NegMerge: Consensual Weight Negation for Strong Machine Unlearning

Hyoseo Kim[1*], **Dongyoon Han** [2†], **Junsuk Choe** [1†]
[1]Sogang University [2]NAVER AI Lab

Abstract

Machine unlearning aims to selectively remove specific knowledge from a model. Current methods, such as task arithmetic, rely on fine-tuning models on the forget set, generating a task vector, and subtracting it from the original model. However, we argue the effectiveness of this approach is highly sensitive to hyperparameter selection, necessitating careful validation to identify the best model among many fine-tuned candidates. In this paper, we propose a novel method that leverages all given fine-tuned models rather than selecting a single one. By constructing task vectors from models trained with varied hyperparameters and merging only the components of the task vectors with consistent signs, we perform unlearning by negating the merged task vector from the original model. Given that existing methods also utilize multiple fine-tuned models, our approach delivers more effective unlearning without incurring additional computational costs. We demonstrate the effectiveness of our method on both vision-language models and standard image classification models, showing improved unlearning performance with minimal degradation on the retain set, outperforming state-of-the-art techniques.

1 Introduction

Recent advances in pre-training [8, 9, 27, 25, 1, 23] have achieved remarkable performance, primarily driven by the use of large-scale datasets. However, the datasets often include underfiltered, unwanted, or sensitive private information, which raises critical concerns about privacy protection. The *Right to be Forgotten* regulation [14] allows individuals to request the deletion of their personal data. However, applying this concept to machine learning models is challenging because the training process deeply embeds the data into the model's parameters, making it difficult to remove its influence. The most straightforward solution is to remove the data from the training set and retrain the model from scratch, which requires enormous computational resources. As a result, ensuring that models forget learned patterns becomes a challenging task. *Machine unlearning* [30, 11, 29, 19, 17, 4, 10] offers a solution by enabling models to erase specific knowledge without the need for full retraining.

Despite promising results, many existing methods struggle to remove only the target knowledge while preserving the rest. This challenge arises because fine-tuning often disrupts knowledge in the *retain set* (i.e., remaining data) during attempts to erase knowledge from the *forget set* (i.e., data to be forgotten) [4, 10]. A known method robust to this issue is task arithmetic [15], where direct fine-tuning of the model is avoided. Instead, this method calculates a task vector – the parameter-wise difference between the original model and a model fine-tuned on the forget set. The task vector is then subtracted from the original model through a negation operation. This process, referred to as *forgetting by negation*, has demonstrated strong unlearning performance while preserving the model's knowledge, similar to continual learning researches [18, 2] addressing catastrophic forgetting [18]. However, we argue that task arithmetic has limitations; not all fine-tuned models are suitable for task vectors, and thus, unlearning performance is highly sensitive to hyperparameter setups used for

[*]Work done during an internship at NAVER AI Lab. † Corresponding author.

38th Conference on Neural Information Processing Systems Workshop on Adaptive Foundation Models.

fine-tuning. As a result, searching for an optimal hyperparameter set for effective unlearning can be both time-consuming and computationally costly.

To address these limitations, we propose a novel method, `NegMerge`, that improves the process of forgetting by negation. We argue that relying on a single optimal model, as current methods [15, 26] do, is not truly optimal. Hyperparameter tuning generates multiple fine-tuned models, and instead of selecting just one, we suggest leveraging all of them. Specifically, we compute the final task vector by merging multiple task vectors derived from the fine-tuned models. This approach draws inspiration from model merging techniques [31, 34, 16], which similarly utilize multiple fine-tuned models to enhance performance. By extending this concept to machine unlearning, we provide a more effective solution. Specifically, unlike these existing techniques, we only combine elements with consistent signs across the task vectors while masking elements with inconsistent signs to zero.

We demonstrate the effectiveness of our approach in two experimental settings. The first involves unlearning specific knowledge from a vision-language model like CLIP [27]. The second focuses on unlearning knowledge from specific data points in a general image classification network [4, 10]. We validate our method using the ViT [9] and ResNet [12] architectures across nine datasets. In both settings, our approach achieves new state-of-the-art performance while using similar or fewer computational resources than existing methods.

2 Method

2.1 Background

Task Arithmetic. Task arithmetic [15] defines a *task vector* $\tau_t = \theta^t_{ft} - \theta_{pre}$. Specifically, the vectors are the result of subtracting (negating) the weights of a pre-trained model θ_{pre} from those of a model θ^t_{ft} fine-tuned on a target task t. We can adjust the model in the desired direction by adding or subtracting the sum of these task vectors $\tau = \sum_t \tau_t$ from the original model's weights, according to the formula $\theta_{new} = \theta_{pre} + \lambda\tau$. This approach is more computationally efficient than fine-tuning, as it leverages pre-trained models from public repositories and eliminates the need for additional training.

Our Unlearning scenarios. In our study, we explore two distinct unlearning scenarios. The first scenario is the one described above, where a vision-language model like CLIP [27] is made to forget the knowledge of a specific class. For this scenario, we adopt the evaluation protocol for unlearning proposed in the original paper [15]. The other scenario involves a standard image classification network like ResNet [12] trained using cross-entropy loss on images and class labels. In this case, the model is made to forget the knowledge of specific training data. Here, we calculate the task vectors by fine-tuning the model only using the forget set: $\theta_{unlearn} = \theta_{ori} - \lambda(\theta^{forget}_{ft} - \theta_{ori})$ for both scenarios.

2.2 `NegMerge`: Improved Task Arithmetic For Machine Unlearning

Given multiple models fine-tuned on the forget set, which applied various training configurations to ensure diversity among the fine-tuned models, we propose a method that neatly aggregates the model for effective unlearning. Our proposed method, `NegMerge`, consists of the following steps: 1) We calculate the task vectors using all the fine-tuned models, 2) We identify the elements corresponding to the forget set in each vector. 3) Finally, we compute final task vector by using the identified elements, and perform machine unlearning by subtracting this final task vector from the original model. We provide a detailed description of each step below, and Figure 1 illustrates the overview of our method.

Preparaing Diverse Fine-Tuned Models. There are numerous methods for preparing diverse fine-tuned models on the forget set. A simple yet effective approach is just altering hyperparameters such as learning rate and the number of epochs or employing data augmentation techniques like RandAugment [7] and CutMix [35]. In this work, we focus on making minimal adjustments to the existing training setup, either by modifying RandAugment parameters or adjusting training configurations like the number of epochs. Further details on these adjustments can be found in Section 3.1. While additional techniques could further enhance model diversity and improve unlearning performance, these are left for future exploration.

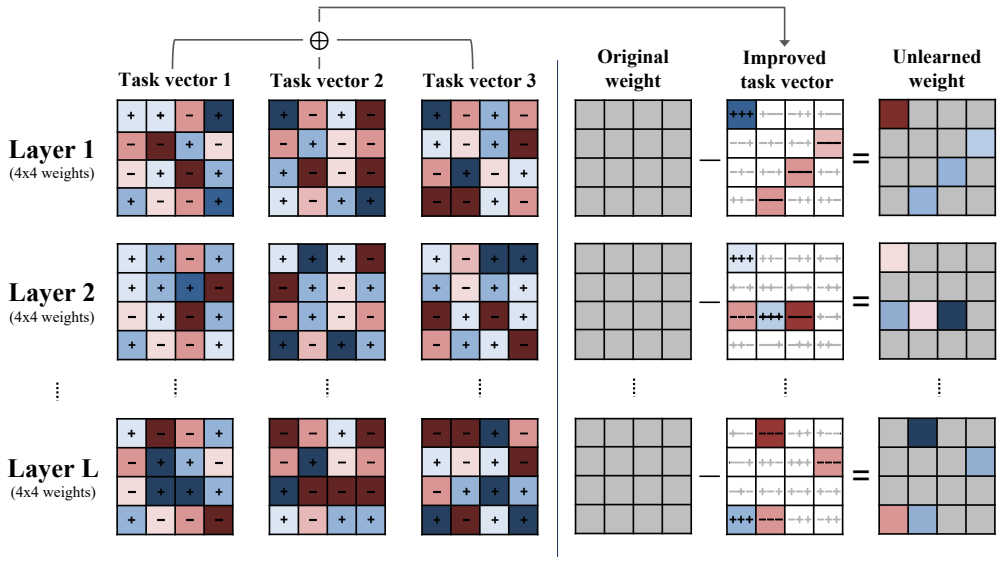

Figure 1: **Illustration of the proposed method.** Our `NegMerge` enhances task arithmetic by computing an improved task vector. Specifically, 1) multiple task vectors derived from fine-tuned models trained with different hyperparameters are utilized. 2) we compute the improved task vector by merging ($\bigoplus$) only the elements that retain a consistent sign across task vectors, while masking elements with differing signs to 0. 3) this refined task vector is used for negation from the original weights. The color intensity in the cells reflects the magnitude of the task vector elements; darker blue represents larger positive values, lighter blue indicates smaller positives, while darker red represents larger negative values, and lighter red indicates smaller negatives.

Identifying Elements in the Task Vector Corresponding to the Forget Set. We derive task vectors from the fine-tuned models and analyze them to determine which elements (in weights) correspond to the forget set. We conjecture that elements that consistently show the same sign across task vectors are attributed to the forget set, as each model is trained specifically to align with this set, regardless of the training configurations. On the other hand, components that exhibit differing signs are considered less related to the forget set, as their variations are more likely a result of different training configurations rather than supervision from the forget set.

Final Task Vector for Negation. We compute the final task vector using the following formulation:

$$\tau_{\text{merged}} = \left(\frac{1}{n} \sum_{k=1}^{n} \tau_k \right) \odot \mathbf{1}_{\text{signs are equal}}, \tag{1}$$

where n is the number of task vectors, $\odot$ denotes the Hadamard product (element-wise multiplication), and the vector $\mathbf{1}_{\text{signs are equal}}$ acts like a filter, containing 1 for elements where the signs of the corresponding components across all task vectors τ_k are the same and 0 where the signs differ[2]. As a result, only the components with consistent signs across all task vectors contribute to the final task vector, while those with differing signs are excluded by being set to zero. We then perform machine unlearning by negating this final task vector to the original model [15].

3 Experiment

3.1 Experimental Setups

Datasets and Backbones. In the CLIP scenario (referred to as the scenario using a vision-language model), we follow the training and evaluation protocols of [15]. We assess unlearning performance on eight datasets: SUN397 [32], Cars [20], RESISC45 [5], EuroSAT [13], SVHN [36], GTSRB [28], MNIST [22], and DTD [6]. We use the pre-trained CLIP ViT-B/32, B/16, L/14 models [27] for

[2]This operation is based on sign unanimity and could be adjusted with additional hyperparameters to allow partial consensus, we opt for a simpler approach.

Table 1: **Unlearning Performance on CLIP ViT Models.** Results are shown for CLIP ViT-{B/32, B/16, L/14}, reporting average accuracy (%) on the eight target tasks we wish to forget (Cars, DTD, EuroSAT, GTSRB, MNIST, RESISC45, SUN397, and SVHN), and the control task to remain (ImageNet). We compare our method with Task Arithmetic [15], Linear Task Arithmetic [26], Uniform Merge [31], Greedy Merge [31], TIES-Merging [33], and MagMax [24]. * indicates that the numbers are borrowed from the original papers. † denotes the best results achieved through hyperparameter search. ‡ combines models in descending order of losses. Time denotes the merging time, measured in seconds, taken to merge 30 models on the Cars dataset using CLIP ViT-B/32, which is averaged over three runs.

Method	ViT-B/32		ViT-B/16		ViT-L/14		Time (sec)
	Acc D_f(↓)	Acc D_r(↑)	Acc D_f(↓)	Acc D_r(↑)	Acc D_f(↓)	Acc D_r(↑)	
Pre-trained	48.13	63.33	55.49	68.32	65.19	75.54	-
Task Arithmetic							
*Paper number**	24.00	60.90	21.30	65.40	19.00	72.90	-
Single Best Model†	23.63	60.60	20.64	64.04	19.17	72.09	-
Uniform Merge	22.50	60.55	21.51	64.60	18.10	71.91	$12_{\pm 0.1}$
Greedy Merge‡	23.31	60.75	21.34	64.54	17.71	71.99	$607_{\pm 2.6}$
TIES-Merging	26.21	61.08	23.78	64.72	22.70	72.41	$128_{\pm 10.1}$
MagMax	25.24	60.95	24.45	64.78	21.71	72.55	$24_{\pm 1.8}$
`NegMerge` **(Ours)**	**20.76**	60.36	**19.24**	64.54	**17.32**	72.08	$37_{\pm 1.2}$
Linear Task Arithmetic							
*Paper number**	10.90	60.80	11.30	64.80	-	-	-
Single Best Model†	8.88	60.16	6.92	64.62	-	-	-
Uniform Merge	9.12	60.47	6.84	65.26	-	-	$19_{\pm 2.3}$
Greedy Merge‡	8.73	60.27	6.80	64.72	-	-	$1696_{\pm 35.3}$
TIES-Merging	10.66	60.38	8.44	65.12	-	-	$378_{\pm 8.0}$
MagMax	11.33	60.67	8.65	65.17	-	-	$164_{\pm 2.4}$
`NegMerge` **(Ours)**	**8.03**	60.58	**6.60**	65.40	-	-	$194_{\pm 1.6}$

these experiments. In the standard classifier scenario, we evaluate unlearning performance on CIFAR-10 [21] using a ResNet-18 [12] model.

Baselines and Metrics. For the CLIP scenario, we compare our method with five existing methods: Task Arithmetic [15], Uniform Merge [31], Greedy Merge [31], TIES-Merging [33], and MagMax [24]. For the Greedy Merge, we rank models by their loss on the retain set and merge them in a direction that minimizes this loss. We evaluate performance by measuring accuracy on the forget set D_f and the retain set D_r.

In the standard classifier scenario, we follow [10] to compare our method against eight unlearning techniques: Fine-tuning [30], Random Labeling [11], Gradient Ascent [29], Influence Unlearning [19], $\ell 1$-sparse [17], Boundary Shrink and Expand [4], and SalUn [10]. We also compare against Task Arithmetic [15] and Uniform Merge [31]. The objective is to match the unlearned model's performance to that of a fully retrained model. We use the accuracies of the retain set D_r, forget set D_f, and test set D_{test} to evaluate performance. To assess privacy protection, we employ the Membership Inference Attack (MIA) metric [3], aiming to achieve similar results to the fully retrained model.

Implementation Details. In the CLIP scenario, for fine-tuning, we set the batch size to 128 and use a learning rate of 1e-5 with a cosine annealing schedule. We utilize the AdamW optimizer, applying a weight decay of 0.1. During fine-tuning, the output of CLIP's text encoder, specifically the final classification layer, remains frozen. We enhance the diversity of the fine-tuned models by adjusting the configurations of RandAugment.

In the standard image classifier unlearning scenario, for the CIFAR-10 dataset, we set the batch size to 256 and the learning rate to 0.05. Since CIFAR-10 has relatively lower image quality, we do not apply data augmentation. Instead, we vary the training hyperparameters.

Table 2: **Unlearning Performance for 10% Random Data Forgetting on CIFAR-10 using ResNet-18.** The results are expressed as a±b, representing the mean (a) and standard deviation (b) across three independent trials. The Avg. Gap is computed as the average of the performance differences observed in various accuracy-related metrics, including Acc D_r, Acc D_f, Acc D_{test}, and MIA. These metrics are favorable when they are close to the performance of the *Retrain model* ($\simeq$). * indicates that the numbers are borrowed from [10]. † denotes the best results achieved through hyperparameter search.

Methods	Used Splits	Acc $D_r(\simeq)$	Acc $D_f(\simeq)$	Acc $D_{test}(\simeq)$	MIA($\simeq$)	Avg. Gap($\downarrow$)
Retrain *	retain	$100.00_{\pm 0.00}$	$94.76_{\pm 0.69}$	$94.26_{\pm 0.02}$	$12.88_{\pm 0.09}$	0.00
Random Labeling *	all	$99.67_{\pm 0.14}$	$92.39_{\pm 0.31}$	$92.83_{\pm 0.38}$	$37.36_{\pm 0.06}$	7.15
Influence *	all	$99.20_{\pm 0.22}$	$98.93_{\pm 0.28}$	$93.20_{\pm 1.03}$	$2.67_{\pm 0.01}$	4.06
SalUn *	all	$99.62_{\pm 0.12}$	$97.15_{\pm 0.43}$	$93.93_{\pm 0.29}$	$14.39_{\pm 0.82}$	1.15
Finetune *	retain	$99.88_{\pm 0.08}$	$99.37_{\pm 0.55}$	$94.06_{\pm 0.27}$	$2.70_{\pm 0.01}$	3.78
$\ell 1$-sparse *	retain	$97.74_{\pm 0.33}$	$95.81_{\pm 0.62}$	$91.59_{\pm 0.57}$	$9.84_{\pm 0.00}$	2.26
Gradient Ascent *	forget	$99.50_{\pm 0.38}$	$99.31_{\pm 0.54}$	$94.01_{\pm 0.47}$	$1.70_{\pm 0.01}$	4.12
Boundary Shrink *	forget	$98.29_{\pm 2.50}$	$98.22_{\pm 2.52}$	$92.69_{\pm 2.99}$	$8.96_{\pm 0.13}$	2.67
Boundary Expanding *	forget	$99.42_{\pm 0.33}$	$99.41_{\pm 0.30}$	$93.85_{\pm 1.02}$	$7.47_{\pm 1.15}$	2.76
Random Labeling	forget	$99.99_{\pm 0.00}$	$99.98_{\pm 0.02}$	$95.04_{\pm 0.11}$	$2.15_{\pm 1.94}$	4.19
SalUn	forget	$99.88_{\pm 0.04}$	$99.89_{\pm 0.04}$	$94.42_{\pm 0.05}$	$9.51_{\pm 2.07}$	2.20
Task Arithmetic						
Single Best Model†	forget	$98.36_{\pm 0.51}$	$94.85_{\pm 0.16}$	$91.49_{\pm 0.80}$	$10.91_{\pm 0.72}$	1.62
Uniform Merge	forget	$98.70_{\pm 0.91}$	$95.83_{\pm 2.17}$	$92.36_{\pm 1.16}$	$10.14_{\pm 2.93}$	1.75
TIES-Merging	forget	$98.38_{\pm 0.17}$	$95.45_{\pm 0.32}$	$92.23_{\pm 0.14}$	$9.36_{\pm 0.31}$	1.96
MagMax	forget	$98.38_{\pm 0.12}$	$97.97_{\pm 0.77}$	$91.53_{\pm 0.00}$	$8.45_{\pm 2.60}$	3.00
`NegMerge` **(Ours)**	forget	$99.15_{\pm 0.24}$	$96.63_{\pm 0.59}$	$92.71_{\pm 0.39}$	$12.87_{\pm 1.29}$	**1.07**

3.2 Experimental Results

CLIP Unlearning Scenario. Table 1 presents the evaluation results across three variants of the CLIP model (ViT-B/32, ViT-B/16, and ViT-L/14), demonstrating strong generalizability. In the ViT-B/32 model, our method reduces the accuracy on the forget set D_f to 20.76%. This outperforms Task Arithmetic (23.63%), Uniform Merge (22.50%), and Greedy Merge (23.31%). In the ViT-B/16 and ViT-L/14 models, our method continues to outperform competitors, achieving the lowest accuracies on D_f at 19.24% and 17.32%, respectively.

Standard Classifier Unlearning Scenario. Table 2 compares unlearning techniques on CIFAR-10 using ResNet-18, where random 10% of the training set is targeted for forgetting. The fully retrained model is the benchmark, evaluating performance on retain set D_r, forget set D_f, test set D_{test}, and MIA score. Our method achieves the lowest average gap at 1.07, closely mimicking the retrained model's performance across all metrics and proving effective in both unlearning and preserving generalization without relying on the retain set.

4 Conclusion

In this paper, we propose a novel machine unlearning technique, `NegMerge`, based on task arithmetic and model merging. We hypothesize that multiple fine-tuned models are necessary for effective unlearning, based on the observation of a trade-off between accuracy on the forget set and the retain set. Building on the fact that existing techniques generate numerous fine-tuned models through validation using various hyperparameters, we propose a method that utilizes all derived fine-tuned models. Assuming that elements with consistent signs across task vectors obtained from the fine-tuned models are related to the forget set, we merge only those elements. This approach enables us to compute task vectors that fit the forget set more effectively while preserving the knowledge in the retain set, thus overcoming the trade-off. We then perform forgetting by negation with the merged task vector. Our `NegMerge` is tested on the CLIP ViT models and the standard ResNet18 classifier, achieving new state-of-the-art performance across nine datasets.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018.

[3] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE, 2022.

[4] Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7766–7775, 2023.

[5] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[6] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014.

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[8] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

[10] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Dennis Wei, Eric Wong, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023.

[11] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[14] Chris Jay Hoofnagle, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. The european union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98, 2019.

[15] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

[16] Dong-Hwan Jang, Sangdoo Yun, and Dongyoon Han. Model stock: All we need is just a few fine-tuned models. *arXiv preprint arXiv:2403.19522*, 2024.

[17] Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsification can simplify machine unlearning. *arXiv preprint arXiv:2304.04934*, 2023.

[18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[19] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

[20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[22] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[24] Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. *arXiv preprint arXiv:2407.06322*, 2024.

[25] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[26] Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36, 2024.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[28] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, pages 1453–1460. IEEE, 2011.

[29] Anvith Thudi, Gabriel Deza, Varun Chandrasekaran, and Nicolas Papernot. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 303–319. IEEE, 2022.

[30] Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021.

[31] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022.

[32] Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 119:3–22, 2016.

[33] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36, 2024.

[34] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. *arXiv preprint arXiv:2310.02575*, 2023.

[35] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[36] Netzer Yuval. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.