# Patching open-vocabulary models by interpolating weights

## Patching open-vocabulary models by interpolating weights

**Gabriel Ilharco***
University of Washington
gamaga@cs.washington.edu

**Mitchell Wortsman***
University of Washington
mitchnw@cs.washington.edu

**Samir Yitzhak Gadre***
Columbia University
sy@cs.columbia.edu

**Shuran Song**
Columbia Universiry
shurans@cs.columbia.edu

**Hannaneh Hajishirzi**
University of Washington
hannaneh@cs.washington.edu

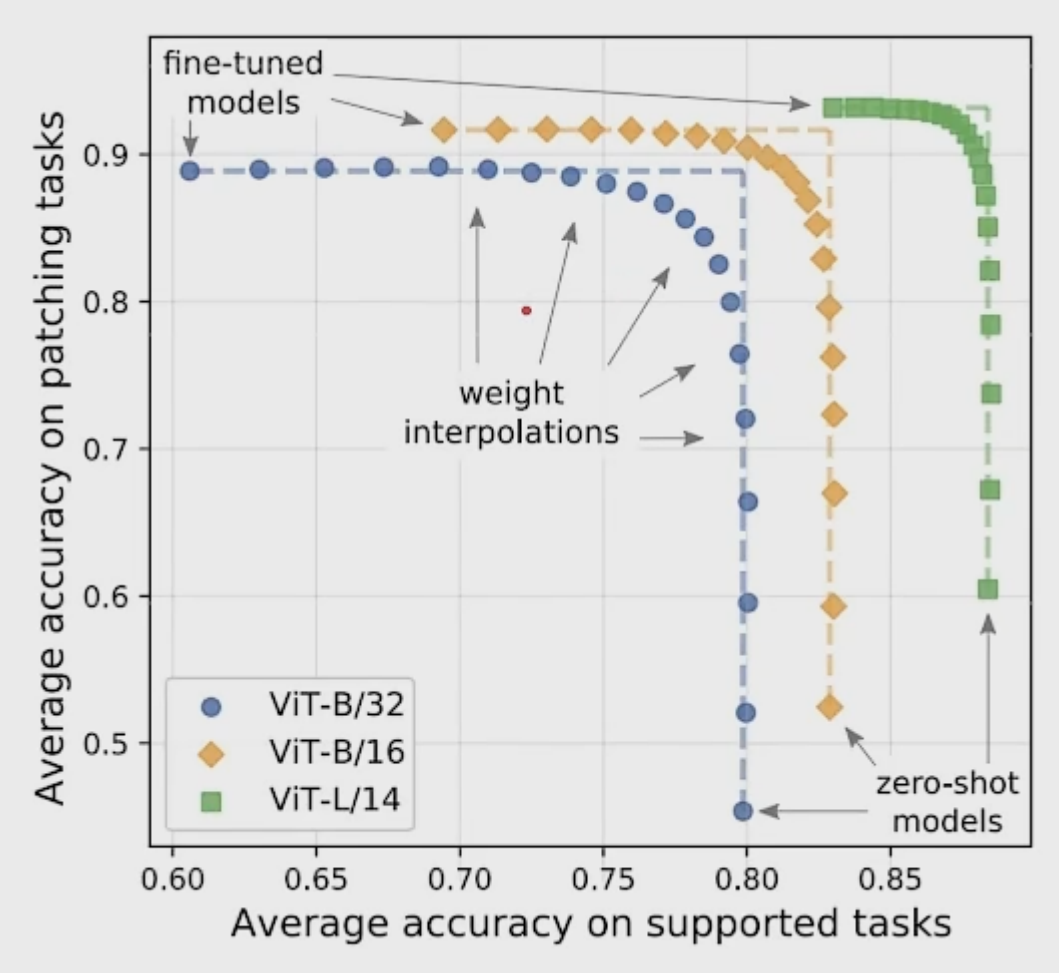
**Simon Kornblith**
Google Research, Brain Team
skornblith@google.com

**Ali Farhadi**
University of Washington
ali@cs.washington.edu

**Ludwig Schmidt**
University of Washington
schmidt@cs.washington.edu

Open-vocabulary models like CLIP achieve high accuracy across many image classification tasks. However, there are still settings where their zero-shot performance is far from optimal. We study model patching, where the goal is to improve accuracy on specific tasks without degrading accuracy on tasks where performance is already adequate. Towards this goal, we introduce PAINT, a patching method that uses interpolations between the weights of a model before fine-tuning and the weights after fine-tuning on a task to be patched. **On nine tasks** where zero-shot CLIP performs poorly, PAINT increases accuracy by 15 to 60 percentage points while preserving accuracy on ImageNet within one percentage point of the zero-shot model. PAINT also allows **a single model** to be patched on **multiple tasks** and improves with model scale. Furthermore, we identify cases of **broad transfer**, where patching on one task increases accuracy on other tasks even when the tasks have disjoint classes. Finally, we investigate applications **beyond common benchmarks** such as counting or reducing the impact of typographic attacks on CLIP. Our findings demonstrate that it is possible to expand the set of tasks on which open-vocabulary models achieve high accuracy without re-training them from scratch.

更好的利用预训练权重的方法，将一个fine-tune后的CLIP和原先的CLIP，做线性插值融合得到一个模型，这个模型在原本的任务上和fine-tune的特定任务上都能取到比较满意的效果。




一个模型在supported tasks上面效果很好，但是在一些patching tasks上面直接zero-shot时候效果就变的不太行了。这时候去做fint-tune，可以把在patching tasks上面的效果调高，但是在原本的supported tasks上效果会变差，这时候将两个模型（zero-shot model & fine-tune mmodel）做线性插值，融入混在一起，调整可以得到一条非常好的曲线，从而在两个任务上都取到比较好的效果。
可以在stable diffusion上面使用，就是训练出不同画风不同设置的model，然后相互融合去做改变。

此论文的任务定义：

## Supported Tasks

- Pretrained model performs well
- ImageNet, CIFAR10, CIFAR100, …

## Patching Tasks

- Linear probes outperform zerom-shto model by over 10%
- Cars, DTD, EuroSAT, KITTI, MNIST, ...

# Patching with Interpolation

## Patching on a single task

Linear interpolation

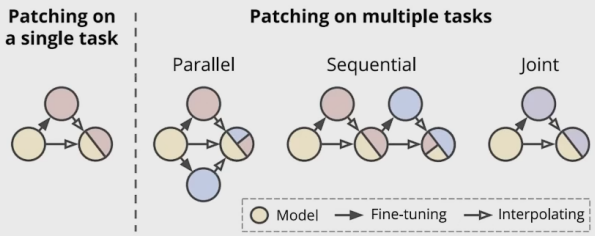



Figure 26: **An illustration of different patching strategies.** The area proportions on the circles that represent patched models are merely illustrative. In practice, the mixing coefficients are determined based on held-out validation sets. These diagrams are inspired by Matena and Raffel [47].

## Patching on a multiple tasks

1. Joint patching: merge all datasets before training
2. Sequential patching:
   iteratively repeat the patching on each new task
   (only use seen validation set)
3. Parallel patching: search for mixing coefficients for each task

# Useful Metrics

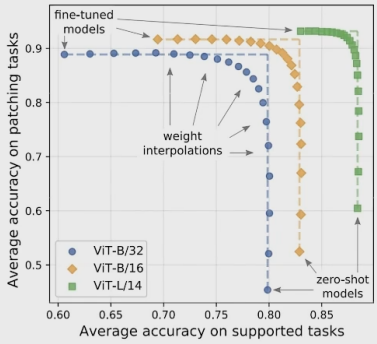
## Effectiveness of Patching

Accuracy distance to optimal

$$\frac{1}{2}\left[\max_{\alpha}\mathsf{Acc}(\theta_\alpha,\mathcal{D}_{\text{supp}})+\max_{\alpha}\mathsf{Acc}(\theta_\alpha,\mathcal{D}_{\text{patch}})\right]-\frac{1}{2}\max_{\alpha}\left[\mathsf{Acc}(\theta_\alpha,\mathcal{D}_{\text{supp}})+\mathsf{Acc}(\theta_\alpha,\mathcal{D}_{\text{patch}})\right]$$

## Model Similarity

Centered Kernel Alignment (CKA)
Cosine similarity




## Combined Accuracy

$$(\mathbb{E}_{\mathcal{D}_{\text{supp}}}[\mathsf{Acc}(\theta,\mathcal{D}_{\text{supp}})]+\mathbb{E}_{\mathcal{D}_{\text{patch}}}[\mathsf{Acc}(\theta,\mathcal{D}_{\text{patch}})])/2$$

# Obs on Single New Task

## Effect of scale

1. Easier to patch
2. Larger model => similar weights
3. Larger model => similar representation.

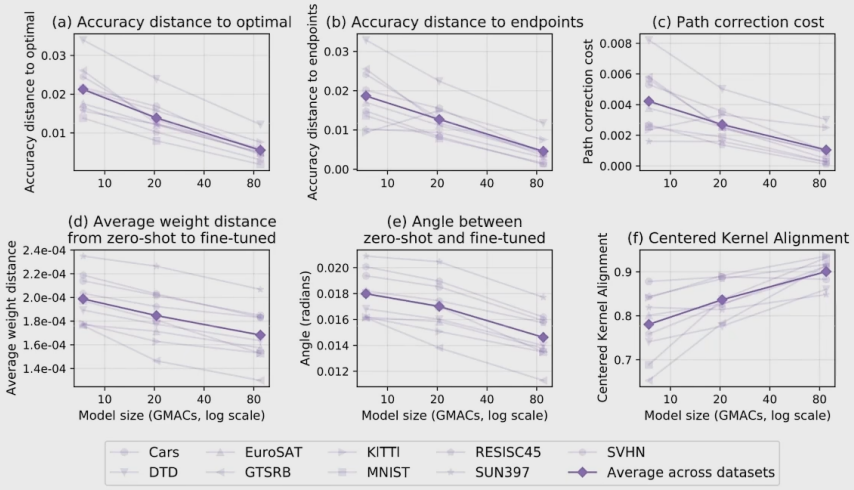



Figure 8: **The effect of scale on model patching**. (a-c) Patching is more effective for larger models. (d, e) Unpatched and fine-tuned models have more similar weights at scale. (f) For larger models, the unpatched and fine-tuned model are more similar with respect to their representations.

## **Elastic weight consolidation** penalizes the movement of important params.

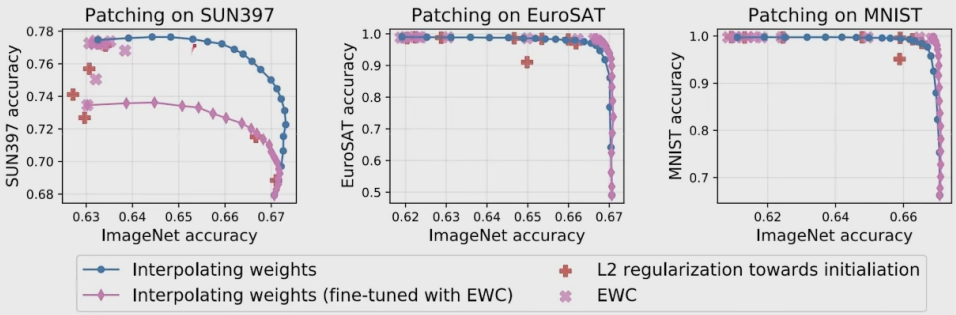



Figure 10: **Comparisons with EWC and regularization towards initialization.** When data is available from the pre-training set it is possible to augment standard fine-tuning with EWC [33]. EWC provides a solution with a good accuracy trade-off when patching on MNIST and EuroSAT, but not SUN397. We also show interpolations between the weights of the unpatched model and a model fine-tuned with EWC. As pre-training data is required to compute the fisher information matrix for EWC, these experiments use a ViT-B/16 model from a CLIP reproduction OpenCLIP [27] pre-trained on LAION 400M [67].

# Obs on Multiple Tasks

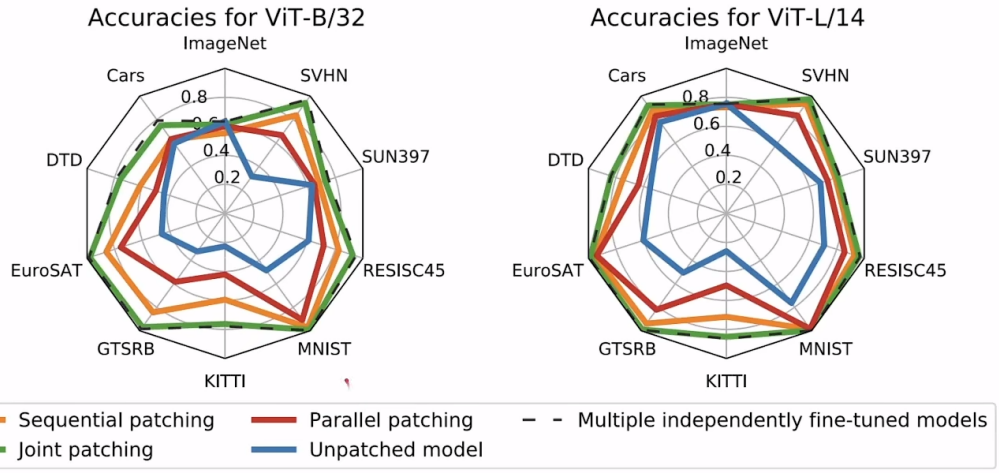



Figure 5: **Contrasting various strategies for patching on multiple tasks.** On all experiments, ImageNet is used as the supported task while the other nine datasets are used for patching. When data from all patching tasks is available, joint patching yields a single model that is competitive with using ten different specialized models. Weight interpolations greatly mitigate catastrophic forgetting on the sequential case, but do not completely eradicate it. Finally, parallel patching underperforms other patching strategies, but still provides improvements over the unpatched model.

# Broad Transfer

## Train on some cls, and test on other cls, similar to zero-shot setting

| | Cars | DTD | EuroSAT | GTSRB | KITTI | MNIST | RESISC45 | SUN397 | SVHN |
|---|---|---|---|---|---|---|---|---|---|
| Unpatched accuracy | 86.2 | 64.9 | 79.9 | 51.7 | 43.4 | 82.6 | 73.4 | 76.9 | 72.8 |
| Patched accuracy | 87.0 (+0.8) | 66.1 (+1.2) | 87.2 (+7.3) | 71.1 (+19.4) | 60.4 (+17.0) | 91.3 (+8.7) | 74.2 (+0.8) | 79.3 (+2.4) | 88.9 (+16.1) |

Table 1: **PAINT can generalize to unseen classes.** We randomly partition each dataset into tasks $A$ and $B$ with disjoint class spaces of roughly equal size. This table reports how patching on task $A$ affects accuracy on task $B$ for the ViT-L/14 model. In all cases, accuracy on task $B$ improves when patching on task $A$ even though the classes are *unseen* during patching.

| Task $A$<br>Task $B$ | MNIST<br>SVHN | SVHN<br>MNIST | EuroSAT<br>RESISC45 | RESISC45<br>EuroSAT | MNIST<br>FashionMNIST | FashionMNIST<br>MNIST | GTSRB<br>MTSD | MTSD<br>GTSRB |
|---|---|---|---|---|---|---|---|---|
| Unpatched accuracy | 58.6 | 76.4 | 71.0 | 60.2 | 67.7 | 76.4 | 19.3 | 50.6 |
| Patched accuracy | 68.9 (+10.3) | 93.2 (+16.8) | 69.7 (-1.3) | 70.4 (+10.2) | 70.8 (+3.1) | 77.5 (+1.1) | 30.8 (+11.5) | 69.8 (+19.2) |

Table 2: **Patching on task $A$ can improve accuracy on a related task $B$.** For a pair of tasks $A$ and $B$, we report accuracy of the ViT-L/14 on task $B$, after patching on task $A$, finding improvements on seven out of eight cases.