On the Surprising Effectiveness of Attention Transfer for Vision Transformers

Alexander C. Li[*]
Carnegie Mellon University

Yuandong Tian
FAIR

Beidi Chen
Carnegie Mellon University

Deepak Pathak
Carnegie Mellon University

Xinlei Chen
FAIR

Abstract

Conventional wisdom suggests that pre-training Vision Transformers (ViT) improves downstream performance by learning useful representations. Is this actually true? We investigate this question and find that the features and representations learned during pre-training are not essential. Surprisingly, using only the attention patterns from pre-training (*i.e.*, guiding how information flows between tokens) is sufficient for models to learn high quality features from scratch and achieve comparable downstream performance. We show this by introducing a simple method called attention transfer, where only the attention patterns from a pre-trained teacher ViT are transferred to a student, either by copying or distilling the attention maps. Since attention transfer lets the student learn its own features, ensembling it with a fine-tuned teacher also further improves accuracy on ImageNet. We systematically study various aspects of our findings on the sufficiency of attention maps, including distribution shift settings where they underperform fine-tuning. We hope our exploration provides a better understanding of what pre-training accomplishes and leads to a useful alternative to the standard practice of fine-tuning. Code to reproduce our results is at https://github.com/alexlioralexli/attention-transfer.

1 Introduction

Pre-training has emerged as a dominant paradigm in machine learning and has significantly improved performance on a variety of tasks [27, 11, 2, 22]. In computer vision in particular, self-supervised representation learning methods [21, 6, 4, 22] and weakly supervised methods [40, 45] have enabled learning from large amounts of images. It is widely accepted that these methods work because they teach models useful features that are relevant for downstream tasks. But is this story actually true? Perhaps there is another capability learned during pre-training that is sufficient to explain its benefits.

In this paper, we present an alternative explanation: pre-training teaches the model how information should be routed between tokens. We specifically focus on Vision Transformers (ViT) [12], not only because they are the most popular architecture for scaling, but also because Transformers explicitly *decouple* this information flow. Inter-token communication is solely fulfilled by attention, while the remaining bulk of computation are intra-token operations that are applied to each token independently. In contrast, other architectures such as ConvNets [33, 20] simultaneously expand the receptive fields and extract the features, making it difficult to isolate the effect of information flow. We hypothesize that the features computed by the intra-token operations are not essential to explain the benefits of pre-training, and that the pre-trained attention maps are typically sufficient for downstream tasks.

We test our hypothesis by introducing a new set of methods called attention transfer. Concretely, we treat a pre-trained ViT as the teacher and train a student model for downstream tasks while

[*]Work done during an internship at FAIR.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

transferring only the attention patterns from the teacher. In contrast to the common fine-tuning paradigm of transferring all the weights (which mixes the effect of features and attention maps), *only* the inter-token flow is transferred. In this way, the student must learn features from scratch, while isolating the benefits of the attention maps learned during pre-training.

We study two types of attention transfer. The first is *Attention Copy*, which directly "copy-and-pastes" the attention maps. The learning is fully decoupled, as inter-token computation is entirely from the teacher, and the student only learns intra-token patterns routed by the teacher's attention maps. This is well-suited as a scientific probe, but is less practical since both networks need to be forwarded during the inference. The second is *Attention Distillation*, where the student simply distills attention patterns from the teacher, whose attention maps are no longer used after training. This is practical, but also helps identify the importance of the teacher's inter-token information flow.

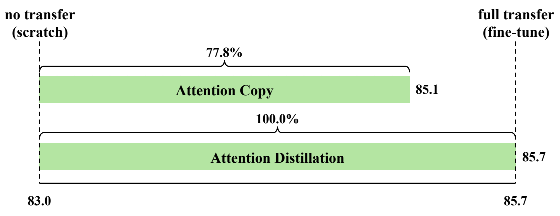

Figure 1: **Using only attention is sufficient for full performance**. By copying the attention maps (top) from a MAE [22] pre-trained ViT-L [12], a ViT-L can reach a top-1 accuracy of 85.1 on ImageNet-1K [10] – recovering 77.8% of the gap between no transfer (training from scratch, 83.0) and full transfer (fine-tuning all the weights, 85.7). Distilling attention maps (bottom) can even *fully* match MAE weight tuning while only transferring the inter-token flow.

While both attention transfer variants are straightforward, we find them *highly effective*. Figure 1 illustrates this with a ViT-L [12] pre-trained using Masked Autoencoding (MAE) [22]. Compared to no transfer (training from scratch) and full transfer (fine-tuning all the MAE weights), Attention Copy can close most of the gap in performance, whereas Attention Distillation can *match* the fine-tuning accuracy on ImageNet-1K classification [10]. This is achieved by only transferring the inter-token flow from the same model. Furthermore, since attention transfer requires the student to learn features from scratch, those features are significantly different from the teachers' (Figure 5) and improve ImageNet-1K accuracy score to 86.3 (+0.6) when ensembled with the teacher (Figure 6).

To summarize, we make the following contributions:

- **Detailed analysis on the sufficiency of attention maps**. We find that solely using the pre-trained attention patterns is typically *sufficient* to achieve the same downstream accuracy as fine-tuning on ImageNet-1K. Furthermore, we observe practical benefits, as ensembling with attention transfer significantly improves ImageNet performance. This calls into question the commonly-believed story that pre-training is only about feature learning. While our main observation is robust w.r.t. different models and pre-training methods, we *do find settings where pre-trained features are indeed necessary* to realize the full gains from pre-training. Our bare-minimum solution for attention transfer is more affected by data distribution shifts compared to weight tuning. Section 4 presents extensive analyses to better understand the behaviors of attention transfer. They are i) partial transfer with a subset of layers or heads; ii) variants of our method that transfer other attention-related activations; and importantly, iii) various ways to verify that the student is *not* just re-learning the teacher model. Section 5 systematically tests how well our findings apply across a variety of pre-training and fine-tuning datasets, pre-training methods, model sizes, and tasks.

- **Attention transfer methods**. We introduce Attention Copy and Attention Distillation, which are methods to train a ViT on a downstream task while utilizing only the attention maps of a pre-trained teacher ViT. These methods help us understand the role of the features versus the attention patterns learned during pre-training. With further research, attention transfer could offer a potential alternative to the decade-long practice of fine-tuning pre-trained vision models [16, 12, 22]. Nearly all aspects of the fine-tuning pipeline have been thoroughly examined, suggesting a probable saturation of recipes. Weight sharing can also face security risks (*e.g.*, white-box attacks [17]). We hope our systematic examination of attention transfer sheds new light on how to leverage pre-trained ViTs, and will help establish this approach as an effective alternative when weight transfer is less applicable.

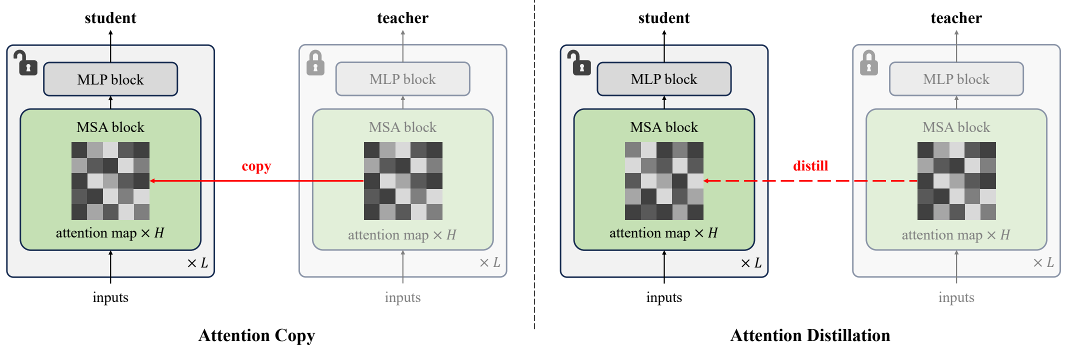

Figure 2: Two types of **Attention transfer** for Vision Transformers. **Attention Copy** (left): We simply "copy-and-paste" the attention maps from a pre-trained teacher model to a randomly initialized student one. Other weights of the student are then trained via supervised learning. This fully decouples inter-token learning (from the teacher) and intra-token learning (in the student); but is less practical. **Attention Distillation** (right): The student computes its own attention maps, with an additional cross-entropy loss to distill patterns from the teacher during training. The teacher is no longer used during inference. H: number of heads; L: number of Transformer layers.

2 Attention Transfer

2.1 Attention Preliminaries

To work with a Vision Transformer (ViT) [12], an image is first "patchified" into N tokens. Their intermediate activations are represented as a sequence $X=[x_1, x_2, \cdots, x_N]^\top$, $x_i \in \mathbb{R}^C$, where C is the embedding dimension. The self-attention [57] mechanism mainly introduces three learnable parameters $W_q, W_k, W_v \in \mathbb{R}^{C \times C/H}$ (H is the number of heads). $Q=XW_q$ is often referred to as the queries, $K=XW_k$ as the keys, and $V=XW_v$ as the values. Then the attention function is defined as:

$$f_{\text{attn}} = \underbrace{\text{softmax}\left(QK^\top\right)}_{\text{attention map}} V, \tag{1}$$

where the softmax function is computed per query for the *attention map*. Attention maps determine how the values from other tokens are aggregated, and with multiple heads, each token uses multiple attention distributions within the same Multi-headed Self-Attention (MSA) block.

For an L-layer Transformer, MSA blocks are interleaved with MLP blocks, and each Transformer layer contains one of each block type. Most operations are *intra-token computations*, which are applied independently to each token: value and projection matrices, normalization layers [1], and MLPs. The only *inter-token computation* is applying the attention map softmax$(QK^\top)$, which is the only way for information to flow between tokens. Transformers are unique because their inter- and intra-token computations are *decoupled*; however, the relative importance of each type of operation is not well understood, and Transformers are typically trained by *jointly* fine-tuning all the weights.

Deviating from the common practice of joint weight tuning, we propose two attention transfer methods with the goal of exploring decoupled training for ViTs, described next.

2.2 Attention Copy

In this setup, we utilize two separate networks: a pre-trained teacher network that *only* does a forward pass to compute its attention maps, and a student network that directly copies the attention maps from the teacher but computes all of the other activations. The student's weights are randomly initialized and trained via back-propagation, while the teacher's weights are kept frozen. This setting fully isolates the attention maps from the features that they are applied to, and thus is ideal for measuring the utility of pre-trained attention patterns when the student network learns to perform other tasks (*e.g.*, image classification).

We call this method *Attention Copy*, as we "copy-and-paste" the attention maps from teacher to student. Figure 2 (left) shows a diagram of this approach. Note that it requires forward passes through both the teacher and student networks during the inference time. Given the extra computation, Attention Copy is not meant to be an entirely practical method. We mitigate this issue next.

2.3 Attention Distillation

In *Attention Distillation*, the teacher network is only utilized at the training time. Given each training example, we forward both networks in parallel, with the student also computing its own attention maps. But besides the task-driven loss, we also enforce a distillation loss between student's attention maps and the teacher's counterparts as (soft) targets. Formally, using $Q_s {K_s}^\top$ for the student and $Q_t {K_t}^\top$ for the teacher, the loss is then defined as:

$$\mathcal{L}_{\text{dist}} = \mathcal{H}\left[\text{softmax}(Q_s {K_s}^\top), \text{softmax}(Q_t {K_t}^\top)\right], \tag{2}$$

where $\mathcal{H}$ computes the cross entropy. As there can be multiple heads and layers in a Transformer, we simply sum up all the losses from wherever attention distillation is applied. Again, the student is trained via back-propagation. Figure 2 (right) shows the diagram of Attention Distillation.

Compared to Attention Copy, Attention Distillation is much more practical. After training, the teacher is no longer needed, and the student can be used as a standalone model. Compared to training ViTs from scratch, the only addition is the distillation loss, meaning most of the optimization (*e.g.*, learning rate, momentum) and regularization (*e.g.*, weight decay, dropout rate [50]) hyperparameters can follow the scratch recipe with minimal adjustments. It does introduce a new hyperparameter λ, which weights the distillation loss and balances it with the task loss.

Attention Distillation can be viewed as a form of generalized knowledge distillation, but it has several key differences from the design proposed by Hinton et al. [26]. Attention Distillation trains the student to match the teacher's intermediate attention maps, not the final teacher output. This gives the flexibility of distilling from models trained on any task, not just models trained on the same final task. This property is well-suited for today's "pre-train and transfer" paradigm, where the pre-training task (*e.g.*, reconstruction) and the downstream task (*e.g.*, classification) are usually different. However, Attention Distillation does add the constraint that the architecture needs to compute attention maps. We leave experimenting on this idea for other architectures as future work.

Overall, while fancier designs can be used for both Attention Copy and Attention Distillation, we choose to keep them simple for cleaner assessments of their effectiveness.

2.4 Connection to Transformer Training Dynamics

Our investigation is also linked to recent attempts to theoretically understand the training dynamics of Transformers. Specifically, the inter-token flow encoded in the pre-trained attention maps can be regarded as a discovered latent *hierarchy* from the dataset. Self-attention can quickly capture frequently co-occurring token pairs [31, 52]. However, more occasional co-occurrences need to be explained by the top-level hierarchy, rather than directly learned in the lower levels [53]. This is due to many potential spurious correlations [30], especially in the over-parameterized setting. Transferring attention maps from a trained teacher reduces these spurious inter-token correlations, so the student can focus on intra-token learning (*i.e.*, computing useful features).

3 Main Results

As featured in Figure 1, attention transfer is highly effective despite its simplicity. Specifically, we demonstrate this with a ViT-L [12] pre-trained with Masked Autoencoding (MAE) [22] for ImageNet-1K classification [10]. Note that this is the *signature* result that established MAE's effectiveness for pre-training: compared to a ViT-L trained from scratch (with an accuracy score of 83.0), fine-tuning the MAE pre-trained on the same dataset results in a significant improvement to 85.7.[1]

For attention transfer, we use the same pre-trained MAE model as our teacher, and since scratch training can be viewed as no transfer, and fine-tuning weights transfers all the weights, the above two results serve as natural lower and upper bounds for the effectiveness of our attention transfer methods. We make two observations (from Table 1 and Figure 1).

[1]If not otherwise specified, our results are based on our faithful reimplementation of the official code in JAX.

method	acc.
scratch	83.0
fine-tune	85.7
attn. copy	85.1
attn. copy from fine-tuned	85.6
attn. distill	85.7
ensemble attn. distill + fine-tune	**86.3**

Table 1: **Main results**. We show that the pre-trained attention patterns are *sufficient* to match fine-tuning accuracy on ImageNet. Attention Copy closes most of the gap, and Attention Distillation achieves the same accuracy.

transfer target	acc.
Q	85.6
K	84.4
V	84.4
Q, K	85.1

Table 2: **Transfer other attention activations**. We test copying alternative attention activations other than the attention map – softmax$(QK^\top)$. All alternatives do better than training from scratch, and transferring queries Q actually does better than transferring the attention map.

Attention Copy can largely close the gap between scratch training and full weight fine-tuning

We report an accuracy of 85.1 with Attention Copy. This has largely closed the gap between scratch and full weight tuning (to be precise, 77.8% of the 2.7 percentage point gap). This is surprising, since the teacher's attention maps are frozen after pre-training for a different task (image reconstruction), and the student must learn everything else (the intra-token operations) completely from scratch.

As another upper-bound, we also experimented with Attention Copy from the *fine-tuned* model. This reaches an accuracy score of 85.6 – almost matching the teacher's performance (85.7), suggesting that adapting attention maps to the specific task is still helpful, but not crucial, especially as MAE pre-training is performed on the same data distribution.

Attention Distillation can match fine-tuning performance

Even more surprisingly, we find Attention Distillation can achieve 85.7 – *on par* with fine-tuning the ViT-L weights from MAE. Since Attention Distillation and weight tuning both result in the same-sized model, which requires the same compute budget for inference, this result suggests Attention Distillation can be an effective drop-in replacement for weight tuning when the latter is less applicable (*e.g.*, if weight sharing poses security risks, we can instead send the teacher's attention maps).

We hypothesize that distillation is better than copy because the student can choose how closely it matches the teacher attention maps, to better suit the task objective. This is also supported by the 85.6 accuracy of copying from fine-tuned MAE, which has the correct task-specific attention maps.

4 Analysis

Next, we provide extensive analyses to better understand the effectiveness of attention transfer. Broadly speaking, the explorations are driven by the following two questions:

(i) How important are different activations, layers, and heads? (Section 4.1)
(ii) Is attention transfer re-learning everything from the teacher? (Section 4.2)

4.1 Variants of Attention Transfer

We study four variants of attention transfer. We use Attention Copy within this section, since it is a fully-decoupled setting well-suited for scientific analysis.

Transfer a subset of Q, K and V. A natural alternative to transferring attention maps is to transfer different activations that come with self-attention (Eq. 1), namely queries Q, keys K, or values V. Without loss of generality, if we transfer the teacher's Q, the student will compute its own K and V and use them normally. Note that transferring both Q and K is equivalent to transferring the map softmax$(QK^\top)$. Table 2 shows that transferring Q works surprisingly well, and is actually better than transferring the attention map.

We suggest that copying Q gives the model the flexibility to deviate from the teacher attention maps and use attention patterns that are better suited for the downstream task. This is supported by the fact

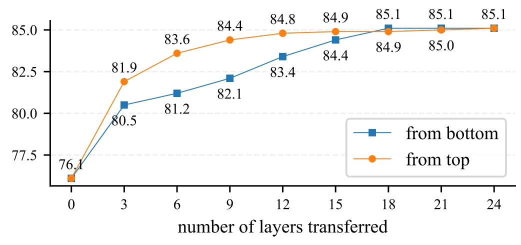

Figure 3: **Copy a subset of layers.** By default, all 24 ViT-L layers are transferred. Here we only transfer a subset, and find: more layers always helps; and attention maps from top layers are more beneficial than those from bottom layers.

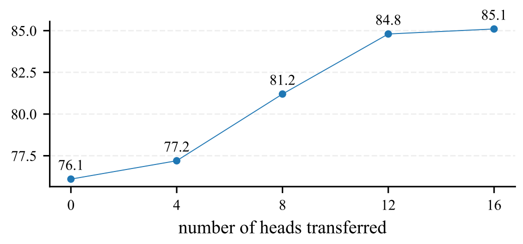

Figure 4: **Copy a subset of heads.** The pre-trained ViT-L has 16 heads in each MSA block. By default, all of them are transferred. Here we only transfer a subset, and find more heads helps in general, but performance saturates at 12 heads.

that copying Q and Attention Copy from the fine-tuned model both achieve the same accuracy of 85.6. Appendix B.3 dives deeper into this hypothesis and finds that the attention maps for copying Q are similar to the teacher's but less constrained than they are in other transfer methods. While more investigation could be done in future work, our findings suggests that the queries Q are more important than the keys, which is consistent with previous findings in text sequence modeling where the number of keys and values per layer can be significantly reduced [49].

Finally, we test whether distilling Q could outperform full Attention Distillation. However, Table 3 shows that Q distillation does significantly worse. We hypothesize that this is because it is harder for the student to learn useful keys K while Q is still being learned, and because Attention Distillation already has the flexibility to adjust its attention maps to suit the downstream task.

method	acc.
attn. distill	85.7
Q distill	81.3

Table 3: Attention Distillation outperforms Q distillation.

Partial transfer: layers. We next change the number of Transformer layers transferred, aiming to identify which layers are more valuable from the teacher. The baseline transfers all the layers. In Figure 3, we try transferring attention maps only from the first or last layers. For the remaining layers, the student learns to compute its own attention maps.

We make several observations: i) We find transferring more layers is always more helpful. This is a bit surprising, as one may expect attention patterns optimized for MAE's patch-wise reconstruction task to be sub-optimal for a high-level recognition task like classification. ii) We find transferring the later attention maps is generally better. In fact, performance roughly saturates when transferring the last 15 attention maps out of a total of 24. This indicates that ViTs are capable of learning good low-level representations, as long as they are told how to combine these features into higher-level ones; but not vice versa. This reinforces the theory from Tian et al. [53] that guidance on top-level hierarchy is more important, as there are many more possibilities, and attention transfer can reduce possible spurious correlations in the lower levels.

Partial transfer: heads. Finally, we switch back to transferring all the layers, but change the number of heads copied from each MSA block. Specifically, instead of copying the attention map from every head, we can selectively choose to use a subset of the teacher's heads. The student can then compute its own attention patterns for the unused heads. Figure 4 shows the effect of transferring fewer heads for each layer. Performance improves as we do attention transfer with more heads, though performance almost saturates at 12 out of 16 heads. Note that we simply follow a naïve selection strategy and use the first set of heads; more advanced strategies based on diversity or activation magnitude can potentially improve the robustness as we reduce the number of heads.

4.2 Are We Re-Learning the Teacher?

Since attention transfer provides a significant amount of information from the teacher (ViT-L attention maps have about 10M activations total per image, see Appendix A.1 for detailed calculations), a natural question is whether the student performs well because it simply relearns the same representations as the teacher. We thoroughly test this hypothesis on different aspects of the student model.

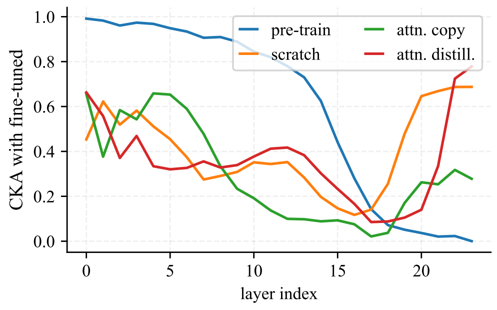

Figure 5: **CKA representation similarity to the fine-tuned model.** We use CKA [32] to measure the layer-wise similarity between representations learned in different models against the fine-tuned MAE model. Higher means more similar. We find that attention transfer methods are quite *dissimlar* to the fine-tuned model, with roughly the same CKA as an independent scratch model.

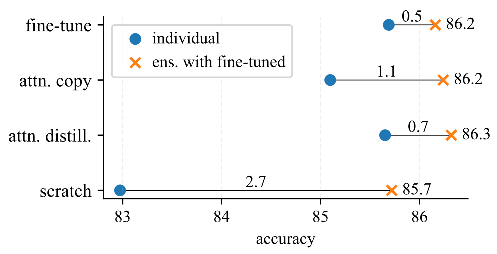

Figure 6: **Ensemble accuracy with the fine-tuned model**. We plot the accuracy of ensembling our attention transfer models and a fine-tuned MAE. This measures model prediction similarity with the fine-tuned model. The ensemble yields notable accuracy gains over the base model, reaching up to 86.3. This is even higher than ensembling two separate fine-tuned MAE models[2] (86.2) and indicates that the attention transfer models are *less* correlated with the fine-tuned model.

Representation similarity. One way that the student can reproduce the teacher is by learning the same intermediate features. We measure this using Centered Kernel Alignment (CKA) [32], a similarity measure for representations that has been shown to be effective even for networks trained with different initializations or architectures. CKA is a layer-wise comparison that ranges from 0 (completely dissimilar) to 1 (identical) and is invariant to orthogonal linear transformations and isotropic scaling of the features. Figure 5 shows the CKA between our fine-tuned model and our attention transfer methods. We also show the pre-trained model and a ViT-L trained from scratch for reference. We compute CKA with respect to the fine-tuned model, even though we copy or distill from the pre-trained MAE model, since the features change significantly during fine-tuning to become more task-specific. Overall, Attention Copy and Attention Distillation are both quite *dissimilar* to the fine-tuned MAE model, following the same similarity trend as the scratch model. Our sanity check passes, as CKA shows that pre-trained and fine-tuned MAE have very similar representations in the early layers. This is expected since fine-tuning with layer-wise learning rate decay [8] means the earliest layers change very little during fine-tuning.

Prediction similarity and ensembling. Our CKA analysis may not catch some similarity of the intermediate representations, as CKA does not detect all forms of the same information (*e.g.*, invertible nonlinear transforms) [32]. Since intermediate representations may not tell the full story, we also examine the similarity of the network outputs. We measure this using network ensembling: given softmax predictions p_{ft} from the fine-tuned model and p_{other} from another model, we test the accuracy of their average $(p_{\text{ft}} + p_{\text{other}})/2$. The more independent the model predictions are, the higher their ensemble accuracy is. Figure 6 compares accuracy before and after ensembling with the fine-tuned model. Attention transfer is dissimilar enough to achieve high ensemble accuracy, and ensembling Attention Distillation with a fine-tuned MAE achieves 86.3, +0.6 over the fine-tuned MAE model.

Finally, Appendix B.4 visualizes the attention maps learned by Attention Distillation and shows that they match for distilled attention blocks but are drastically different for layers that are not distilled.

5 Generalization and Limitations

In this section, we test how well our findings on attention transfer apply across a variety of pre-training and fine-tuning datasets, pre-training methods, model sizes, and tasks.

[2]New result: we fine-tune two MAE models (85.8, 86.0) from a shared pre-trained MAE initialization, then ensemble them to achieve 86.2. These use a *new* pre-trained MAE model, as the original MAE checkpoint used in this paper was accidentally deleted. Note that these fine-tuned models are *stronger* than the original fine-tune (85.7), yet underperform the ensemble of the original fine-tuned MAE and the Attention Distillation model.

pre-training data	tune	copy
ImageNet	85.7	85.1
ImageNet-22K	85.5	84.4
COCO	85.2	83.1

Table 4: **Different pre-training datasets**. We pre-train MAE on more datasets, and then either fine-tune or do copy for ImageNet-1K classification. Attention transfer works well when the data distribution stays stable, but its effectiveness is more negatively affected by distribution shifts.

eval. data	scratch	tune	copy	distill
iNat 2017	49.7	75.9	69.1	69.3
iNat 2018	64.3	79.9	71.8	74.1
iNat 2019	66.2	83.8	77.9	80.0

Table 5: **Long-tail recognition on iNaturalist**, with ImageNet-1K pre-trained MAE. We tune weights or do attention transfer on iNaturalist, and we again find attention transfer is worse than tuning weights when the pre-training dataset is different from the downstream dataset.

out-of-distribution evaluation	scratch	tune	copy	distill
ImageNet-A [24]	32.0	56.5	48.9	54.3
ImageNet-R [23]	51.9	59.6	57.5	56.8
ImageNet-S [60]	38.0	45.2	43.1	42.9
ImageNet-V2 [47]	72.4	76.4	75.5	75.9

Table 6: **Out-of-distribution robustness**. We take two models that achieve the same accuracy on ImageNet-1K (fine-tuned and distilled), and evaluate them on a suite of distribution shifts. Attention Distillation does well when the distribution is close (*e.g.*, on ImageNet-V2), but loses the mild "effective robustness" that fine-tuned MAE has been found to have [14].

5.1 Pre-training and fine-tuning datasets

So far, we have focused on a MAE model pre-trained and evaluated on ImageNet-1K. What happens if we pre-train or evaluate on different datasets? We first test this by pre-training MAE ViT-L models on two new datasets: ImageNet-22K [10] and COCO [37]. These have substantially different dataset bias [54] from ImageNet, across axes like appearance, class balance, and diversity. Table 4 shows that the resulting MAE models maintain relatively good performance when fine-tuning on ImageNet-1K, with a maximum drop of at most 0.5. However, Attention Copy accuracy drops more, losing as much as 2.1. We see a similar phenomenon in Table 5 where we use a MAE pre-trained on ImageNet and transfer to the iNaturalist datasets [56]. Again, when the pre-training dataset does not match the transfer dataset, Attention Copy accuracy drops significantly. We hypothesize that the frozen teacher's attention maps are ill-suited for the fine-tuning dataset, which limits the performance.

5.2 Out-of-distribution robustness

One notable aspect of a standard fine-tuned MAE model is that it shows slight "effective robustness," *i.e.*, it achieves slightly better out-of-distribution (OOD) accuracy than expected based on its in-distribution (ID) accuracy [14]. We test whether Attention Distillation, which achieves the same ID accuracy, has the same benefits OOD. Table 6 shows that Attention Distillation still does quite well, but has lower accuracy than fine-tuned MAE on all 4 distribution shifts we tried. These results indicate that the attention maps do not account for the full robustness benefits, and that the features learned by MAE during pre-training are helpful OOD even if they are not ID.

5.3 Pre-training methods

We have so far focused on MAE, a reconstruction-based pre-training method. We now check whether attention transfer still works if the teacher has been pre-trained with a different algorithm. Specifically, we test MoCo-v3 [7], a self-supervised contrastive learning approach, and FLIP [36], which does image-text contrastive learning. Table 7 shows that Attention Copy still achieves most of the performance benefits for each pre-training method. *Impressively, ViT-L is even able to reach 86.6 by just transferring attention maps from FLIP*. This confirms that learning the proper attention patterns is indeed a significant bottleneck during learning. Note that the FLIP model we used is pre-trained on LAION-2B [48], yet its effectiveness is less affected by distribution shifts to ImageNet-1K.

pre-training method	tune	copy	distill
MAE [22]	85.7	85.1	85.7
MoCo-v3 [7]	84.0	82.5	83.3
FLIP [36]	87.4	86.6	86.1
DINO[†] [4]	83.2	82.3	82.8
none	83.0	72.7	76.3

Table 7: **Different pre-training methods.** Attention transfer works for all pre-training methods, even reaching 86.6 with a FLIP teacher. As a sanity check, transferring from a randomly initialized ViT significantly hurts. [†]DINO is ViT-B.

model	scratch	tune	copy	distill
ViT-B	82.5	83.6	82.0	83.4
ViT-L	83.0	85.7	85.1	85.7
ViT-H	83.0	86.9	86.1	86.3

Table 8: **Different model size** with MAE pre-trained on ImageNet-1K. Similar to weight tuning, the classification accuracy of attention transfer scales well as we vary the model size, while scratch training saturates.

metric	scratch	tune	distill
AP^{box}	39.1	46.3 (+7.2)	43.6 (+4.5)
AP^{mask}	34.6	40.6 (+6.0)	38.7 (+4.1)

Table 9: **Object detection results** on COCO with a MAE ViT-B pre-trained on COCO. Attention transfer achieves a majority of the gains of pre-training in this setting as well.

5.4 Model size

We test whether attention transfer works across model sizes. For all experiments so far, we have used ViT-L; here, we try Attention Copy from a smaller (ViT-B) and larger (ViT-H) model, both pre-trained with MAE. Table 8 shows that Attention Copy continues to improve with scale, even reaching 86.1% accuracy with ViT-H. It can do this even though scratch model performance already saturates at the ViT-L size. Again, this indicates that models need proper inter-token routing in order to learn good features that generalize. Otherwise, they cannot properly utilize increased model capacity.

5.5 Object Detection

Finally, we examine the performance of attention transfer in the standard ViTDet pipeline [35] for COCO object detection. We compare training from scratch against fine-tuning and attention transfer from a MAE ViT-B pre-trained on COCO, which is done to mitigate the effect of distribution shift. For fair comparisons, we use a 448×448 input to remove the effect from window attention and positional embedding interpolation, and remove the effect of relative positional embeddings. Table 9 shows that Attention Distillation recovers a majority of the gains from pre-training in this dense prediction setting as well. Based on Table 8, we anticipate that the gap between fine-tuning and attention transfer will decrease with ViT-L, but we are limited by computational resources.

6 Related Work

Structure in attention patterns. Previous works have studied the attention patterns of pre-trained vision transformers [59, 63, 43]. These works present these differences only as qualitative observations, whereas we are able to isolate the attention patterns and show that they are causally responsible for most of the differences in fine-tuning performance. Other methods, such as Lora [28] or Prompt-to-Prompt [25], do rely on the importance of high quality attention patterns within pre-trained networks, but they also utilize pre-trained features and do not provide our insight that *these features are typically unnecessary* for the tasks we examine. Trockman and Kolter [55] observe diagonal structure within the product of attention layer weights in a trained supervised network. They show that initializing the weights with this structure moderately improves accuracy for small models early in training. Zhang et al. [68], in the language domain, find that pre-trained BERT models improve length generalization on particular synthetic tasks. They attribute this to the attention patterns of a few, specific heads and show that hardcoding these patterns into the network achieves the same benefit. Our work is complementary and emphasizes the importance of attention maps over features.

Decoupling inter- and intra-token operations. GLoMo [64] also attempts to decouple features from the way they should be combined. They use unsupervised pre-training to train a network to output a graph, which is later used to combine task-specific features. We find that there is no need to develop a specialized architecture to achieve this – Vision Transformers *already do this naturally*.

Knowledge Distillation Knowledge distillation is a popular framework for training small, high-performing student networks [26]. Knowledge distillation methods typically add a loss to encourage the student network to match the teacher network's logits, but variants often use other feature statistics as targets, such as the final representations [41, 13], intermediate feature similarities [19], or intermediate feature magnitudes [66, 34]. This last approach has also previously been called "attention transfer," but their method is quite different and actually refers to distilling spatial activation magnitudes in ConvNets. These knowledge distillation approaches all assume that students need to be explicitly taught the right features. In contrast, our analysis with attention transfer shows that attention maps are sufficient to recover all of the gains from pre-training. Some papers have used attention distillation as an auxiliary loss to help a smaller model learn the teacher outputs more effectively [62, 61]. However, these only consider transferring the same function across model sizes, instead of transferring knowledge from a pre-trained model to a different downstream task.

Connection to the lottery ticket hypothesis The lottery ticket hypothesis [15] suggests that large, dense neural networks contain small, sparse subnetworks ("winning tickets") that, when trained from scratch, can match or even outperform the performance of the original dense network. This is particularly interesting because these sparse subnetworks maintain their performance only with their original initialization values; the strength of their connections between *neurons* is special in some way. Our findings draw a parallel, indicating that the connections between *patches*, controlled solely by the attention patterns, are similarly special within pretrained ViTs. Frankle and Carbin [15] further conjecture that overparameterization improves performance because larger models contain exponentially more sparse subnetworks in superposition and are thus more likely to contain a "winning ticket" – a hypothesis supported by subsequent empirical and theoretical work [46, 44, 42, 3]. However, this phenomenon does not arise in our setting with ViT attention patterns, since there are only a handful of attention maps per layer (rather than thousands of neurons). Consequently, good attention patterns are unlikely to appear by chance and must instead be learned during pre-training. We hope that a new model architecture that efficiently combines many more attention maps per layer can address this limitation and learn better from scratch than existing ViTs.

7 Conclusion

Even as Transformers have surged in popularity, the way we use them has remained stagnant: pre-train, then fine-tune the weights. In this work, we present attention transfer, a simple alternative to ViT fine-tuning that decouples intra-token operations (how to extract more usable features for each token) from inter-token operations (how those features should be combined). Our key finding is that the attention patterns (inter-token operations) are the key factor behind much of the effectiveness of pre-training – our Attention Distillation method *completely matches fine-tuning* on ImageNet-1K. We do find some limitations: attention transfer does not work well if the pre-training and transfer datasets are different, and it loses a bit of OOD robustness. Nevertheless, our findings provide insights into the role of attention in pre-trained ViTs, and we hope future work fixes attention transfer's shortcomings and explores the advantages of this new transfer method.

Some directions for future work are particularly interesting. First, a deeper investigation of the queries Q could help us better understand their importance and potentially yield better transfer strategies. Second, attention transfer eliminates the need for tricks that fine-tuning requires, such as layerwise learning rate decay. Layerwise learning rate decay adds the prior that early layers should change less compared to later layers. However, this prior may be overly restrictive for next-generation models, since it prevents early features from changing, and getting rid of it could open up new opportunities. Finally, attention maps could potentially be transferred more easily across model sizes. Pre-training a smaller model and transferring its attention patterns to a larger downstream model could be more practical than the current practice of fine-tuning.

Acknowledgments and Disclosure of Funding

AL is supported by the NSF GRFP DGE1745016 and DGE2140739 and performed the work during an internship at FAIR.

References

[1] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[3] R. Burkholz. Most activation functions can win the lottery without excessive depth. *Advances in Neural Information Processing Systems*, 35:18707–18720, 2022.

[4] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[5] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[7] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised Vision Transformers. In *ICCV*, 2021.

[8] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[9] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[13] Q. Duval, I. Misra, and N. Ballas. A simple recipe for competitive low-compute self supervised vision models. *arXiv preprint arXiv:2301.09451*, 2023.

[14] A. Fang, G. Ilharco, M. Wortsman, Y. Wan, V. Shankar, A. Dave, and L. Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). In *ICML*, 2022.

[15] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[18] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv:1706.02677*, 2017.

[19] Z. Hao, J. Guo, D. Jia, K. Han, Y. Tang, C. Zhang, H. Hu, and Y. Wang. Learning efficient vision transformers via fine-grained manifold distillation. In *NeurIPS*, 2022.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[22] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[23] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021.

[24] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. In *CVPR*, 2021.

[25] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[26] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[27] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[29] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

[30] P. Izmailov, P. Kirichenko, N. Gruver, and A. G. Wilson. On feature learning in the presence of spurious correlations. *Advances in Neural Information Processing Systems*, 35:38516–38532, 2022.

[31] S. Jelassi, M. Sander, and Y. Li. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 35:37822–37836, 2022.

[32] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.

[33] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989.

[34] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021.

[35] Y. Li, H. Mao, R. Girshick, and K. He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022.

[36] Y. Li, H. Fan, R. Hu, C. Feichtenhofer, and K. He. Scaling language-image pre-training via masking. In *CVPR*, 2023.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[38] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[39] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[40] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

[41] K. Navaneet, S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash. Simreg: Regression as a simple yet effective tool for self-supervised knowledge distillation. *arXiv preprint arXiv:2201.05131*, 2022.

[42] L. Orseau, M. Hutter, and O. Rivasplata. Logarithmic pruning is all you need. *Advances in Neural Information Processing Systems*, 33:2925–2934, 2020.

[43] N. Park, W. Kim, B. Heo, T. Kim, and S. Yun. What do self-supervised vision transformers learn? *arXiv preprint arXiv:2305.00729*, 2023.

[44] A. Pensia, S. Rajput, A. Nagle, H. Vishwakarma, and D. Papailiopoulos. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. *Advances in neural information processing systems*, 33:2599–2610, 2020.

[45] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[46] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, and M. Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11893–11902, 2020.

[47] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

[48] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022.

[49] N. Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

[50] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, pages 1929–1958, 2014.

[51] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[52] Y. Tian, Y. Wang, B. Chen, and S. Du. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *NeurIPS*, 2023.

[53] Y. Tian, Y. Wang, Z. Zhang, B. Chen, and S. Du. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. *ICLR*, 2024.

[54] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[55] A. Trockman and J. Z. Kolter. Mimetic initialization of self-attention layers. *arXiv preprint arXiv:2305.09828*, 2023.

[56] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The iNaturalist species classification and detection dataset. In *CVPR*, 2018.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[58] S. Venkataramanan, A. Ghodrati, Y. M. Asano, F. Porikli, and A. Habibian. Skip-attention: Improving vision transformers by paying less attention. *arXiv preprint arXiv:2301.02240*, 2023.

[59] M. Walmer, S. Suri, K. Gupta, and A. Shrivastava. Teaching matters: Investigating the role of supervision in vision transformers. In *CVPR*, 2023.

[60] H. Wang, S. Ge, Z. Lipton, and E. P. Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019.

[61] K. Wang, F. Yang, and J. van de Weijer. Attention distillation: self-supervised vision transformer students need more guidance. *arXiv preprint arXiv:2210.00944*, 2022.

[62] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

[63] Z. Xie, Z. Geng, J. Hu, Z. Zhang, H. Hu, and Y. Cao. Revealing the dark secrets of masked image modeling. In *CVPR*, 2023.

[64] Z. Yang, J. Zhao, B. Dhingra, K. He, W. W. Cohen, R. Salakhutdinov, and Y. LeCun. Glomo: Unsupervisedly learned relational graphs as transferable representations. *arXiv preprint arXiv:1806.05662v1*, 2018.

[65] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[66] S. Zagoruyko and N. Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[67] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[68] Y. Zhang, A. Backurs, S. Bubeck, R. Eldan, S. Gunasekar, and T. Wagner. Unveiling transformers with lego: a synthetic reasoning task. *arXiv preprint arXiv:2206.04301*, 2022.

Appendix

A Key Numbers

A.1 Information in Attention Transfer

How much information is transferred during attention transfer? Table 10 shows two ways of doing this accounting. If considering the map as 24 layers $\times$ 16 heads $\times$ 197 query tokens $\times$ 197 key tokens, there are about 15 million activations transferred per example. However, $QK^\top$ is low rank since Q and K are very "tall," so the attention map softmax$(QK^\top)$ can be considered 24 layers $\times$ 16 heads $\times$ 197 tokens $\times$ 64 head dim $\times$ 2 matrices, which is about 9.7 million activations.

Accounting Method	Parameters
Count sizes of Q, K	$24 \times 16 \times 197 \times 64 \times 2 \approx 9.7$M
Count size of $QK^\top$	$24 \times 16 \times 197 \times 197 \approx 15$M

Table 10: Number of parameters activations transferred per example for a ViT-L teacher.

A.2 Computational Cost of Attention Transfer

Attention transfer has the same computational and memory cost as any other knowledge distillation method that does a forward pass through a teacher. We compare fine-tuning vs attention distillation on a 16GB NVIDIA GP100 with ViT-L and a batch size of 16:

Method	Memory (GB)	Time per iteration (s)
Weight Fine-tuning	9.4	0.93
Knowledge Distillation	11.1	1.23
Attention Copy/Distillation	11.1	1.23

Table 11: Training cost of Attention Transfer.

Training these large models on ImageNet-1K is quite computationally expensive. 100 epochs of regular fine-tuning is about 2070 GPU-hours per 100 epochs, and 100 epochs of attention transfer is about 2735 GPU-hours. In total, we estimate about 150k GPU-hours are required to reproduce all experiments.

B Additional Analysis

B.1 Aggregated Attention Transfer

In Section 4.1, we conducted a thorough analysis of which aspects of attention matter the most. Another way to identify key properties of the teacher's attention maps is to average them across some axis during the transfer. For example, one can average the attention maps over all layers of the teacher network, so that the student uses the same map at every layer. Table 12 shows the results with aggregations over several natural axes. Averaging over examples (*i.e.*, using the same attention map, independent of the input) or averaging over query tokens (*i.e.*, each attention distribution is the same, regardless of the query token given an image) does quite poorly. This indicates, unsurprisingly, that these are key elements of self-attention. This also shows that prior work that focuses on aggregate statistics of the attention maps (*e.g.*, averaged over examples) [59] fail to capture the per-example nature of the attention maps that are actually responsible for full fine-tuning performance. Attention copy performance is more reasonable when averaging over heads or layers. This partially corroborates previous findings that attention maps can largely be shared across all layers [58]. However, while the results can be potentially improved with more recipe search, the performance is far short of the full fine-tuning accuracy (85.7).

average over	acc.
examples	79.7
layers	82.7
heads	82.2
query tokens	79.9
none	85.1

Table 12: **Aggregated attention transfer**. The full set of attention maps from the teacher has shape (examples, layers, heads, query tokens, key tokens). Here we try to average over each of these axes *before* the transfer. Performance drops the most when averaging over examples or query tokens, indicating that these are the most important aspects of the attention maps.

method	acc.
attn. distill	**85.7**
feature distill	81.3
fine-tune	**85.7**
scratch	83.0

Table 13: **Comparison with knowledge distillation**. We try knowledge distillation from the pre-trained teacher by adding an auxiliary MSE loss on the residual stream output. This encourages the student model to match the representations of the teacher ("feature distill"). We find that this does much worse than Attention Distillation.

B.2 Comparison to Knowledge Distillation

Our central hypothesis has been that the pre-trained attention maps are *sufficient*, and the pre-trained features are not *necessary*. Since our attention transfer methods are special instances of knowledge distillation [26], we additionally compare to a baseline of distilling the residual stream features from a pre-trained MAE ViT-L. In Table 13, we obtain a downstream accuracy of 81.3 on ImageNet-1k. This is significantly lower than the 85.7 that can be achieved through fine-tuning or attention distillation. This makes sense: the features learned during self-supervised pre-training are not directly well-suited for classification, so trying to match them can hurt performance. CKA analysis of the features (Figure 5) supports this hypothesis – the fine-tuned MAE does well by significantly changing the features in the latter half of the network. Overall, transferring attention appears to do much better than distilling the features.

B.3 Attention Map Analysis for Transferring Q

In Section 4.1, we found that copying the queries Q does surprisingly well, almost matching Attention Distillation or fine-tuning the pre-trained weights. Here, we compare the attention maps learned by the copy Q model to those of other models, in hopes of understanding why copy Q does so well.

Each of the 24 layers within a ViT-L has 16 attention heads, which each compute an $L \times L$ attention map for an image with L patches. We would like to determine the similarity between the attention heads in two models using some divergence measure; we use the Jensen-Shannon divergence (JSD) because it is symmetric in its arguments. However, there is one caveat. Because the output of the attention layer is invariant to the ordering of its heads, it is insufficient to compare the ith head of one model against the ith head of another. We need to properly match heads up across models. We explored four ways of doing so:

1. Direct pair: this is the naive approach of computing the JSD between the ith head of the first model and the ith head of the second model. This can fail since similar heads may not be in the same order across models.
2. Bipartite matching: for each layer, we compute the JSD between each of the 16 heads in the first model and the 16 heads in the second model. We then use bipartite matching to create a one-to-one pairing between the heads that minimizes the cumulative JSD. This solves the previous problem, but can still be thrown off, such as if one of the models has heads that it applies no weight to ($V = 0$ or $W_{proj} = 0$ or W_{proj} is orthogonal to the values).

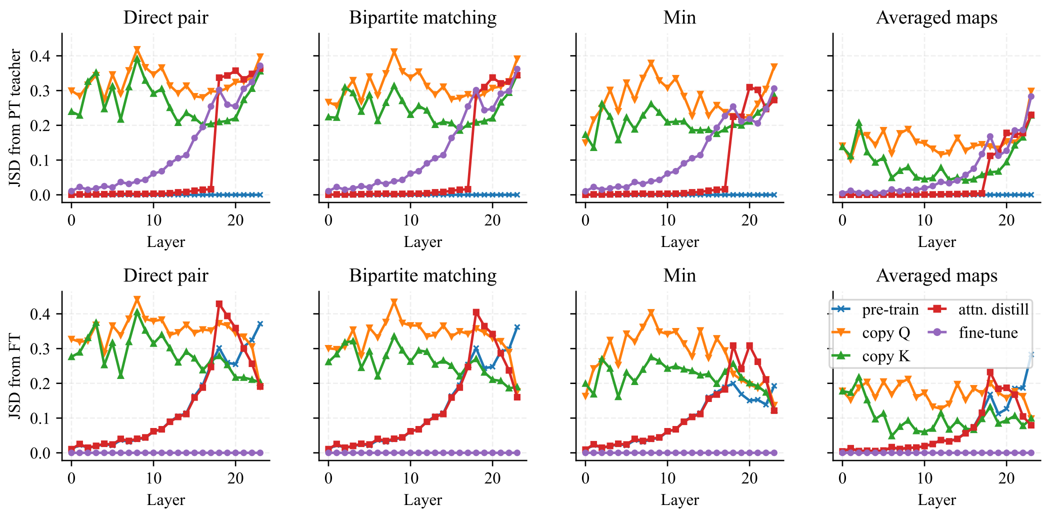

Figure 7: **Attention map similarity across methods**. Each column corresponds to a different way of matching up attention heads between two models. The top row shows the Jensen-Shannon divergence (JSD) with respect to the MAE pre-trained teacher, whereas the bottom row shows the JSD with respect to the fine-tuned MAE model. These plots match our intuition on distillation or fine-tuning methods, but copy Q is consistently dissimilar from the PT and FT models. Note that this may be a limitation of this particular analysis, since there are many settings of Q, K, and V that lead to the same layer output.

3. Minimum: instead of creating a one-to-one matching, we allow many-to-one matching between heads. We call this Minimum because each head in the first model is paired with the head from the second model with the smallest JSD. This allows our metric to potentially ignore extraneous heads in the second model, but is still susceptible to extraneous heads in the first model.

4. Averaged maps: we average the attention maps of all heads in a layer and compare the averaged maps across models. This can still be thrown off by extraneous heads.

Figure 7 shows the results of comparing models against the pre-trained teacher (top row) or fine-tuned model (bottom row) as the second model in the JSD. Most of our findings align with our intuition. In the top row, when comparing against the pre-trained teacher, attention distillation matches the teacher maps closely until layer 18, the last layer whose attention maps it is trained to approximate. The fine-tuned model's attention maps diverge more in later layers, since layerwise learning rate decay ensures that the earlier layers don't change much. However, copy Q is only somewhat similar to the pre-trained teacher or the fine-tuned model, across all of our ways to measure attention map similarity. Furthermore, it is less similar than copy K is, even though copy K has much lower downstream performance than copy Q.

Note that these plots have major limitations in what kinds of similarity they capture. With enough attention heads per layer, the same exact attention map can be partitioned differently across the heads between two models. Hypothetically, let's say that an attention layer wants to attend uniformly across all locations (*i.e.,* perform average pooling), and that we have 3 models, each with 2 attention heads:

1. Head 1 attends uniformly over all locations, head 2 attends arbitrarily over locations, and the second head's values are set to 0.

2. Head 1 attends uniformly over the top half of the image, head 2 attends uniformly over the bottom half of the image, and both use values $V/2$.

3. Head 1 attends uniformly over the left half of the image, head 2 attends uniformly over the right half of the image, and both use values $V/2$.

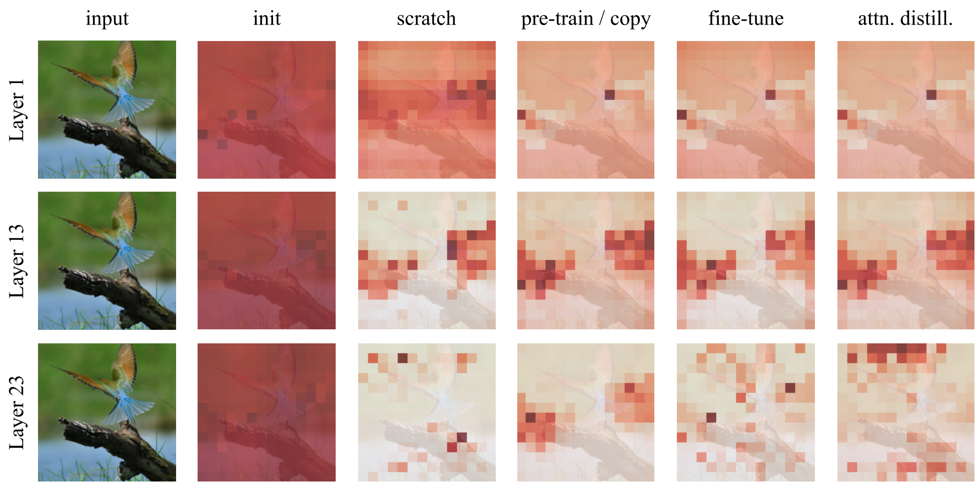

Figure 8: **Visualization of attention maps for different methods**. We show what the [CLS] token attends to at various layers within the network. Darker patches indicate more attention weight. Notably, the pre-trained MAE model's attention maps provide a significant prior over what the model should use, separating the object from potentially spurious cues like the branch or the background. In contrast, models right at random initialization ("init") start off attending uniformly over the image, which leads the scratch model to use more of the spurious patches. We show more attention map visualizations in Appendix D.

All 3 heads compute the same exact attention operation, yet would register as highly dissimilar in the setup from Figure 7. Overall, this experiment shows that copy Q's behavior is highly complex, and its strong downstream performance is still not fully understand.

B.4 Attention Map Visualizations

Section 4.2 provided several results to show that attention transfer does not simply "relearn" the teacher. Here, we examine one final piece of evidence. We show the attention maps of different networks in Figure 8. We focus on Attention Distillation, since Attention Copy's maps are identical to those in the teacher. Attention Distillation's maps generally match the teacher (pre-trained MAE), but are not completely identical for layers that are distilled (*e.g.,* layer 13). For layer 24, which is not distilled, Attention Distillation looks very different from pre-trained model, instead resembling the attention map of a model trained from scratch. These visualizations also highlight the fact that these attention maps are a very strong prior on what the model should use. While the randomly initialized model attends completely uniformly over all tokens, the pre-trained teacher attention maps already separate the relevant object from potentially spurious correlations (like the branch or background in the example). We show additional attention map visualizations in Figure 11 and Figure 12.

B.5 Attention Distillation Hyperparameter Sensitivity

We show the sensitivity of Attention Distillation to its hyperparameters.

Distillation loss weight We first consider the distillation loss weight λ which is used to compute the overall loss for the student:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda L_{\text{dist}} \tag{3}$$

Table 9 shows that a larger weight, $\lambda = 3$, does best. This may be because it encourages the student to learn useful attention maps more quickly, letting it guide feature learning earlier in training. We use this value of λ for our main result, where we match the 85.7 accuracy of fine-tuning. However, all other results in this paper use $\lambda = 1$ for simplicity.

Partial distillation: layers Just as we tried for Attention Copy, we also tried distilling various numbers of layers from the MAE teacher network, starting from the bottom of the network. Table 10 shows that there is a "sweet spot" when distilling the first 21 out of 24 layers. Distilling all layers may

Distillation loss weight λ	Accuracy
0.3	84.7
1.0	85.3
3.0	85.7

Figure 9: **Weight on distillation loss**

Layers distilled	Accuracy
18	85.3
21	85.5
24	85.1

Figure 10: **Number of layers distilled**

Teacher	Student	Accuracy
Random	Random	72.7
MAE	Random	85.1
MoCo-v3	Random	82.5
FLIP	Random	86.6
FLIP	MAE	84.2
MAE	MAE	85.4
MoCo-v3	MAE	82.9
MAE	FLIP	83.2

Table 14: **Decoupling attention maps from features**. We try Attention Distillation with various pre-trained students.

hurt performance by forcing the student to use attention maps that are more suited for reconstruction than classification. Note that all other distillation results in this paper use the first 18 layers by default.

B.6 Mix and Match, Student and Teacher

In the main paper, we focused on transferring attention maps from a pre-trained teacher to a randomly initialized student. However, the fact that Transformers have decoupled inter- and intra-token computation means that we can actually initialize the student with a pre-trained network as well. This entails testing whether the attention patterns from one network can improve the features of an already-pre-trained student model. We try Attention Distillation for various combinations of MAE, MoCo-v3, FLIP, and a randomly initialized network. Table 14 shows that this "mix-and-match" training does better than training from scratch (83.0) but does not match the performance in Table 7, where the students are randomly initialized. These are preliminary results, as the overall training recipe may need to be changed to accommodate the different learning dynamics of a different student model. Further hyperparameter tuning may significantly improve these results.

C Implementation Details

We present the training recipe for Attention Copy in Table 15 and the recipe for Attention Distillation in Table 16. For our partial layer transfer experiments in Figure 3, we set $\beta_2 = 0.95$ as it helps avoid training instabilities.

Config	Value
optimizer	AdamW [39]
base learning rate	1e-3
minimum absolute lr	2e-3
weight decay	0.05
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.999$
layerwise lr decay	0.75
batch size	2048
learning rate schedule	cosine decay [38]
warmup epochs [18]	5
training epochs	100
augmentation	RandAug (9, 0.5) [9]
label smoothing [51]	0.1
mixup [67]	0.8
cutmix [65]	1.0
drop path [29]	0
exp. moving average (EMA)	0.9999
layers copied	24

Table 15: **Training recipe for Attention Copy on ViT-L**.

Config	Value
optimizer	AdamW
base learning rate	1e-4
weight decay	0.3
optimizer momentum	$\beta_1 = 0.9, \beta_2 = 0.95$ [5]
batch size	2048
learning rate schedule	cosine decay
warmup epochs	20
training epochs	200
augmentation	RandAug (9, 0.5)
label smoothing	0.1
mixup	0.8
cutmix	1.0
drop path	0.2
exp. moving average (EMA)	0.9999
layers copied	18
distillation weight λ	3

Table 16: **Training recipe for Attention Distillation on ViT-L**.

D Additional Attention Map Visualizations

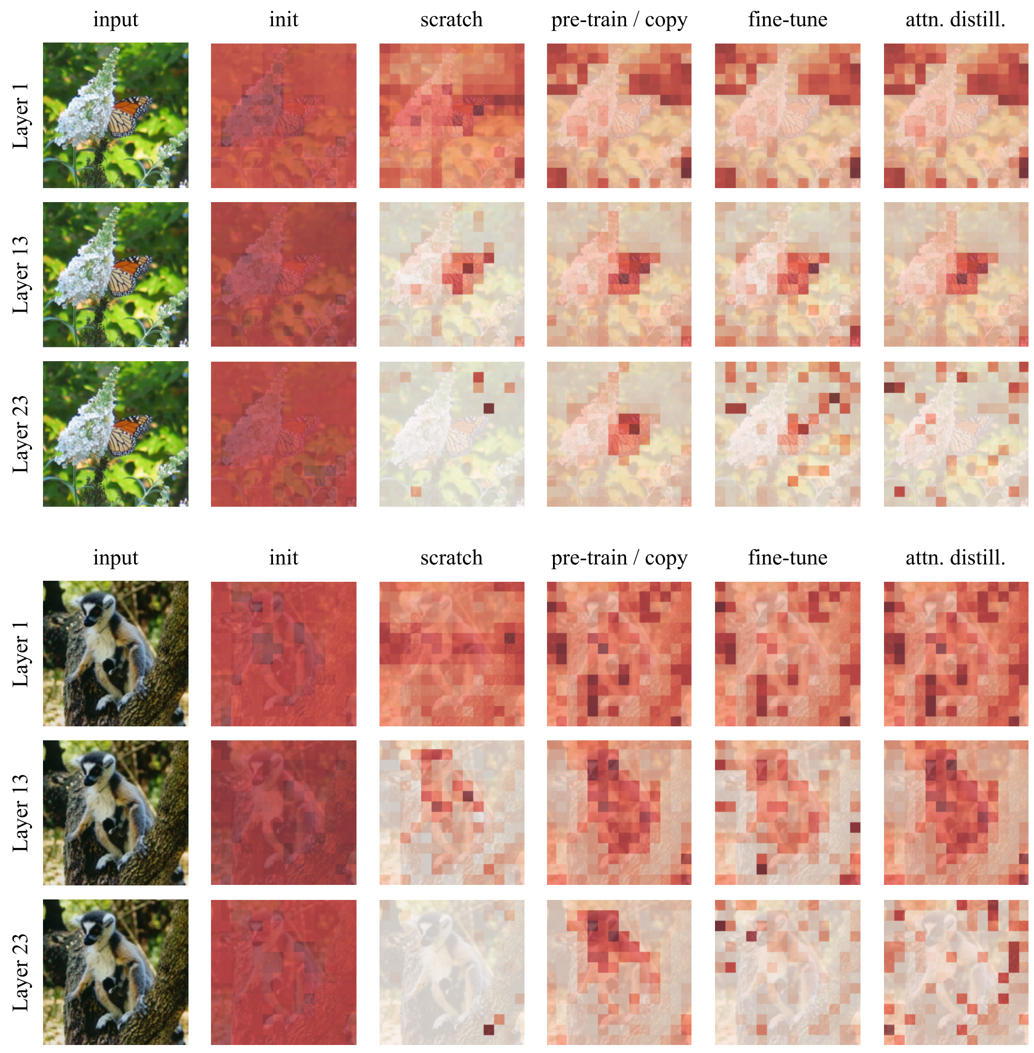

Figure 11: **Attention map visualizations on more examples**

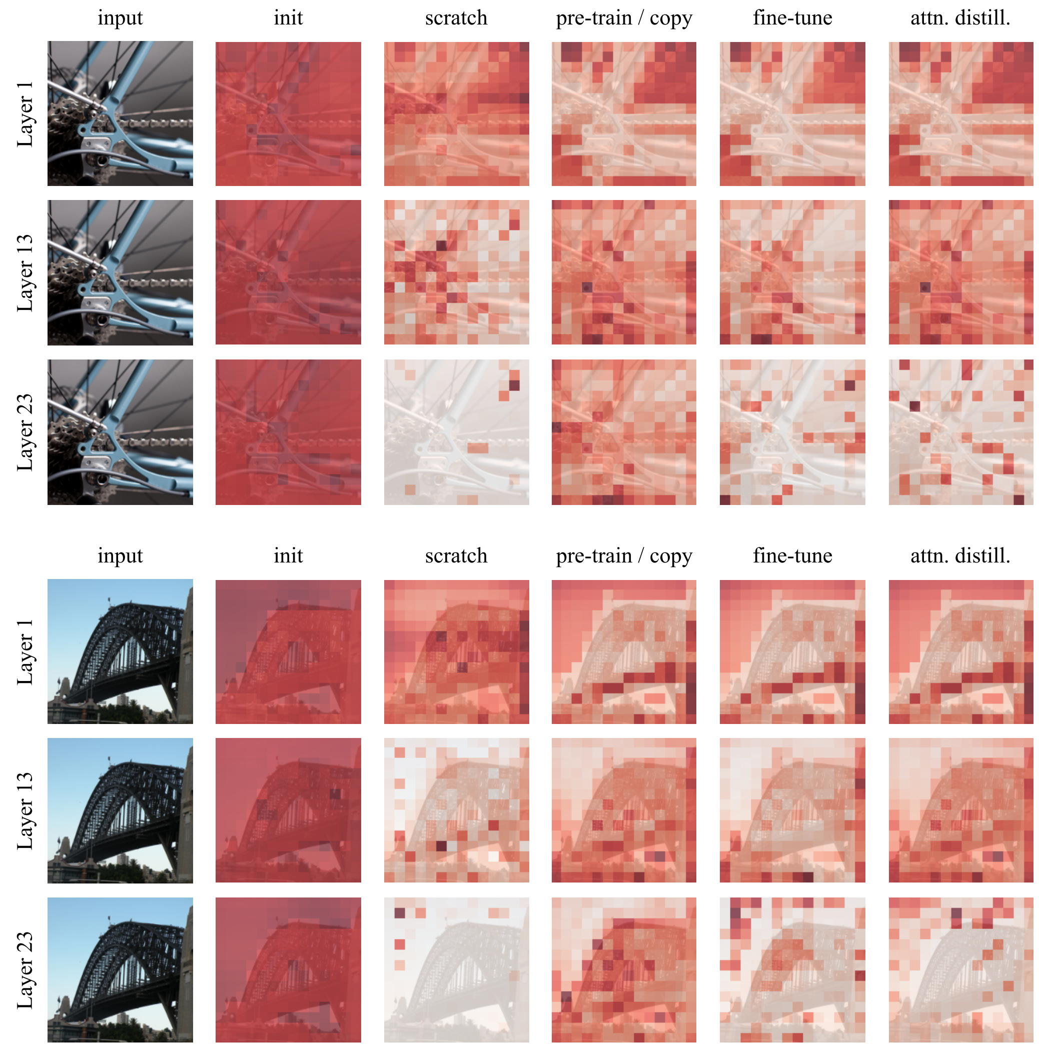

Figure 12: **Attention map visualizations on more examples**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We make clear in the abstract and introduction the limits to our claims.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 thoroughly discusses the settings where our analysis holds, and Appendix A.2 discusses the computational cost of attention transfer.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We do not provide any theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide thorough experimental details in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release our code at https://github.com/alexlioralexli/attention-transfer.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide thorough experimental details in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not typically reported in our settings (classification, detection) because it would be too computationally expensive. Furthermore, the relative difference we have observed is much bigger than the standard deviation (*e.g.*, < 0.2), so we follow common practice and do not report error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide this in Appendix A.2

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and believe that we conform with the guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The goal of our work is to advance the field of machine learn- ing. There are potential societal consequences of our work, but we believe none of them are significant and immediate.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper focuses on image classification and object detection, and we do not plan to release checkpoints or data.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all datasets and pre-trained checkpoints that we use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release any new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not use human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not use human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.