Task Addition and Weight Disentanglement in Closed-Vocabulary Models

Adam Hazimeh [*1] **Alessandro Favero** [*1 2] **Pascal Frossard** [1]

Abstract

Task arithmetic has recently emerged as a promising method for editing pre-trained *open-vocabulary* models, offering a cost-effective alternative to standard multi-task fine-tuning. However, despite the abundance of *closed-vocabulary* models that are not pre-trained with language supervision, applying task arithmetic to these models remains unexplored. In this paper, we deploy and study task addition in closed-vocabulary image classification models. We consider different pre-training schemes and find that *weight disentanglement* – the property enabling task arithmetic – is a general consequence of pre-training, as it appears in different pre-trained closed-vocabulary models. In fact, we find that pre-trained closed-vocabulary vision transformers can also be edited with task arithmetic, achieving high task addition performance and enabling the efficient deployment of multi-task models. Finally, we demonstrate that simple linear probing is a competitive baseline to task addition. Overall, our findings expand the applicability of task arithmetic to a broader class of pre-trained models and open the way for more efficient use of pre-trained models in diverse settings.

1. Introduction

Pre-trained models are widely used as backbones in modern machine learning systems. However, to enhance their performance on downstream tasks (Zhuang et al., 2020; Sanh et al., 2021; Ilharco et al., 2022a) and increase their robustness (Santurkar et al., 2021; Ortiz-Jiménez et al., 2021; Wortsman et al., 2022b), these models often require further *editing*. The most common editing method is *fine-tuning*, where pre-trained models are re-trained on specific target tasks. However, aligning models with multiple downstream tasks simultaneously requires joint fine-tuning, which is computationally expensive.

Recently, more cost-effective, scalable, and modular techniques have been introduced, such as editing models directly in weight space via weight interpolation (Wortsman et al., 2022b; Izmailov et al., 2018; Ainsworth et al., 2022; Wortsman et al., 2022a; Yadav et al., 2024) or task arithmetic (Ilharco et al., 2022a; Ortiz-Jimenez et al., 2024; Wang et al., 2024). In particular, *task arithmetic* combines multiple, independently fine-tuned model weights through arithmetic operations, thus avoiding the costs of joint fine-tuning. This approach has shown significant potential in preserving both pre-training and fine-tuning performance. However, task arithmetic has thus far only been applied to *open-vocabulary* models, such as CLIP (Radford et al., 2021), which undergo large-scale contrastive pre-training on billions of image-caption pairs and are not limited to a predefined set of classes.

In contrast, in computer vision, a major portion of pre-trained models, which we will henceforth refer to as *closed-vocabulary* or *closed*, is not pre-trained with (weak) language supervision but through standard supervised or self-supervised strategies on typically smaller data scales. Importantly, such models lack the flexibility of their open-vocabulary counterparts and require task-specific heads[2] depending on the details of the target tasks. Understanding whether such models can be edited with task arithmetic remains an open question. In particular, Ortiz-Jimenez et al. (2024) showed that task arithmetic is enabled by *weight disentanglement* and that such a property emerges with contrastive vision-language pre-training. Yet, as different models can learn varying internal representations, it is unclear if weight disentanglement also emerges with closed-vocabulary pre-training.

In this paper, we study the scope of task arithmetic to determine whether its success can be leveraged for models pre-trained with diverse pre-training schemes and data scales. Our main contributions are as follows:

1. We deploy and study task addition in closed-vocabulary models. In particular, before fine-tuning

[*]Equal contribution [1]LTS4, EPFL, Lausanne, Switzerland [2]PCSL, EPFL, Lausanne, Switzerland. Correspondence to: Adam Hazimeh <adam.hazimeh@epfl.ch>.

Efficient Systems for Foundation Models (ES-FoMo) Workshop, 41st International Conference on Machine Learning, Vienna, Austria. 2024.

[2]For more details on how open-vocabulary models deal with multiple tasks, see Appendix B.

the encoder, we introduce a task-specific classification head, which we align with the pre-trained encoder via *linear probing*.

2. We consider different common pre-training schemes and observe that weight disentanglement is not exclusive to vision-language contrastive pre-training but instead is a general consequence of pre-training.
3. For the same pre-training schemes, we study the performance of task addition with vision transformers of different sizes on 8 image classification tasks, showing that closed-vocabulary models achieve high task addition performance.
4. Finally, we show that linear probing alone achieves competitive performance to task addition, making it a cheap alternative to task addition for practitioners.

2. Background

In this section, we present the relevant background on task arithmetic and weight disentanglement. We refer the reader to Appendix A for additional context.

Task arithmetic As introduced by Ilharco et al. (2022a) and further formalized by Ortiz-Jimenez et al. (2024), task arithmetic operates on fine-tuned models by isolating their fine-tuned weights from the pre-trained initialization and performing simple arithmetic operations on the resulting weight differences, known as *task vectors*. Formally, a task t is defined by a dataset and an associated loss function. The corresponding task vector τ_t is the element-wise difference between the network's pre-trained weights θ_{pre} and its fine-tuned weights θ_{ft}^t, i.e., $\tau_t = \theta_{\text{ft}}^t - \theta_{\text{pre}}$. Task arithmetic is performed by applying arithmetic operations between different task vectors. For instance, *task addition* adds scaled task vectors to the pre-training weights to produce a multi-task model that is aligned with the target tasks, i.e., $\theta_{\text{new}} = \theta_{\text{pre}} + \sum_t \lambda_t \tau_t$.

Weight disentanglement Such a simple editing technique has shown great performance when applied to open-vocabulary models like CLIP, retaining a significant portion of the accuracy of the single-task fine-tuned models. However, given the non-linear nature of neural networks, why does it work? Ortiz-Jimenez et al. (2024) attribute the success of task arithmetic to *weight disentanglement*. This property, seemingly emerging during pre-training, is the ability of a network to decompose its weight space into distinct linear subspaces associated with high performance on different tasks. Weight disentanglement has been observed in Ortiz-Jimenez et al. (2024) in different CLIP architectures based on Vision Transformers (Dosovitskiy et al., 2020) and ConvNexts (Liu et al., 2022). However, it is still unclear if this property can be generalized to *all* pre-training schemes, including those that do not rely on natural language supervision. Furthermore, the influence of different pre-training factors on the emergence of weight disentanglement, such as data characteristics and the training algorithm, is yet to be understood.

To address these gaps, we study task arithmetic, specifically task addition, in the closed-vocabulary setting.

3. Task Arithmetic with Closed Models

3.1. Handling the Classification Head

The main challenge in extending task arithmetic to the closed-vocabulary setting is the need for task-specific heads. For open-vocabulary vision models, we can use the model's pre-trained language encoder and leverage the universality of natural language to describe different tasks and classes (see Appendix B for more details). However, in the closed setting, this is clearly impossible.

Following Kumar et al. (2021), for each task, we propose to first fine-tune a randomly initialized head while freezing the rest of the network – referred to as *linear probing* – and then fine-tune the encoder while freezing the head. The first alignment phase is motivated by CLIP fine-tuning, which occurs while keeping the weights of the well-initialized language encoder frozen. In fact, in the closed setting, the first probing phase aligns the head and encoder weights to achieve a better head initialization before fine-tuning the vision encoder. In Appendix G, we show that full fine-tuning, i.e., fine-tuning the encoder and a randomly initialized head simultaneously, leads to significantly worse results. Note that, while merging, we apply the task vector to the pre-trained encoder weights and then plug in the task-specific classification head.

3.2. Experimental Setup

Task arithmetic We focus on task addition over image classification tasks. We consider a uniform set of scaling coefficients for our task vectors, i.e., $\forall t,\ \lambda_t = \lambda$, which is determined via a line search over 21 equispaced values of $\lambda \in \{0, 0.05, 0.1, ..., 0.95, 1\}$, similarly to Ortiz-Jimenez et al. (2024). The best coefficient is chosen to be the one that maximizes the normalized average accuracy of the resultant model across all tasks, where the normalization is performed with respect to the single-task accuracies of each independently fine-tuned model.

Datasets We select the same 8 image classification tasks evaluated in Ilharco et al. (2022a): Stanford Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2012), MNIST (LeCun & Cortes, 2010), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2010), and SVHN (Netzer et al., 2011).

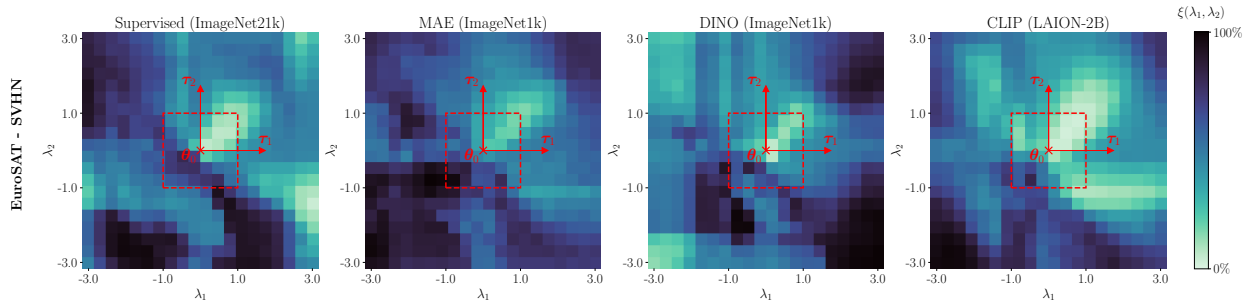

Figure 1. **Weight disentanglement error heatmaps for the different pre-training algorithms.** The heatmaps show the pairwise weight disentanglement error $\xi_{\tau_1,\tau_2}(\lambda_1, \lambda_2)$ of ViT-B-16 models pre-trained with different schemes. Light areas denote regions of weight space enjoying stronger weight disentanglement. The red box delimits the search space used to compute the best scaling coefficient $\lambda \in [-1, 1]$.

Pre-training factors We study the role of the following pre-training variables:

1. *Pre-training Algorithm:* Supervised, Self-Supervised (Masked Autoencoder (He et al., 2022) and DINO (Caron et al., 2021)), and Contrastive (CLIP). We include the CLIP encoder with a random classification head in order to *i)* directly compare task addition performance to the one obtained by CLIP as reported in Ilharco et al. (2022a) and Ortiz-Jimenez et al. (2024), where it is used in the usual open-vocabulary setting, and *ii)* compare pre-training with and without language supervision.

2. *Data Size:* ImageNet1k vs. ImageNet21k (Deng et al., 2009) and LAION-400M (Schuhmann et al., 2021) vs. LAION-2B (Schuhmann et al., 2022), noting that ImageNet and LAION are multiple orders of magnitude apart in terms of the number of samples in each dataset.

3. *Model Scale:* ViT-B-16 and ViT-L-14/16 (Dosovitskiy et al., 2020).

We provide a full list of the relevant model checkpoints in Table 2 of Appendix D.

4. Results

In this section, we first study whether closed-vocabulary models also exhibit weight disentanglement, allowing them to perform task arithmetic. Then, we test the effectiveness of task addition with closed-vocabulary models. Finally, we argue that simple baselines, such as linear probing alone, achieve competitive performance.

4.1. Weight Disentanglement vs. Pre-Training Scheme

To investigate the presence of weight disentanglement, we follow the methodology of Ortiz-Jimenez et al. (2024) and measure the weight disentanglement error between pairs of tasks $\xi_{\tau_1,\tau_2}(\lambda_1, \lambda_2)$, formally defined as

$$\sum_{t=1}^{2} \mathbb{E}_{x\sim\mu_t} d\left(f(x; \theta_{\text{pre}} + \lambda_t \tau_t), f(x; \theta_{\text{pre}} + \lambda_1\tau_1 + \lambda_2\tau_2)\right),$$

where μ_t denotes the input distribution of task t, $f(x, \theta)$ denotes the output function of the model, and d denotes the prediction error, i.e., $d(y_1, y_2) = \mathbb{1}(y_1 \neq y_2)$.

Figure 1 displays the disentanglement error for ViT-B-16 models fine-tuned from the different pre-training schemes, for all $\lambda_1, \lambda_2 \in [-3, 3]$. Light regions in the plot indicate areas of the weight space with strong disentanglement. We observe a significant presence of weight disentanglement in regions around the pre-trained initialization, located at the center of the plots, for all models. This finding suggests that weight disentanglement is a general property of pre-training.

When comparing the disentanglement error of different pre-trained models, we find that those relying on self-supervised pre-training techniques, such as MAE and DINO, achieve less disentanglement compared to models subject to standard supervised or contrastive pre-training. Notably, the large-scale pre-training of CLIP achieves the best weight disentanglement among the models considered. Furthermore, in Appendix F, we show that weight disentanglement strengthens as model sizes increase (Figure 4).

4.2. Task Addition with Closed-Vocabulary Models

Given the presence of weight disentanglement in closed-vocabulary models, we now turn to the study of task addition as outlined in Section 3. Table 1 presents the average single-task accuracy alongside the average task addition accuracy (both absolute and normalized[3]) for the different pre-trained schemes across all 8 tasks. Notably, task addition performance is high for all models, with the only exception being MAE, which retains only 73% of the fine-tuning perfor-

[3]Normalization is done w.r.t. the single-task (fine-tuning) accuracy, as outlined in Appendix E.

Table 1. **Task addition performance.** Evaluating the probing, single-task, and task addition accuracy across all 8 tasks (averaged). We vary the data size, training algorithm, and model scale, and indicate the scaling coefficient (λ) used in task addition for each model.

Model	Avg. Single-Task Accuracy (%)		Avg. Task Addition Accuracy (%)		λ
	Probing	*Final*	*Absolute*	*Normalized*	
ViT-B-16					
Supervised (IN1k)	72.1	90.0	73.8	82.0	0.05
Supervised (IN21k)	80.7	91.7	81.7	89.3	0.05
MAE (IN1k)	62.8	84.5	63.1	73.0	0.05
DINO (IN1k)	82.2	90.6	82.2	90.9	0.00
CLIP (LAION-400M)	**86.3**	**92.9**	**89.8**	**94.9**	0.10
CLIP (LAION-2B)	86.0	92.6	89.6	94.5	0.15
ViT-L-14/16					
Supervised (IN21k)	80.3	92.2	83.4	90.8	0.05
CLIP (LAION-2B)	**90.8**	**95.8**	**92.6**	**96.7**	0.15

mance of the single-task models.[4] The pre-trained CLIP vision encoder significantly outperforms all other models, followed by the ViT model pre-trained on Imagenet21k with standard supervision. Consistently with the weight disentanglement results, models pre-trained on larger pre-training corpora and with larger parameter sizes achieve better task addition performance.

A closer examination of the scaling coefficients of the task vectors λ reveals that their values are small, indicating small changes to the pre-trained visual encoder's weights. This suggests that the high addition performance can be largely attributed to the performance obtained by only probing the head while using the pre-trained encoder weights, implying a minor effect of task addition. Indeed, we notice that task addition accuracies are close to the average single-task accuracies obtained just after linear probing the classification head. Figure 2 shows the task addition normalized accuracies of all pre-trained models while varying the scaling coefficient λ. This plot indicates that task addition provides only a small (2-3%) performance gain over linear probing (which corresponds to setting $\lambda = 0$), or offers no advantage in the case of DINO pre-training. As λ increases beyond the optimal values, normalized accuracy decreases monotonically.

Probing vs. task addition Linear probing is a significantly cheaper strategy compared to closed-vocabulary task addition, as it only requires training a linear layer for each task while keeping the visual encoder frozen at its pre-trained state. Therefore, at the cost of 2-3 accuracy points, linear probing can be a competitive alternative to task addition in the closed-vocabulary setting for non-critical applications. Moreover, when comparing the average probing performance with that of task addition for open-vocabulary models (Ilharco et al., 2022a; Ortiz-Jimenez et al., 2024), we find that linear probing achieves considerably higher multi-task performance (see Table 4 in Appendix H). Crucially, as probing only updates the weights of the task-specific heads, the new knowledge acquired with fine-tuning is leveraged solely for those specific target tasks in isolation and with the specific class labels used during single-task fine-tuning. Thus, for users open to forgoing the modularity and flexibility of open-vocabulary models and interested only in high multi-task performance on a stationary set of target tasks, our results show that probing alone can be a valid alternative to task addition.

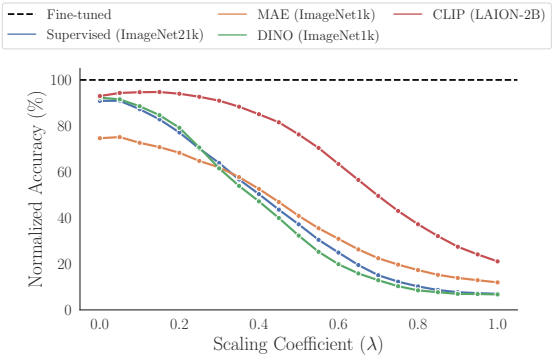

Figure 2. **Average normalized task addition accuracy for different pre-training algorithms as a function of the scaling coefficient** λ. The value $\lambda = 0$ corresponds to simple linear probing with no task vector added to the visual encoder.

5. Conclusion

In this work, we extended the application of task addition to a variety of closed-vocabulary image classification settings. We showed that weight disentanglement is not exclusive to

[4]Notice that this result aligns with the fact that MAE displays less weight disentanglement, cf. Figure 1.

vision-language contrastive pre-training but is a more general consequence of pre-training. Thus, task arithmetic can be used for editing a much larger class of models. Moreover, we demonstrated that simple linear probing can achieve competitive multi-task accuracy to both closed-vocabulary and open-vocabulary task addition at the expense of sacrificing the open-vocabulary nature of models like CLIP, which was not observed in previous studies.

In the future, understanding if task arithmetic can be applied to models sharing the same backbone (e.g., feature extractor) but differ in the final layers (e.g., image classification vs. segmentation) is an interesting extension of this work. Moreover, given the presence of weight disentanglement in models not undergoing large-scale pertaining such as CLIP, this study opens the avenue for studying the emergence of weight disentanglement during pre-training by performing controlled experiments in smaller-scale scenarios.

Acknowledgements

We thank Guillermo Ortiz-Jimenez for the many insightful discussions and his guidance throughout this project. We also thank the anonymous reviewers for their helpful feedback and comments.

References

Ainsworth, S., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2022.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *International Conference on Computer Vision*, pp. 9650–9660, 2021.

Cheng, G., Han, J., and Lu, X. Remote sensing image scene classification: Benchmark and state of the art. In *Proceedings of the IEEE*, volume 105, pp. 1865–1883, 2017.

Choshen, L., Venezian, E., Slonim, N., and Katz, Y. Fusing finetuned models for better pretraining. *arXiv preprint arXiv:2204.03044*, 2022.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Don-Yehiya, S., Venezian, E., Raffel, C., Slonim, N., and Choshen, L. Cold fusion: Collaborative descent for distributed multitask finetuning. In *Annual Meeting Of The Association For Computational Linguistics*, 2023.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Fort, S., Dziugaite, G. K., Paul, M., Kharaghani, S., Roy, D. M., and Ganguli, S. Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5850–5861, 2020.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269, 2020.

French, R. M. Catastrophic forgetting in connectionist networks. In *Trends in Cognitive Sciences*, volume 3, pp. 128–135, 1999.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Helber, P., Bischke, B., Dengel, A., and Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, volume 12, pp. 2217–2226, 2019.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2022a.

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems*, volume 35, pp. 29262–29277, 2022b.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence*, pp. 876–885, 2018.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision workshops*, pp. 554–561, 2013.

Kumar, A., Raghunathan, A., Jones, R. M., Ma, T., and Liang, P. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2021.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Matena, M. S. and Raffel, C. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems*, 2022.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A. Y., et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 7. Granada, Spain, 2011.

Ortiz-Jiménez, G., Modas, A., Moosavi-Dezfooli, S.-M., and Frossard, P. Optimism in the face of adversity: Understanding and improving deep learning through adversarial robustness. In *Proceedings of the IEEE*, volume 109, pp. 635–659, 2021.

Ortiz-Jimenez, G., Favero, A., and Frossard, P. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

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

Ramé, A., Ahuja, K., Zhang, J., Cord, M., Bottou, L., and Lopez-Paz, D. Model ratatouille: Recycling diverse models for out-of-distribution generalization. In *International Conference on Machine Learning*, pp. 28656–28679, 2023.

Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2021.

Santurkar, S., Tsipras, D., Elango, M., Bau, D., Torralba, A., and Madry, A. Editing a classifier by rewriting its prediction rules. In *Advances in Neural Information Processing Systems*, volume 34, pp. 23359–23373, 2021.

Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 25278–25294, 2022.

Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. In *Neural networks*, volume 32, pp. 323–332, 2012.

Wang, K., Dimitriadis, N., Ortiz-Jiménez, G., Fleuret, F., and Frossard, P. Localizing task information for improved model merging and compression. In *International Conference on Machine Learning*, 2024.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998, 2022a.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. Robust fine-tuning of zero-shot models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7959–7971, 2022b.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A comprehensive survey on transfer learning. In *Proceedings of the IEEE*, volume 109, pp. 43–76, 2020.

A. Further Related Work

Significant progress has been made in studying weight interpolation techniques to enhance the capabilities of pre-trained models. Recent studies demonstrate that interpolating between fine-tuned weights and their pre-trained initializations can improve single-task performance (Matena & Raffel, 2022; Ramé et al., 2023; Wortsman et al., 2022b; Izmailov et al., 2018; Frankle et al., 2020). Similarly, averaging weights from multiple independently fine-tuned models has been shown to produce high-performing multi-task models (Ilharco et al., 2022a; Ortiz-Jimenez et al., 2024; Wang et al., 2024; Yadav et al., 2024; Ilharco et al., 2022b; Wortsman et al., 2022a). Such techniques can provide better parameter initializations for later training (Don-Yehiya et al., 2023; Choshen et al., 2022), as well as reduce catastrophic forgetting (French, 1999). Notably, task arithmetic, a multi-task weight interpolation technique, is modular, allowing it to directly accept checkpoints downloaded from online repositories in a plug-and-play fashion. However, notice that successful interpolation requires alignment between the models being merged, ensuring they share a common optimization path early in training (Fort et al., 2020; Ainsworth et al., 2022).

B. Zero-shot Head Initialization

To initialize a zero-shot head for open-vocabulary models like CLIP, the following approach is typically used:

1. For each task, collect its associated set of textual classes (e.g., $C = \{$Airplane, Car, Bird, Cat, Dog, ...$\}$ of size N and specify a set of templates into which the classes can be plugged (e.g., $T = \{$A photo of [class], A blurry photo of [class], A photo of a big [class], ...$\}$

2. Use the model's text encoder to obtain embeddings of dimension d_{emb} for each class and each template.

3. Compute the average of these embeddings across each class to obtain a general text embedding of all classes $W_{\text{text}} \in \mathbb{R}^{d_{\text{emb}} \times N}$

4. Build an empty classification head and initialize its weights using W_{text}, connecting it to the model's image encoder.

5. Discard the text encoder and fine-tune the image encoder only (freezing the zero-shot head).

C. Experimental Hyperparameters

We follow the same hyperparameter setting as Ortiz-Jimenez et al. (2024) and Ilharco et al. (2022a). Namely, for each model configuration, we fine-tune all datasets starting from the same model checkpoint. We fine-tune (and probe when applicable) for 2,000 iterations with a batch size of 128 using the AdamW optimizer (Loshchilov & Hutter, 2018), and a learning rate determined by line search on the Stanford Cars (Krause et al., 2013) dataset (Probing: $\{$1e-3, 3e-3, 1e-2, 3e-2, 1e-1, 3e-1$\}$ and full or encoder fine-tuning: $\{$1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1$\}$) under a cosine annealing learning rate schedule with 200 warmup steps.

D. Model Checkpoints

In Table 2, we list the model checkpoints used in this study.

E. Normalized Task Addition Accuracy

The normalization of the task addition accuracy is done with respect to the average single-task accuracy obtained by independently fine-tuning on each task. In particular,

$$\text{Normalized accuracy} = \frac{1}{T}\sum_{t=1}^{T} \frac{\underset{x\sim\mu_t}{\text{acc}}\left[f(x;\theta_{\text{pre}} + \sum_{t'}\tau_{t'})\right]}{\underset{x\sim\mu_t}{\text{acc}}\left[f(x;\theta_{\text{pre}} + \tau_t)\right]}. \tag{1}$$

Table 2. **HuggingFace Model Checkpoints.** The URL of each checkpoint is also provided as a hyperlink.

Pre-training Scheme	Architecture	HuggingFace Repository
Supervised (IN21k)	ViT-B-16	timm/vit_base_patch16_224.augreg_in21k
	ViT-L-16	google/vit-large-patch16-224-in21k
Supervised (IN1k)	ViT-B-16	timm/vit_base_patch16_224.augreg_in1k
MAE (IN1k)	ViT-B-16	facebook/vit-mae-base
DINO (IN1k)	ViT-B-16	facebook/dino-vitb16
CLIP (LAION-400M)	ViT-B-16	laion/Model-B-16_Data-400M_Samples-34B_lr-1e-3_bs-88k
CLIP (LAION-2B)	ViT-B-16	laion/Model-B-16_Data-2B_Samples-34B_lr-1e-3_bs-88k

F. Further Weight Disentanglement Results

F.1. Effect of Pre-training Data Size

We compare the weight disentanglement error of a Supervised ViT-B-16 pre-trained on ImageNet1k vs. ImageNet21k (∼1 order of magnitude difference in data size) in Figure 3. While the observed gain in WD strength might seem minimal, the results in Tables 1 and 3 indicate that the ImageNet21k model can achieve higher task addition accuracy compared to the ImageNet1k ViT.

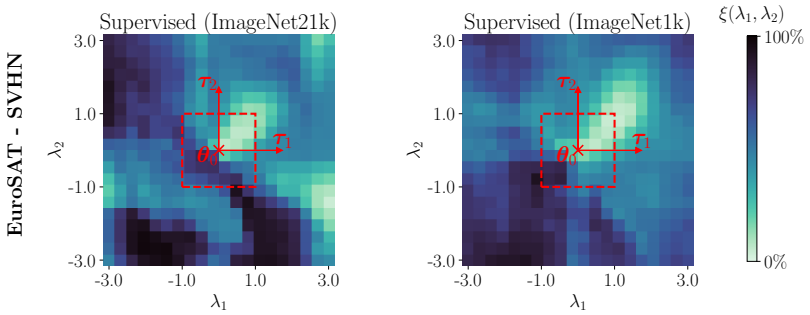

Figure 3. **Weight disentanglement error heatmaps for Supervised pre-training on ImageNet1k vs. ImageNet21k.** Both models are based on a ViT-B-16 architecture. The red box delimits the search space used to compute the best scaling coefficient λ.

F.2. Effect of Model Scale

Similar to Appendix F.1, we compare the error of two supervised models with the same pre-training dataset (ImageNet21k), but with a varying number of parameters (ViT-B-16 vs. ViT-L-16). We can clearly observe that the larger ViT-L-16 is significantly more weight disentangled than its smaller counterpart. This finding is supported by the results in Tables 1 and 3, which show that the supervised ViT-L-16 admits a task addition advantage over ViT-B-16.

G. Full Fine-tuning

Table 3 reports the task addition performance under the full-fine tuning regime, wherein both the image encoder and the classification head are simultaneously fine-tuned. While task addition achieves non-trivial accuracy with full fine-tuning, it significantly loses downstream performance when comparing each model's absolute task addition accuracy to its average single-task accuracy, as opposed to the results obtained with aligned fine-tuning (Table 1). Notably, if we consider full fine-tuning for CLIP (LAION-2B) in the *open* setting (Table 4; starting from CLIP's zero-shot head initialization), we can see that it achieves around 72% absolute addition accuracy for ViT-B-16, compared to 47% in the closed setting (with similar single-task performance). This suggests that the full fine-tuning approach might be inadequate to fully leverage the potential of closed-task arithmetic.

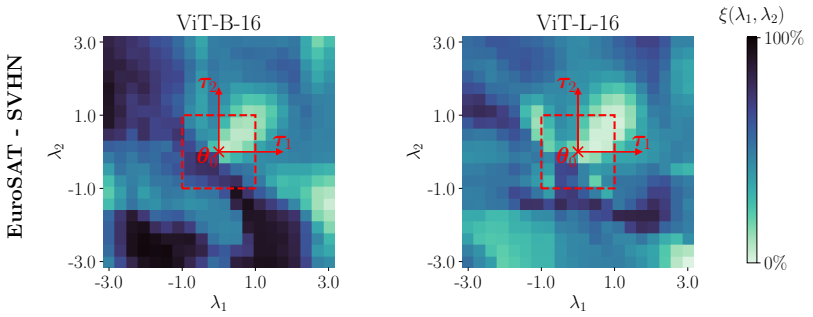

Figure 4. **Weight disentanglement error heatmaps highlighting the effect of model scale.** The two models are ViT-B and ViT-L, both pre-trained on ImageNet21k in a supervised manner. The red box delimits the search space used to compute the best scaling coefficient λ.

Table 3. **Full Fine-tuning: Evaluating the single-task and task addition accuracy across all tasks (averaged).** We vary the data size, training algorithm, and model scale. We also indicate the scaling coefficient (λ) used in task addition as done in Table 1.

Model	Avg. Single-Task Accuracy (%)	Avg. Task Addition Accuracy (%)		λ
		Absolute	*Normalized*	
ViT-B-16				
Supervised (IN1k)	87.9	44.5	46.5	0.35
Supervised (IN21k)	90.2	**55.5**	**58.5**	0.25
MAE (IN1k)	82.1	25.3	26.9	0.25
DINO (IN1k)	87.9	27.9	28.8	0.25
CLIP (LAION-400M)	91.6	45.8	47.1	0.30
CLIP (LAION-2B)	**92.0**	47.0	48.2	0.30
ViT-L-14/16				
Supervised (IN21k)	90.9	59.9	62.2	0.25
CLIP (LAION-2B)	**94.1**	**70.9**	72.9	0.40

H. Task Addition with Open-vocabulary CLIP

We perform task addition on open-vocabulary CLIP under two different fine-tuning regimes: Encoder fine-tuning (as done in Ilharco et al. (2022a) and Ortiz-Jimenez et al. (2024), wherein the classification head is frozen) and full fine-tuning. We report the results in Table 4.

Table 4. **Open CLIP Fine-tuning.** Task addition on open-vocabulary CLIP (ViT-B-16 on LAION-2B).

Fine-tuning Regime	Average Single-Task Accuracy (%)		Average Task Addition Accuracy (%)		λ
	Zeroshot	*Final*	*Absolute*	*Normalized*	
Encoder	54.42	91.36	72.36	77.21	0.25
Full	54.42	91.63	71.16	75.89	0.25

I. Varying the Number of Tasks: Extra Experiments

Our results in Figure 5 reveal a strong presence of weight disentanglement in a region around the pre-trained initialization of models that follow the full fine-tuning regime. This finding suggests that the suboptimal performance of task addition under this regime might be a direct result of evaluating too many tasks. To verify this claim, we evaluate task addition while

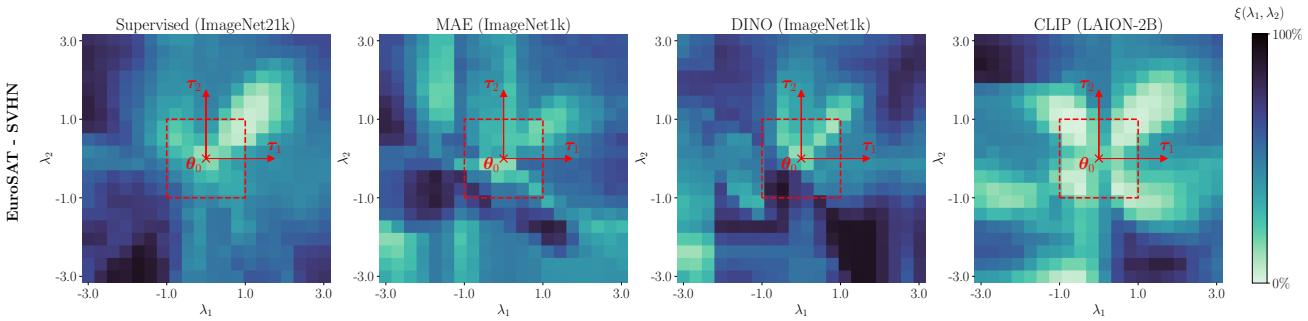

Figure 5. **Full Fine-tuning: Weight disentanglement error heatmaps for the different pre-training algorithms.** All models are based on a ViT-B-16 architecture following the full fine-tuning regime. The red box delimits the search space used to compute the best scaling coefficient λ.

varying the number of merged tasks. In Figure 6, we plot task addition performance for a Supervised ViT-B-16 pre-trained ImageNet21k and multiple different combinations of tasks. We observe that performance is high when the number of tasks is small and follows linear decay with the number of tasks.

In general, in Figure 7, we observe that most of our models, except for MAE, maintain high normalized task addition accuracy for 2-3 tasks, and this performance drops as we add more tasks. We also show, in Figures 8 and 9 respectively, that increasing the model scale from 86M parameters (ViT-B-16) to 300M (ViT-L-16) and the pre-training data size from 1M samples (ImageNet1k) to 14M (ImageNet21k) both yield small asymptotic improvements in task addition accuracy.

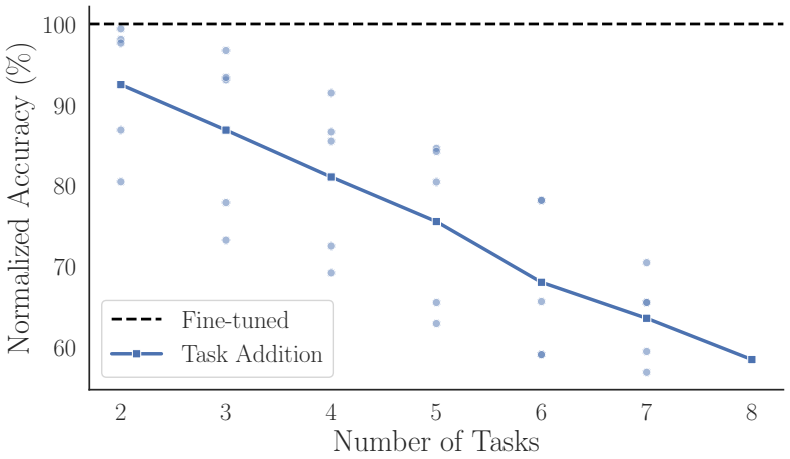

Figure 6. **Task addition performance for Supervised ViT-B-16 (ImageNet21k) under a varying number of tasks.** The solid line represents the mean accuracy, while the dots represent different combinations of tasks.

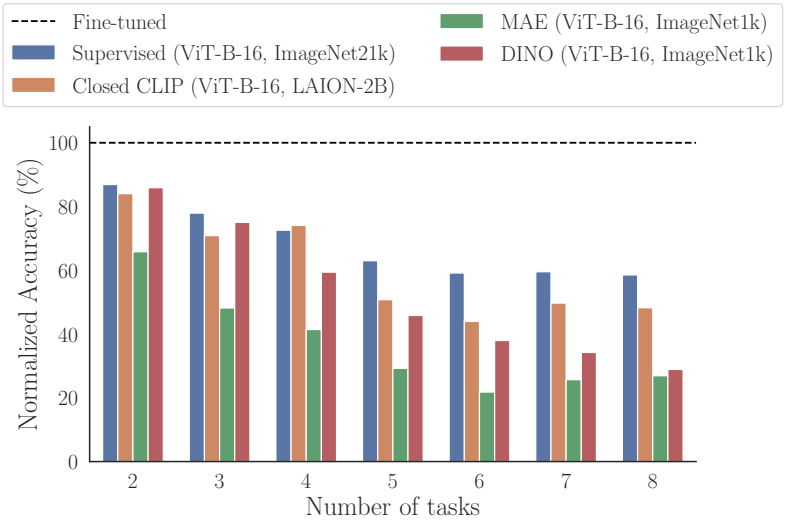

Figure 7. Task addition performance for all pre-training settings under a varying number of tasks.

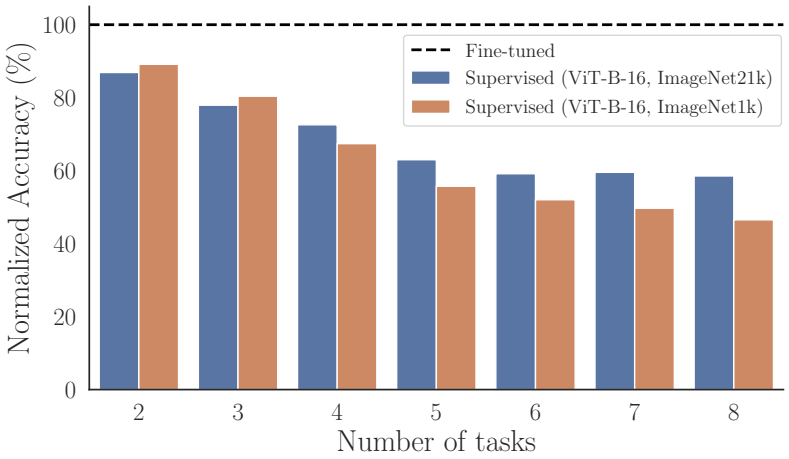

Figure 8. Normalized task addition accuracy for Supervised ViT-B-16 pre-trained on ImageNet1k vs. ImageNet21k under a varying number of tasks.

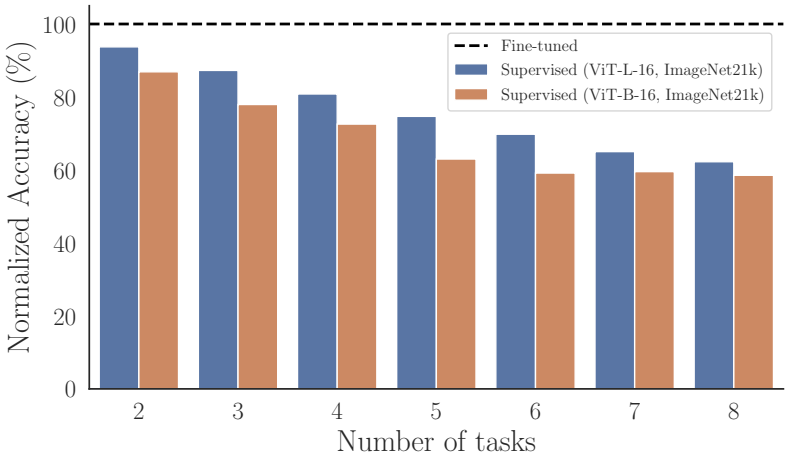

Figure 9. Normalized task addition accuracy for a Supervised ViT-B-16 vs. ViT-L-16 pre-trained on ImageNet21k under a varying number of tasks.