerous down-stream tasks including image segmentation [66, 16], object detection [36, 46] and video recognition [24, 15]. 2) The contrastive language-image pre-training (CLIP) task is to connect paired images and texts and separate unpaired ones, which opens up the field of zero-shot recognition [42, 23], and proves to be powerful in multi-modality down-stream tasks [47, 40, 44]. 3) Instance contrastive learning (CLR) method performs pre-training in a self-supervised manner by contrasting the augmentation views of the same image with others [18, 57]. The method achieves impressive accuracy using linear and few-shot evaluations [3, 5, 31]. 4) Masked image modeling (MIM) learns representations also in a self-supervised way, which first masks a large portion of the image area and learns to predict the pixel values or features of the masked area. It excels in fine-tuning evaluations [2, 17, 58].

In this paper, we propose to adopt the classical feature map distillation framework to derive a same-size refreshed CLIP model which performs better on downstream tasks and largely preserves the semantic information incorporated in original CLIP. We also generalize our method to non-CLIP models, including supervised models, like DeiT [51] and SwinV2 [37] and self-supervised models, like DINO [3].

**Knowledge Distillation** Knowledge distillation [21, 26] is firstly proposed in CNN models, and further explored in Transformers [51, 53] to boost the supervised training performance and compress a compact small model. *"Dark knowledge"* is proven to be useful in distillation when student models mimic the logits prediction of the teachers [12]. Beyond supervised learning, knowledge distillation is also widely used in deep reinforcement learning [48, 4], lifelong learning [62] and recommendation system [7].

Our work does *not* aim to propose a new distillation approach, but leverage this flexible framework with crucial designs to mitigate the differences on input ratios and target granularity between CLIP and MIM methods. We find that tasks with token-level target granularity maybe a key ingredient to the success of MIM methods. And feature distillation serves as a counter-part for CLIP, which improves its fine-tuning performance and largely maintain its original semantic information.

Table 3: Ingredients comparison between CLIP and MIM methods from the perspective of input ratios, training target granularity and loss format.

| Method | Input | Target | Loss | Semantics |
|---|---|---|---|---|
| BeiT [2] | Partial | Token-level | Cross-entropy | |
| MAE [17] | Partial | Token-level | Regression | |
| CLIP [42] | Full | Image-level | Cross-entropy | ✓ |
| **FD**-CLIP | Full | Token-level | Regression | ✓ |

**Model Diagnosing and Explanation** Model diagnosing is important for demystifying the "black box" of deep learning models due to their high-dimensional and non-linear nature [59]. There have been also works [67, 11, 9, 14, 29, 43, 41] seeking to understand Transformers, including attention analysis [67, 11, 56], loss landscapes visualization [33], CKA [28] and knowledge neuron discovery [9, 14]. Inspired by these works, we adopt a set of attention- and optimization-related diagnostic tools to reveal the unique properties of MIM method. And these tools are also applied on **FD**-CLIP to provide a better understanding of feature distillation process.

## 3. Improving CLIP by Feature Distillation

The CLIP method is known for its ability to incorporate rich semantics learned by contrasting tremendous image-text pairs. Compared to the MIM methods (*e.g.* MAE), CLIP is more consistent with human concepts, as evidenced by its superior linear probing performance (as shown in Tab. 1). However, its impressive semantic capability seems to marginally benefit downstream tasks fine-tuning. By closely examining the training loss curves for the object detection task, depicted in Fig. 2, we note that the classification loss, *i.e.* $L_{\text{cls}}$, is close between MAE pre-training and CLIP pre-training, but MAE appears to have better localization ability with lower $L_{\text{bbox}}$. These differences motivated us to investigate the factors behind CLIP's sub-optimal fine-tuning performance and explore ways to unleash its powerful semantic capability better.

We compare the ingredient differences between CLIP and MIM methods shown in Tab. 3. Based on it, we spec-

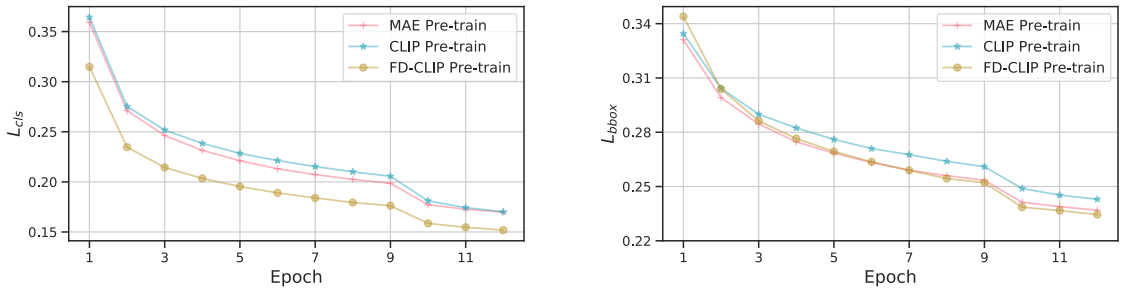



Figure 2: Fine-tuning MAE, CLIP and **FD**-CLIP on COCO object detection task. We visualize the loss curves of $L_{\text{cls}}$ and $L_{\text{bbox}}$ w.r.t the training epoch. Although CLIP pre-training is comparable to MAE pre-training on $L_{\text{cls}}$, it shows worse localization ability, reflected by the $L_{\text{bbox}}$ curve.

ulate the input ratios (*i.e.* full image *vs*. partial image) and training target granularity (*i.e.* image-level *vs*. token-level) might be responsible for CLIP's inferior fine-tuning performance. While it is relatively easy to use partial inputs in CLIP, directly changing the training target from image-level to token-level would be challenging, as CLIP and existing vision-language training data are designed for image-level supervision. More importantly, re-training CLIP is costly that we would like to avoid.

Therefore, we propose to use distillation techniques, which are usually used for model compression, as a bridge for converting the training target granularity of CLIP models from image-level to token-level, while preserving the semantic information of the pre-trained model. To be specific, the pre-trained model serves as a frozen teacher, and a new same model with randomly initialized weights plays as the student, as illustrated in Fig. 1.

Instead of distilling logits like most previous distillation works [21, 51], we adopt the full output feature map of the pre-trained model as the distillation target, dubbed *"feature distillation"*. This approach allows us to work with any pre-trained model including those not having logits output. Moreover, distilling the feature map also leads to higher fine-tuning accuracy than only distilling a reduced single feature vector (see Tab. 4), emphasizing the importance of training target granularity for pre-trained models. To ensure that the feature maps of the teacher and student are aligned, we apply the same augmentation view to each original image. A light-weight projector is added on top of the student network to allow for different output feature map dimensions between the teacher and student models, further generalizing the method.

While the goal of the student model is to closely mimic the teacher model, training the student network from scratch allows a different optimization path. This relaxation of the optimization path provides the possibility for the student model to possess similar properties to those of MIM while maintaining the most of expressive power of the teacher network. Then we propose the following designs to make better relaxation and introduce desirable inductive biases and regularization, which further boosts the transferability of the student models.

**Standardizing the teacher's feature map** Different pre-trained models may have very different orders of feature magnitudes, which will make difficulties in hyper-parameter tuning. In addition, the subtle information that encoded in small values may not be well distilled into the student network without amplification. To solve these issues, we normalize the output feature map of the teacher network by a standardization operation, which is implemented by a non-parametric layer normalization operator [1] and proven to be important in Tab. 7 (a).

In distillation, we employ a smooth $\ell_1$ loss between the student and teacher feature maps:

$$\mathcal{L}_{\text{distill}}(\mathbf{s}, \mathbf{t}) = \begin{cases} \frac{1}{2}(g(\mathbf{s}) - t')^2/\beta, & |g(\mathbf{s}) - t'| \leq \beta \\ (|g(\mathbf{s}) - t'| - \frac{1}{2}\beta), & \text{otherwise} \end{cases}, \quad (1)$$

where $\beta$ is set 2.0 by default; $t' = \text{standardization}(\mathbf{t})$; $\mathbf{s}$ and $\mathbf{t}$ are output feature vectors of the student and teacher networks, respectively; $g$ is a $1 \times 1$ convolution layer served as the projector. We amplify the distillation loss weight on `[CLS]` token by 10.0 as it aggregates the global image information during CLIP pre-training.

**Asymmetric drop path rates** The two-branch structure in the feature distillation framework allows for asymmetric regularization on the teacher and student networks. We find that applying a strategy of asymmetric drop path [22] rates can learn better representations. Specifically, the strategy of a drop path rate of 0.1 on the student branch with no drop path regularization on the teacher branch works best on ViT-B, as shown in Tab. 7 (b).

**Shared relative position bias** The original CLIP model [42] adopts the absolute position encoding (APE), but recent works [38, 35, 34] found the relative position bias (RPB) shows benefit on downstream tasks. Benefiting from the flexibility of feature distillation, we are able to

re-examine the impacts of position encoding configuration in the student architecture. In particular, we investigate a *shared RPB* configuration, where all layers share the same relative positional bias matrices. Our experiments show that the *shared RPB* performs best overall, as shown in Tab. 7 (c). We find that the *shared RPB* enhances the diversify of heads, particularly for the deeper layers (as shown in a figure in the supplementary material), which likely contributes to its slightly better fine-tuning performance.

# 4. Diagnostic tools

In addition to verifying the effectiveness of feature distillation through experimental results, we analyze several interesting properties of the learned visual representations to provide an understanding of the behind mechanism, by following diagnostic tools:

- *Average attention distances per head [11].* This diagnostic tool measures the average relative distance each patch token attends to in the image, which partially reflects the receptive field size for each attention head, computed using the attention weights. The `[CLS]` token and each patch itself are omitted in measurement, and the distances are pixel-level.
- *Average attention maps for each layer [67].* We visualize the attention maps averaged over all heads per layer. There are two common patterns in the attention maps: *diagonal* and *vertical-bar*. The *diagonal* pattern reveals that the model relies more on visual cues from relationships of relative locations. It also suggests better translation in-variance of the model, which is often a beneficial property for various down-stream visual tasks. However, the *vertical-bar* pattern reflects the strong impact of the patches in a fixed location to all other locations, which is translation variant. For the *diagonal* pattern, concentrating to a centered diagonal can also reflect *locality prior*, i.e. the more concentrated to the center, the stronger the locality prior.
- *Normalized loss landscapes [33].* In this diagnostic tool, the trained model weights are perturbed by a series of Gaussian noises with varying degrees. Following [33], each noise level is normalized to the $\ell_2$ norm of each filter to account for the effects of varying weight amplitudes of different models. Visually flatter minimums usually correspond to lower test error and better generalization ability [33].

# 5. Experiments and Analysis

In this section, we investigate whether the fine-tuning performance of CLIP can be improved via bridging the gap with MIM methods through feature distillation. We firstly study the impacts on fine-tuning performance of different input ratios and training target granularity during distillation, and then we ablate several key designs in our method. Additionally, we provide a detailed analysis of the models before and after distillation, using our diagnosis tools.

## 5.1. Experimental Settings

**Distillation settings**. For all experiments, we perform feature distillation on 1.28M ImageNet-1K training images [10]. In ablation, we distill 100 epochs for all experiments, except for 300 epochs in Tab. 1 and Tab. 2. The default model size is ViT-B/16 if not mentioned else. Other details are in *supplemental materials*.

**Evaluation settings**. We include 4 evaluation benchmarks: ImageNet-1K classification [10], ADE20K semantic segmentation [66], COCO object detection and instance segmentation [36] and NYUv2 depth estimation [49].

- *ImageNet-1K classification*. For fine-tuning, we follow [2] to use the AdamW optimizer [27] with layer-wise decayed learning rates and an input size of $224 \times 224$. For ViT-B, we fine-tune it by 100 epochs, and for ViT-L, we fine-tune it by 50 epochs. For linear probing, we follow [17] to use the LARS optimizer [60] with a base learning rate of 0.1 and a weight decay of 0 training for 90 epochs. Top-1 accuracy is reported. Other details are in *supplemental materials*.
- *ADE20K semantic segmentation*. We follow [38] to use an UPerNet framework [55] for experiments. The AdamW [27] optimizer is employed with the training length of 80K, a batch size of 32, and a weight decay of 0.05. Other hyper-parameters are set as: learning rate 4e-4, layer decay 0.65, and drop path rate 0.2. In training, the input image size is $512 \times 512$. In inference, we follow the single-scale testing of [38]. Mean IoU on the validation set is reported.
- *COCO object detection and instance segmentation*. We follow the most settings in [6] including a Mask-RCNN framework [19] with $1\times$ schedule, multi-scale training and single-scale testing. To reduce the GPU memory cost brought by global self-attention on high-resolution COCO images, we adopt a shifted window attention like [38] and set the window size as 14. An additional global self-attention layer is added on the top to aggregate information from whole images. Bbox mAP and mask mAP on the validation set are reported. Other details are in *supplemental materials*.
- *NYUv2 depth estimation*. We follow the settings in [56, 25]. The input images are randomly cropped to $480 \times 480$ with a batch size of 24, maximal learning rate 5e-5 and 25-epoch training. We evaluate the RMSE (Root Mean Square Error) on this task. Other details are in *supplemental materials*.

Table 4: Ablation on distilling target granularity. The models are distilled on ImageNet-1K dataset [10] with 100 epochs. Token-level targets is vital to boost the fine-tuning performance of CLIP.

| Method | IN-1K | ADE20K | COCO | | NYUv2 |
|---|---|---|---|---|---|
| | % | mIoU | $AP_{box}$ | $AP_{mask}$ | RMSE (↓) |
| MAE [17] | 83.6 | 48.1 | 46.5 | 40.9 | 0.383 |
| `[CLS]` token | 81.9 | 47.5 | 44.8 | 39.6 | 0.396 |
| GAP feature | 83.3 | 50.3 | 46.3 | 40.6 | 0.393 |
| Full map | **84.4** | **51.8** | **47.9** | **42.2** | **0.350** |

## 5.2. On training target granularity and input ratios

**Training target granularity**. We firstly investigate the impacts of training target granularity. To disentangle the impacts of distillation and target granularity, we ablate three different distillation targets:

- `[CLS]` *token*. The `[CLS]` token of the visual encoder plays a unique role in CLIP, which not only aggregates the global image information, but also aligns to the language modality with rich semantics. In this setting, we use the output feature of `[CLS]` token from the teacher model as the target to guide the corresponding output of the student model.
- *GAP feature*. In this setting, we use a reduced feature vector as the target. Specifically, a global average pooling layer is applied on the whole feature map to build targets with information from every tokens but lack of resolutions.
- *Full map*. We use the whole feature map without reduction as the target to create token-level supervision, which is the default setting in **FD**-CLIP.

Tab. 4 shows the results. Distilling `[CLS]` token shows an improvement on depth estimation, but performs worse on other tasks than original CLIP. In comparison, distilling GAP feature shows marginal benefits. The use of full feature map performs best among all distillation targets on all the downstream tasks, and also surpassing the MAE model.

**Input Ratios**. We further study the effects of input ratios by distilling masked images. Tab. 5 shows the results of different mask ratios on the student branch, ranging among [75%, 50%, 25%, 0% (*i.e.* Full)]. We find that under the same training epochs, there are no significant differences between distilling full images and partial images, except for the setting with only 25% input that is notably worse.

With the above experiments, we draw the conclusion that the training target granularity is crucial for achieving better fine-tuning performance. We further extend the distillation epochs to 300 and conduct experiments on the largest CLIP model, ViT-L/14. As shown in Tab. 1, **FD**-CLIP outperforms original CLIP by 2 points on ImageNet-1K and ADE20K and earns around +3 mAP gains on COCO. It also presents advantages on low-level tasks like depth estimation on NYUv2, reducing RMSE by 0.033 compared to MAE. When scaling up to ViT-L model, we earns 87.7% top-1 accuracy on ImageNet-1K fine-tuning, surpassing the original CLIP by 1.6% (see Tab. 2). Incorporating with intermediate fine-tuning on ImageNet-22K [10] and a higher fine-tuning resolution to 336 × 336, we reach 89.0% on ImageNet-1K with ViT-L. The other downstream tasks are not conducted on ViT-L/14 due to the inconsistency between the multi-resolution FPN and the model's patch size of 14, which is not an exponential power of 2.

Although distilling the full feature map is different from the pre-training objective of CLIP, Tab. 6 shows that the distilled student model has largely preserved zero-shot and linear probing performance of original CLIP model. That is, the distillation of the full feature map could inherit much information incorporated in the CLIP model while taking the advantages of MIM methods, which may lead to its superior performance.

Table 5: Ablation on partial inputs for distillation. The models are distilled on ImageNet-1K dataset [10] with 100 epochs. ×% map is equal to (100 − ×)% masking. † means we also input a masked image into the teacher model, otherwise we use the full image for the teacher model by default.

| Method | IN-1K | ADE20K | COCO | | NYUv2 |
|---|---|---|---|---|---|
| | % | mIoU | $AP_{box}$ | $AP_{mask}$ | RMSE (↓) |
| MAE [17] | 83.6 | 48.1 | 46.5 | 40.9 | 0.383 |
| 25% input† | 83.3 | 47.8 | 45.2 | 40.1 | 0.397 |
| 25% input | 83.1 | 48.8 | 45.1 | 39.8 | 0.379 |
| 50% input | 84.2 | 51.5 | 47.5 | 41.7 | 0.351 |
| 75% input | **84.4** | **52.0** | 47.8 | 41.9 | **0.347** |
| Full input | **84.4** | 51.8 | **47.9** | **42.2** | 0.350 |

Table 6: Evaluating the distilled CLIP performance on ImageNet-1K [10]. The feature distillation could largely preserve the semantic capability of CLIP.

| Method | CLIP | **FD**-CLIP | Δ |
|---|---|---|---|
| Zero-shot (%) | 68.6 | 68.0 | -0.6 |
| Linear probing (%) | 79.5 | 80.1 | +0.6 |

Table 7: Ablation on other design choices in feature distillation. **Bold** ones are our default settings.

| (a) Normalization | None | $\ell_2$ norm | Standardization |
|---|---|---|---|
| IN-1K (%) | 83.5 | 83.9 | **84.4** |
| (b) Std. / Tea. d.p.r | 0.1 / 0.1 | 0.1 / 0 | 0.2 / 0 |
| IN-1K (%) | 84.0 | **84.4** | 84.0 |
| (c) Position config. | APE | Non-shared RPB | Shared RPB |
| IN-1K (%) | 84.0 | 83.9 | **84.4** |

Table 8: Applying feature distillation on ViT-B pre-trained with DINO [3] and DeiT [51]. The models are distilled on ImageNet-1K dataset [10] with 300 epochs.

| Method | IN-1K | ADE20K | COCO | | NYUv2 |
|---|---|---|---|---|---|
| | % | mIoU | $AP_{box}$ | $AP_{mask}$ | RMSE (↓) |
| DINO [3] | 82.8 | 46.2 | 45.8 | 40.7 | 0.412 |
| **FD**-DINO | 83.8 | 47.7 | 46.1 | 40.9 | 0.394 |
| Δ | ↑**1.0** | ↑**1.5** | ↑**0.3** | ↑**0.2** | ↓**0.018** |
| DeiT [51] | 81.8 | 47.0 | 45.8 | 40.7 | 0.403 |
| **FD**-DeiT | 83.0 | 48.0 | 46.4 | 41.0 | 0.404 |
| Δ | ↑**1.2** | ↑**1.0** | ↑**0.6** | ↑**0.3** | ↑**0.001** |

Table 9: Feature distillation also improves the advanced SwinV2-G model [37] on various downstream tasks.

| Method | IN-1K | COCO | | ADE20K |
|---|---|---|---|---|
| | | $AP_{box}$ | $AP_{mask}$ | mIoU |
| GLIPv2-CoSwin-H [65] | - | 62.4 | - | - |
| Florence-CoSwin-H [61] | - | 62.4 | - | - |
| DINO-Swin-L [64] | - | 63.3 | - | - |
| MaskDINO-Swin-L [32] | - | - | 54.7 | 60.8 |
| ViT-Adapter-L [8] | - | - | - | 60.5 |
| SwinV2-G [37] | 89.2 | 63.1 | 54.4 | 59.9 |
| **FD**-SwinV2-G | **89.4** (+0.2) | **64.2** (+1.1) | **55.4** (+1.0) | **61.4** (+1.5) |

### 5.3. Ablation of design choices

In this section, we ablate other designs of in our feature distillation framework. All experiments are performed on ImageNet-1K dataset using ViT-B and 100-epoch training.

**On the normalization of teacher features**. Tab. 7 (a) ablates the effect of whether and how to perform teacher feature map normalization. Teacher feature map standardization brings +0.9% improvement over using the original feature maps. Comparing two normalization approaches of $\ell_2$ norm and standardization, the latter one shows a gain of +0.5%. Normalization also makes feature distillation hyper-parameters insensitive to the pre-training models.

**On asymmetric drop path rates**. Tab. 7 (b) ablates the effect of different degrees of drop path regularization. Moderately adding the drop path regularization on the student network would be beneficial, possibly due to the relief of over-fitting. However, adding drop path regularization on the teacher model damages the performance, indicating that an accurate teacher signal is beneficial. Therefore, we adopt this asymmetric drop path rate strategy by default.

**On position encoding configurations**. Tab. 7 (c) ablates the effect of varying position encoding configurations in the student network. The results reveal that the shared relative position bias (*shared RPB*) configuration outperforms others. Nonetheless, all configurations perform quite well, so the proper position encoding configuration is not the decisive factor for the success of feature distillation.

### 5.4. Evaluation on more models

Experiments shown in Tab. 1 and Tab. 2 reveal the effectiveness of feature distillation on CLIP models. While the motivation of this work is to improve the fine-tuning performance of CLIP, the feature distillation approach also works with other pre-training models. In Tab. 8, we apply the feature distillation on DINO [3] and DeiT [51] and observe consistent improvements on various downstream tasks. It shows that the feature distillation approach is also effective on models pre-trained with different pre-training objects. We also conduct similar experiments on MAE [17], shown in appendix. **FD**-MAE earns marginal gain on most tasks, verifying our observations that the gain of our method is largely from a token-level task.

We also improve the 3-billion-parameter SwinV2-G to achieve **61.4 mIoU** and **64.2 mAP** on ADE20K semantic segmentation and COCO object detection (using the same UperNet / HTC++ framework and the same evaluation settings as the original Swin V2 [37]), creating new records with +0.6 mIoU and +0.9 mAP higher than previous state-of-the-art reported in (Mask) DINO [64, 32], respectively, as shown in Tab. 9. These results suggest the general applicability of our approach to different pre-training methods and model architectures.

### 5.5. Analysis

Extensive experimental results have shown that feature distillation can facilitate the fine-tuning performance of CLIP models. In this section, we diagnose the models with tools mentioned in Sec. 4 to understand how feature distillation affects the model behaviors. All the analyses are performed on 50,000 ImageNet-1K validation images.

**Diversified attention heads**. We firstly examine the attention diversity of different heads w.r.t. network layers. Fig. 3 shows the average attention distance per head in different network layers of MAE, CLIP and feature distilled CLIP (**FD**-CLIP), respectively. At shallow layers, the learned representations of all models are diverged across heads. However, the diversity on attended distances of different heads is rapidly converging in deep layers of the CLIP model. Intuitively, the converged representation indicating the model capacity is not fully utilized and may have redundancy [56]. In comparison, **FD**-CLIP alleviates this issue and its representations are more similar to the ones in MAE.

**Enhanced translational invariance**. Fig. 4 shows the average attention maps of different models. Compared to CLIP models, the MAE pre-trained model focuses more on the visual cues from the relative locations, and shows more *diagonal* patterns than others, suggests better translational invariance of MAE. This locality property may benefit down-

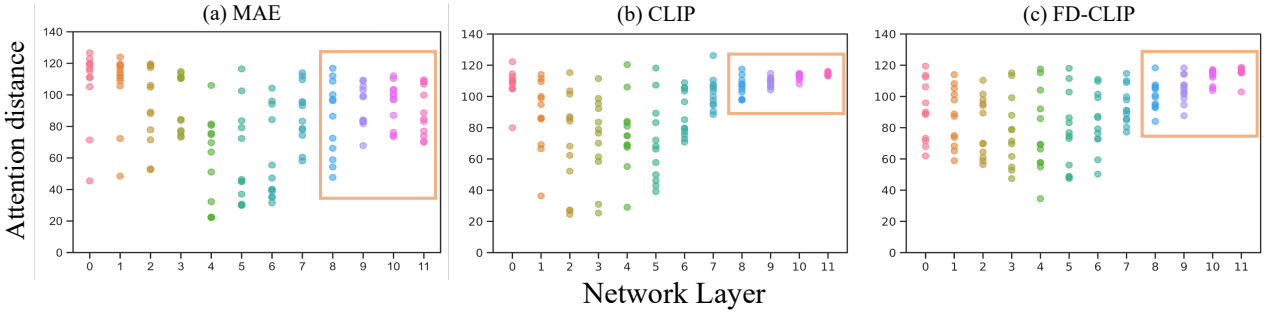



Figure 3: The average attention distance per head at each layer depth on (a) MAE [17], (b) CLIP [42] and (c) **FD**-CLIP. The distances are measured on the pixel level.

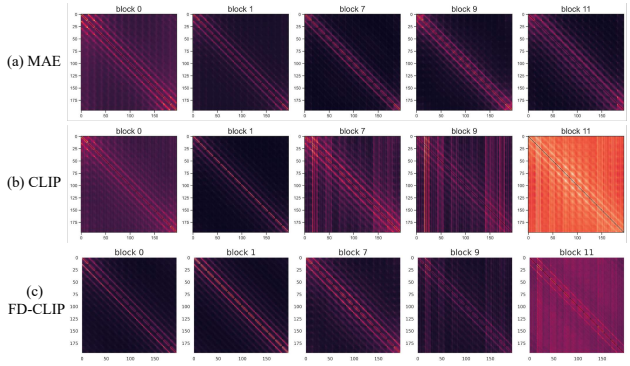
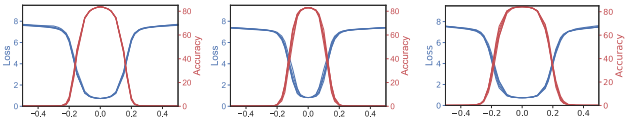



Figure 4: The average attention maps on (a) MAE [17], (b) CLIP [42] and (c) **FD**-CLIP. The maps are averaged over all heads and all images. Five representative layers, 0th, 1st, 7th, 9th, 11th, are selected to save the space. Full attention maps can be found in the *supplementary materials*.




(a) MAE (b) CLIP (c) FD-CLIP

Figure 5: The loss / accuracy landscapes [33] of (a) MAE [17], (b) CLIP [42] and (c) **FD**-CLIP, where x-axis represents the noise strength and y-axis is the loss / accuracy. Each plot has 5 landscapes using 5 randomly generated directions.

stream tasks that requires a fine-grained localization ability. In contrast, the attention maps of CLIP have much more *vertical-bar* patterns in deeper layers (*e.g.* block 7-11), which indicates the CLIP features are dominated by certain patches on absolute locations. The *vertical-bar* patterns partly disappear after feature distillation, revealing that the distilled model relies more on encoding relationship of visual cues from relative locations like what MAE does, and shows better translational invariance.

**Flattened loss landscapes**. Fig. 5 visualizes the loss and accuracy landscapes [33] of different models. It turns out that the landscapes of MAE and **FD**-CLIP are relatively flatter than original ones, which generally reflects its optimization friendliness and better generalization. This observation is also consistent with their better fine-tuning accuracy.

## 6. Conclusion

This paper seeks to adopt a classical feature distillation framework on CLIP models to improve their fine-tuning performance and simultaneously inherit the original semantic capability. By analyzing the ingredient differences and behaviors differences on classification and localization tasks between CLIP and MIM methods, we found that a task with token-level target granularity is one of the key to the success of MIM methods, especially to their impressive fine-tuning performance. From this perspective, we introduced the classical feature distillation framework with several crucial designs to provide a token-level task for pre-trained CLIP models. We gained consistent and clear improvements on various downstream tasks compared to the original CLIP models and largely preserved their semantic capability, like zero-shot and linear probing on ImageNet-1K classification. Besides, we analyzed **FD**-CLIP with MIM and CLIP using several attention- and optimization-related diagnosing tools. The visualizations revealed that after distillation, **FD**-CLIP shares more similar patterns with MIM. Moreover, we further generalized the framework to more various models including DeiT, DINO and the advanced SwinV2-G and observed consistent gains.

**Limitations** Although the performance improvements are noticeable after feature distillation, the model training pipeline becomes more complicated and additional training cost is required, *e.g.* 3% more compared to CLIP pre-training. We further discuss the cost in the appendix. Besides, the diagnosing tools which analyze the MAE and **FD**-CLIP are also intuitive and may not be able to indicate the fine-tuning performance directly.

## References


[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 2, 3, 5

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021. 2, 3, 7

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018. 3

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020. 3

[6] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 5

[7] Xu Chen, Yongfeng Zhang, Hongteng Xu, Zheng Qin, and Hongyuan Zha. Adversarial distillation for efficient recommendation with external knowledge. *ACM Transactions on Information Systems (TOIS)*, 37(1):1–28, 2018. 3

[8] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. *arXiv preprint arXiv:2205.08534*, 2022. 7

[9] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021. 3

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 1, 2, 3, 5, 6, 7

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 3, 5

[12] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018. 2, 3

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, 2013. 1

[14] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30, 2021. 3

[15] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 3

[16] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5356–5364, 2019. 3

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv:2111.06377*, 2021. 1, 3, 5, 6, 7, 8

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *CVPR*, 2020. 3

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 5

[20] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006. 1

[21] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 2, 3, 4

[22] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 4

[23] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. 2021. 3

[24] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3

[25] Doyeon Kim, Woonghyun Ga, Pyungwhan Ahn, Donggyu Joo, Sehwan Chun, and Junmo Kim. Global-local path networks for monocular depth estimation with vertical cutdepth. *arXiv preprint arXiv:2201.07436*, 2022. 5

[26] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31, 2018. 3

[27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[28] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 3

[29] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of*

*the IEEE/CVF conference on computer vision and pattern recognition*, pages 2661–2671, 2019. 3

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. 2012. 1, 3

[31] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *International Conference on Learning Representations (ICLR)*, 2022. 3

[32] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation, 2022. 7

[33] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets, 2017. 3, 5, 8

[34] Yanghao Li, Hanzi Mao, Ross B. Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *ArXiv*, abs/2203.16527, 2022. 4

[35] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 4

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2, 3, 5

[37] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022. 2, 3, 7

[38] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. pages 10012–10022, 2021. 4, 5

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[40] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 3

[41] Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022. 3

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 1, 2, 3, 4, 8

[43] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021. 3

[44] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[45] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 2

[46] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 3

[47] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can CLIP benefit vision-and-language tasks? In *International Conference on Learning Representations*, 2022. 3

[48] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 3

[49] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *ECCV (5)*, 7576:746–760, 2012. 2, 5

[50] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 3

[51] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 2, 3, 4, 7

[52] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022. 2

[53] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020. 3

[54] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models, 2021. 2

[55] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 5

[56] Zhenda Xie, Zigang Geng, Jingcheng Hu, Zheng Zhang, Han Hu, and Yue Cao. Revealing the dark secrets of masked image modeling. *arXiv preprint arXiv:2205.13543*, 2022. 3, 5, 7

[57] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level

consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021. 3

[58] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3

[59] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014. 3

[60] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 5

[61] 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. 7

[62] Mengyao Zhai, Lei Chen, Frederick Tung, Jiawei He, Megha Nawhal, and Greg Mori. Lifelong gan: Continual learning for conditional image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2759–2768, 2019. 3

[63] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. 2

[64] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection, 2022. 7

[65] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding, 2022. 7

[66] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal on Computer Vision*, 2018. 2, 3, 5

[67] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 3, 5