Diminishing Returns in Self-Supervised Learning: More Is Not Always That Much Better

Huey Sun

University College London, London UK ucabhss@ucl.ac.uk
https://www.ucl.ac.uk/computer-science/

Abstract. While transformer-based architectures have taken computer vision and NLP by storm, they often require a vast amount of parameters and training data to attain strong performance. In this work, we experiment with three distinct pre-training, intermediate fine-tuning, and downstream datasets and training objectives to explore their marginal benefits on a small 5M-parameter vision transformer. We find that while pre-training and fine-tuning always help our model but have diminishing returns, intermediate fine-tuning can actually show harmful impact on downstream performance, potentially due to dissimilarity in task mechanics.

1 Introduction

1.1 Background

Innovations in deep learning have led to tremendous advances in machine vision from image classification and object detection to semantic segmentation. However, the sparsity of quality labelled data is often a barrier to practical application and usage of deep learning models. One solution, called self-supervised learning (SSL), "pre-trains" a model in an unsupervised manner on a vast corpus of unlabelled data before fine-tuning it for a downstream task. This approach leverages larger and potentially more accessible data to learn useful representations that then transfer to the supervised task at hand.

With ample pre-training, Vision Transformers (ViTs) have emerged as a highly-scalable alternative to traditional convolutional neural networks, and exhibit state-of-the-art performance at their largest size [8]. By splitting input images into patches and treating each patch analogously to a token, ViTs leverage the self-attention transformer architecture for vision, unlocking cross-over adaptation between novel research in Natural Language Processing (NLP). However, configuring ViT's with a massive amount of parameters requires immense and often prohibitively expensive compute to train and perform inference. As such, understanding the baseline performance of ViTs at small scale is crucial to the democratization of computer vision and widespread application of new algorithms. In this research, we will use a ViT that's tiny (ViNy) to investigate how the quantity and quality of pre-training and fine-tuning affects downstream tasks such as semantic segmentation.

To do so, we use masked image modeling (MIM) as our pre-training algorithm. With MIM, portions of input images are hidden ("masked") and a model is trained to predict the missing pixels [5]. MIM allows models to develop representations of the image that encode semantic information such as edges or shapes which can be transferred to downstream tasks. MIM has been shown to produce significant increases in performance in massive ViTs [13], making it an interesting subject of study at small scale.

Beyond unsupervised pre-training, intermediate fine-tuning can also be applied before the downstream task to further improve model performance. In this research, we evaluate the performance of ViNy on a downstream supervised segmentation task using the Oxford-IIIT Pet Dataset [9] with and without SimMIM pre-training on ImageNet-1K [12]. Across the runs, we explore the interactions between the varying amounts of pre-training and fine-tuning data used to train our model. Furthermore, we evaluate whether adding an additional intermediate classification fine-tuning routine with the Intel Image Classification dataset [1] helps our model's performance. Our findings suggest that while more training data results in better performance, the relationship has diminishing returns, and that inappropriate intermediate fine-tuning can actually worsen your model.

1.2 Related Literature

In NLP, self-supervised pre-training has most prominently been used in the Generative Pre-trained Transformer (GPT) [11] series of models, while masked modeling in particular has gained popularity as a result of the success of deep bidirectional transformers (BERT) [6]. Supervised intermediate fine-tuning, on the other hand, was first introduced by Phang et al. [10] as Supplementary Training on Intermediate Labeled-data Tasks (STILTS), and has been shown to increase performance in pre-trained models across a variety of NLP benchmarks. These intermediate tasks are chosen to be generally useful or data-rich, often requiring complex reasoning, and are thought to improve the model's ability to transfer to any target downstream task. However, recent research has shown that the benefits of intermediate fine-tuning may be less linked to reasoning ability as previously thought. To show this, Chang and Lu [4] successively modify intermediate fine-tuning datasets to remove the reasoning component while preserving their utility as a supplementary task, painting a highly nuanced and intricate picture on the nature of intermediate training. Furthermore, they show that the success of intermediate fine-tuning can be surprisingly dependant on the dataset size of the task.

Pre-training and intermediate training have also been explored in computer vision. When Bao et al. [2] introduced their adaptation of BERT to images, they used intermediate fine-tuning on top of their pre-training to improve performance on downstream classification and semantic segmentation tasks. Furthermore, El-Nouby et al. [7] show that pre-training data does not need to be massive in order to be useful. In this work, we build off these insights to study the interactions and tradeoffs between pre-training, finetuning, and intermediate training on a small scale.

2 Methodology

2.1 ViNy

All of our experiments use ViNy, a tiny ViT configured with an input image size of 128, a patch size of 16, a dimension of 128, a depth of 12, 8 heads, and a multi-layer perceptron dimension of 512. These values were chosen to reduce the amount of parameters from ViT-Base, which has 86 million parameters, to 4,843,138, representing a meaningful decrease in size.

2.2 Pre-training with SimMIM

To perform unsupervised pre-training with ViNy, we use SimMIM [13], a simple framework that adapts masked modeling to images. SimMIM applies masking over patches of the input image and treats the masked prediction as a regression task with an L1 loss. To promote rich semantic representation learning from the transformer body, SimMIM keeps a lightweight linear layer as its prediction head. In our work, we use SimMIM with the recommended masking ratio of 0.5 and pre-train our model on ImageNet-1K. While ImageNet is the standard for pre-training given its curated quality and breadth [12], ImageNet-1K has been shown to be effective at smaller scale [3], making it a natural choice for our work. In our experiments, we vary the pre-training between 0 (no pre-training), 45k, 100k, and 200k examples with training parameters specified in Appendix A1.

2.3 Intermediate Fine-tuning

While classification with ImageNet-1K has been shown to be a beneficial intermediate fine-tuning task paired with ImageNet-1K pre-training [2], it is unclear if this gain in performance stems from the richness of the data or from the nature of the task itself. To investigate whether the act of applying the semantic knowledge gained from pre-training on an unrelated classification task can improve downstream segmentation, we perform intermediate fine-tuning on the Intel Classification Data [1], a lightweight data set that contains 14 thousand training examples across 6 classes of natural scenes such as glaciers, buildings, and mountains. To predict the 6 classes, we fit ViNy with a 128x6 multi-layer perceptron head, which is then replaced with the segmentation head for downstream fine-tuning. While we would likely do better on our downstream task if we picked a more similar intermediate dataset, our main goal with this work is to explore the dynamics of training rather than maximize model performance.

2.4 Downstream Segmentation Fine-tuning

Our downstream supervised task is to segment the Oxford-IIIT Pet Dataset [9] into tri-grams of foreground, background, and unknown pixels. Across the experiments, we vary the amount of fine-tuning data between 250, 500, 1000, 2000, 4000, and 6000 examples, and evaluate our model on 1000 held-out test

examples randomly split with a generator seeded at 42. ViNy is equipped with a 128x768 multi-layer perceptron head to predict the segmentation, as our patch size is 16 and our target is a tri-map ($16 * 16 * 3 = 768$). Along with accuracy, we also report the mean intersection over union (mIoU) of our predictions, a stricter metric that averages the ratio between the correctly predicted pixels over all predicted pixels for each of the three classes. A visualization of tri-map segmentation can be found in Figure A1 in the appendix.

3 Experiments

First, we conduct an experiment to determine how the baseline ViNy model performs as we increase the number of fine-tuning examples, evaluating accuracy and mIoU on 1000 held-out test examples (Table 1).

Table 1. Baseline ViNy performance over 3 runs by number of fine-tuning examples

# **Finetune**	**Accuracy**	**mIoU**
250	70.33 ± 0.89%	43.73 ± 1.02%
500	73.71 ± 0.08%	47.41 ± 0.21%
1000	77.81 ± 0.56%	53.28 ± 0.43%
2000	80.03 ± 0.42%	56.87 ± 0.51%
4000	84.16 ± 0.06%	62.99 ± 0.16%
6000	85.60 ± 0.03%	65.07 ± 0.01%

We then repeat this experiment for ViNy models pre-trained using SimMIM with a varying number of training examples from ImageNet 1-K. The results are visualized in Figure 1 and listed in Table A2 in the appendix.

Finally, we repeated this experiment for each pre-trained model, adding intermediate fine-tuning before downstream fine-tuning. The impact of the additional intermediate fine-tuning is visualized in Figure 2 and the data is listed in Table A3 in the appendix. In every pre-trained model and at every downstream fine-tuning size, the additional of intermediate fine-tuning worsens the mIoU of our model on the downstream segmentation task.

4 Results

Across our data, we see a clear trend: increasing the amount of pre-training and fine-tuning data improves test accuracy and mIoU on the downstream segmentation task. The benefits of pre-training are particularly pronounced with low fine-tuning size: with 250 fine-tuning examples, the baseline ViNy achieves a mIoU of 43.73% while the 200k pre-trained model gets 57.14%, an increase of more than 13%. Even the smaller 45k pre-trained model gets a significant boost of 9%. With 6000 fine-tuning examples, this gap closes, and the 45k and 200K

Fig. 1. Comparing ViNy mIoU% across pre-training and fine-tuning data size (3 runs)

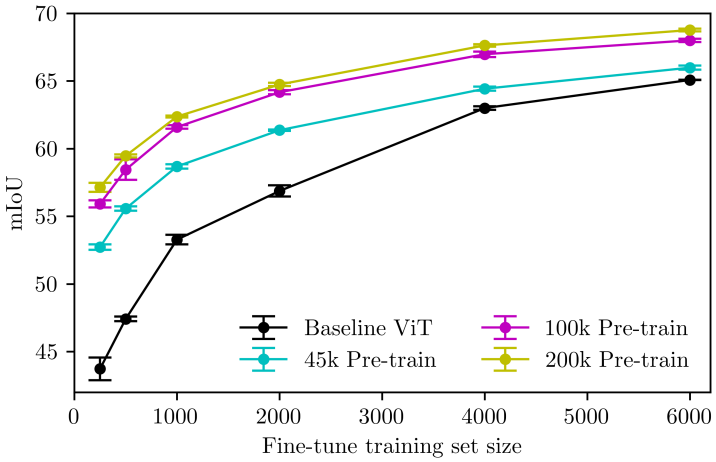
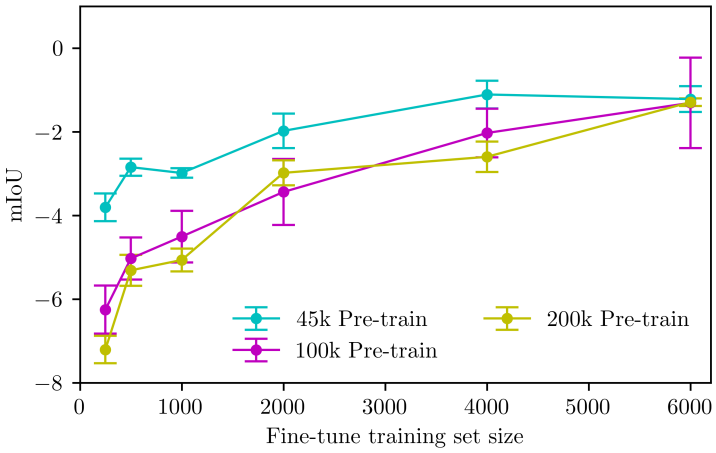


Fig. 2. Plotting the mean and standard error of the difference in mIoU% with and without intermediate fine-tuning across 3 runs.



pre-trained models get an increase of 0.92% and 3.69%, respectively. These results align with our expectations, as pre-training is meant to give our models a strong starting point, which is naturally less important as we fine-tune on more data.

We can also observe that both pre-training and fine-tuning have diminishing returns: for pre-training, the gap between the baseline and 100k-PT models is 8.31% and 3.98% for 1000 and 2000 fine-tuning examples, while the gap between the 100k and 200k pre-trained models for the same fine-tuning sizes are 0.78%

and 0.65%. While fine-tuning also plateaus in all the models (see Figure 1), the pre-trained models plateau at higher mIoU than the baseline, indicating that pre-training gives the model an advantage that fine-tuning can't make up.

Intermediate fine-tuning, on the other hand, causes our models to worsen with every type of pre-training and fine-tuning. While increasing the amount of fine-tuning data lessens the negative impact of intermediate fine-tuning, its inclusion is still decidedly negative for all models (see Figure 2). Furthermore, this effect is also more pronounced on models that receive bigger benefits from pre-training, indicating that it may be negating the gains from pre-training.

5 Discussion

It's not surprising that using more pre-training and fine-tuning data results in better models, but it is interesting that the model exhibits diminishing returns at a peak of 68.76% mIoU. One explanation could be that the model is simply too small to take advantage of further pre-training. Future work could explore the relationship between model size and pre-training efficacy. Another explanation could be that our pre-training at 100 epochs is not long enough to extract all the signal from the data. One limitation of our work is that we did not optimize the parameters for each of our training runs due to time constraints, so our training may have gotten more out of the smaller data sets than larger ones.

It is also important to understand why intermediate fine-tuning causes performance degradation in our downstream task. While we naively suppose that any type of training improves latent representations that are easily transferable, research into intermediate fine-tuning in NLP has shown that successful intermediate datasets do not always need to contain useful information [4]. To explain this, Chang and Lu theorize that the mechanism of the intermediate task has great influence on its utility. In our case, the classification-based training may be pushing the model to contain and aggregate semantic information within the [CLS] token that is attached to the input, making it orthogonal (or even opposed) to the aims of semantic segmentation, which is far more local. While classification-based intermediate training has been previously shown to aid semantic segmentation [2] when applied to the data also used for pre-training, it would be interesting to further investigate what was learned in that setting and how much the intermediate task can be modified while maintaining downstream performance.

6 Conclusion

In this work, we evaluate the impact of pre-training, intermediate fine-tuning, and downstream fine-tuning on semantic segmentation performance across different data sizes. The conclusions are unambiguous: use as much pre-training and fine-tuning data as you can, prioritizing the latter in a low-compute regime, and be very judicious with intermediate fine-tuning, as an unrelated task can degrade your model's performance.

References

1. Intel image classification dataset, https://www.kaggle.com/puneet6060/intel-image-classification
2. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=p-BhZSz59o4
3. Beyer, L., Zhai, X., Kolesnikov, A.: Better plain vit baselines for imagenet-1k (2022)
4. Chang, T.Y., Lu, C.J.: Rethinking why intermediate-task fine-tuning works. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 706–713. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.findings-emnlp.61, https://aclanthology.org/2021.findings-emnlp.61
5. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 1691–1703. PMLR (13–18 Jul 2020), https://proceedings.mlr.press/v119/chen20s.html
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
7. El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jégou, H., Grave, E.: Are large-scale datasets necessary for self-supervised pre-training? ArXiv **abs/2112.10740** (2021), https://api.semanticscholar.org/CorpusID:245334705
8. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15979–15988 (2022). https://doi.org/10.1109/CVPR52688.2022.01553
9. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3498–3505 (2012). https://doi.org/10.1109/CVPR.2012.6248092
10. Phang, J., Févry, T., Bowman, S.R.: Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. CoRR **abs/1811.01088** (2018), http://arxiv.org/abs/1811.01088
11. Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018), https://api.semanticscholar.org/CorpusID:49313245
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., Fei-Fei, L.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115** (09 2014). https://doi.org/10.1007/s11263-015-0816-y
13. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: a simple framework for masked image modeling. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9643–9653 (2022). https://doi.org/10.1109/CVPR52688.2022.00943

7 Appendix

Table A1. Training Configurations for the Optimizer and Learning Rate Scheduler

		AdamW Params		MultiStepLR Params	
Phase	**Epochs**	**Learning Rate**	**Weight Decay**	**Gamma**	**Milestones**
Pre-training	100	1×10^{-4}	0.05	0.1	{50, 85}
Intermediate	200	8×10^{-4}	0	0.1	{180, 190}
Fine-tuning	100	2×10^{-3}	1×10^{-4}	0.5	{70, 90, 95}

Fig. A1. A semantic segmentation example on the Oxford III Pets data set. The left image is the input, the center image is the ground truth, and the right image is our model's prediction

Table A2. Pre-trained ViNy test accuracy and mIoU across 3 runs by number of pre-training examples and number of fine-tuning examples

# Pre-train	# Fine-tune	Accuracy	mIoU
45k	250	77.44 ± 0.06%	52.73 ± 0.25%
	500	79.67 ± 0.20%	55.57 ± 0.19%
	1000	81.92 ± 0.20%	58.68 ± 0.20%
	2000	83.38 ± 0.03%	61.36 ± 0.05%
	4000	85.41 ± 0.13%	64.42 ± 0.19%
	6000	86.40 ± 0.11%	65.98 ± 0.19%
100k	250	79.67 ± 0.35%	55.91 ± 0.32%
	500	81.62 ± 0.62%	58.44 ± 0.93%
	1000	83.60 ± 0.14%	61.59 ± 0.16%
	2000	84.99 ± 0.18%	64.17 ± 0.18%
	4000	86.81 ± 0.12%	66.97 ± 0.25%
	6000	87.53 ± 0.08%	68.00 ± 0.14%
200k	250	80.49 ± 0.18%	57.14 ± 0.42%
	500	82.31 ± 0.04%	59.47 ± 0.13%
	1000	84.17 ± 0.15%	62.37 ± 0.08%
	2000	85.38 ± 0.11%	64.74 ± 0.14%
	4000	87.23 ± 0.06%	67.62 ± 0.12%
	6000	87.93 ± 0.04%	68.76 ± 0.13%

Table A3. Pre-trained and intermediate fine-tuned ViNy test accuracy and mIoU across 3 runs by number of pre-training examples and number of fine-tuning examples

# Pre-train	# Fine-tune	Accuracy	mIoU
45k	250	74.65 ± 0.97%	48.92 ± 0.66%
	500	77.86 ± 0.32%	52.72 ± 0.39%
	1000	79.80 ± 0.11%	55.70 ± 0.13%
	2000	82.08 ± 0.57%	59.38 ± 0.88%
	4000	84.86 ± 0.28%	63.31 ± 0.68%
	6000	85.70 ± 0.42%	64.76 ± 0.62%
100k	250	75.72 ± 0.49%	50.20 ± 0.29%
	500	78.59 ± 0.25%	53.85 ± 0.44%
	1000	81.13 ± 0.19%	57.80 ± 0.10%
	2000	83.77 ± 0.26%	61.62 ± 0.11%
	4000	86.13 ± 0.11%	65.29 ± 0.61%
	6000	87.32 ± 0.48%	67.79 ± 0.43%
200k	250	75.83 ± 0.66%	49.93 ± 0.56%
	500	78.49 ± 0.69%	54.15 ± 0.77%
	1000	80.96 ± 0.26%	57.30 ± 0.57%
	2000	83.96 ± 0.40%	61.75 ± 0.62%
	4000	85.88 ± 0.48%	65.02 ± 0.77%
	6000	87.18 ± 0.14%	67.47 ± 0.15%