Machine Unlearning *via* Task Simplex Arithmetic

Junhao Dong[1,2], Hao Zhu[3], Yifei Zhang[1], Xinghua Qu[3],
Yew-Soon Ong[1,2*], and Piotr Koniusz[4,5,6*]
[1]Nanyang Technological University, [2]CFAR, IHPC, A*STAR, [3]Bytedance,
[4]Data61♥CSIRO, [5]University of New South Wales, [6]Australian National University

Abstract

As foundation Vision-Language Models (VLMs) unlock fine-tuning on smaller datasets while leveraging large-scale pre-training data, machine unlearning becomes critical in addressing privacy concerns and regulatory compliance. Task vector, representing the difference between parameters of models fine-tuned with and without specific data, is a popular retraining-free unlearning strategy. However, we observe that task vectors exhibit substantial sensitivity to various fine-tuning configurations, resulting in unstable unlearning effectiveness that correlates negatively with the prediction-level variance. While aggregating multiple functions (*e.g.*, VLM with classifier) whose parameters are represented by different task vectors reduces function variance and improves unlearning, the computational cost of obtaining numerous task vectors and aggregating functions is computationally high. Thus, in order to capture the space of task vectors induced by diverse fine-tuning strategies, we propose modeling it within the convex hull of $(Q-1)$-simplex whose vertices represent Q task vectors. Although a function ensemble can be formed by sampling numerous task vectors from such a simplex, we derive a closed-form ensemble of an infinite number of functions whose parameters are uniformly sampled from the simplex, enabling efficient function-level task vector ensembling with enhanced unlearning performance. Extensive experiments and analyses across diverse datasets and scenarios demonstrate the efficacy of our method.

1 Introduction

Vision-Language Models (VLMs), such as CLIP [52] and its derivatives [39, 41] including adversarially fine-tuned VLMs [45, 13, 14, 20, 12, 15], enjoy remarkable generalization due to pre-training on massive-scale datasets [65]. Given their superior performance on various tasks, including notable robustness in single-modal settings [17, 16, 19, 10, 18, 9], significant privacy concerns and regulatory compliance challenges arise regarding *data erasure rights* mandated by regulations [53, 2]. Consequently, developing efficient unlearning models that can effectively "forget" specific data is critical to responsible deployment, as simply retraining VLMs from scratch is prohibitive [42].

Among machine unlearning approaches [23, 25, 4], so-called task vector arithmetic [29] is a very effective plug-and-play strategy. Task vector is defined as the difference of parameters of the model fine-tuned on the target data (*e.g.*, data to be removed) and the original model. The simplicity of task vector arithmetic makes it suitable for large-scale VLMs. Subtracting a task vector from the original model's parameters enables its efficient unlearning without retraining. It reduces the model's performance on a target dataset (*forget set*) and preserves performance on the remainder of the original data. However, the unlearning performance inherently depends on the quality of task vectors, which differ between fine-tuning runs, leading to prediction-level uncertainty due to variance that is unaccounted for in task vector arithmetic.

*Corresponding authors. PK also in charge of theory & derivations.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

To support our claim, Figure 1 reveals the relationship between the average unlearning accuracy on the forget set and the corresponding average prediction-level variance across 8 datasets, computed using function-level ensembles (VLMs with classifier) with varying numbers of task vectors Q. Specifically, we obtain an ensemble $\boldsymbol{\mu}(\boldsymbol{x}) = \frac{1}{Q}\sum_{i=1}^{Q} f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)$ (see [8]) with variance $\boldsymbol{\sigma}^2(\boldsymbol{x}) = -\boldsymbol{\mu}^2(\boldsymbol{x}) + \frac{1}{Q}\sum_{i=1}^{Q} f^2(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)$ where $f(\cdot\,; \boldsymbol{\theta}_0) : \mathbb{R}^{\texttt{in\_size}} \to [0,1]^C$ is a VLM with original training parameters $\boldsymbol{\theta}_0 \in \mathbb{R}^{\texttt{par\_size}}$ and a SoftMax classifier with C classes. Let $\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_Q$ be Q task vectors obtained by fine-tuning $f(\cdot, \boldsymbol{\theta}_0)$ Q times under different augmentations on the forget set (protocol of [29]), which transforms $\boldsymbol{\theta}_0$ into $\boldsymbol{\theta}_i$ for $i = 1, \ldots, Q$. Task vectors, defined as $\boldsymbol{\tau}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_0$, are constructed by fine-tuning on 8 datasets, and evaluated on the function ensemble by averaging the forget accuracy and variance over their test sets.

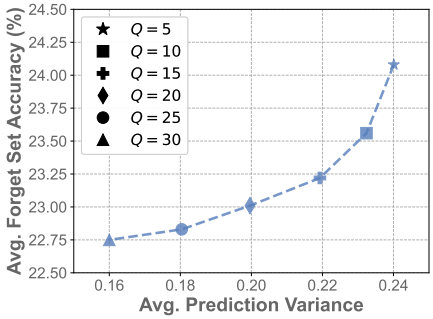

Figure 1: Accuracy on forget set *vs*. class variance w.r.t. the number of task vectors Q used by Q prediction functions that we aggregate. We varied λ (defined in text) to sustain the retain set accuracy of 60.4%±0.3 as per protocol [29].

In detail, we fine-tuned/tested on forget datasets (Stanford Cars, DTD, EuroSAT, GTSRB, MNIST, RESISC45, SUN397, SVHN). Retain accuracy is obtained on ImageNet. See Section 4.1 for details.

Our findings indicate a strong negative correlation: increased prediction variance corresponds to poorer unlearning (higher forget set accuracy indicates worse unlearning). Conversely, aggregating additional task vectors into the function-level ensemble consistently reduces prediction variance and improves unlearning efficacy. This finding is supported by the Bienaymé formula under the average correlation of distinct variables $\rho \geq 0$ and a shared variance $\boldsymbol{\sigma}^2$, defined as $\bar{\boldsymbol{\sigma}}^2 = \frac{\boldsymbol{\sigma}^2}{Q} + \left(1 - \frac{1}{Q}\right)\rho\boldsymbol{\sigma}^2$, which dictates that $\bar{\boldsymbol{\sigma}}^2$ decreases as Q grows ifor sufficiently small ρ. Thus, we note that aggregating multiple functions of different task vectors can lower the prediction variance and improve unlearning.

However, increasing the number of aggregated functions (one per task vector) is costly, as obtaining each task vector requires fine-tuning. Thus, we propose to replace individual task vector arithmetic with the task $(Q-1)$-simplex whose vertices are Q task vectors. Averaging task vectors realizes a multi-task setting [29]. Thus, uniformly sampling a simplex can provide an infinite number of interpolations between task vectors (*i.e*., interpolation between $1, 2, \ldots, Q$ task vectors), realizing an infinite number of tasks encapsulated by the task simplex. However, aggregating an infinite number of functions by naively sampling the task simplex is prohibitive. Instead, we leverage a Taylor expansion and derive a computationally efficient closed-form expectation over an infinite number of functions realized by an infinite uniform sampling of such a task simplex. We show that our model leads to the celebrated bias-variance trade-off [33] and we provide the means of penalizing the variance by learning the importance of vertices or improving their location within a small ℓ_2 ball of ϵ radius.

Our task simplex unlearning outperforms state-of-the-art approaches based on task vector unlearning across 8 datasets. We also show that with task simplex, one can devise a simple incremental unlearning, allowing gradual removal of data. Our main contributions are summarized as follows:

i. We analyze unlearning effectiveness *vs*. function-level (VLM with a classifier) variance in an ensemble of functions of task vectors. As the Bienaymé formula dictates, increasing the number of functions (and thus task vectors) decreases the variance and improves unlearning performance.

ii. Motivated by the additive task vector arithmetic, we propose a $(Q-1)$-task simplex whose vertices are task vectors. Sampling from such a simplex can yield an interpolated task vector between any number of task vector vertices, facilitating multi-task learning.

iii. As aggregating a large number of functions of task vectors is computationally prohibitive, we propose a closed-form ensemble of the infinite number of functions of interpolated task vectors sampled uniformly from the task simplex. To this end, we use 1) Taylor expansion, 2) Jacobian and Hessian of the function evaluated at $\boldsymbol{\theta}_0$, and 3) the expected value over outer products of interpolated vectors sampled from the simplex, obtained from properties of Dirichlet distribution.

iv. We show our closed-form ensemble leads to the bias-variance trade-off by enabling penalizing variance. We also replace the mean aggregation of an infinite number of functions with the probability of at least one success of each class in the infinite trial. Finally, we show that we can distill unlearned parameters from the ensemble (if the final model follows the original VLM).

2 Related Works

Machine Unlearning for VLMs facilitates selective removal of specific knowledge or data (*forget set*) while preserving knowledge of remaining data, and helps address privacy protection and regulatory compliance [2]. This privacy challenge is especially pronounced in VLMs, given their superior generalization due to extensive pre-trained data [42]. Recent efforts in unlearning for foundation VLM include: (i) incremental fine-tuning of VLMs on datasets from which the target knowledge is removed [38, 61], (ii) gradient ascent at the parameter level to erase target knowledge [63, 57], and (iii) pruning model neurons associated with the target knowledge [43]. Arithmetic on *task vector* [29], defined as parameter differences between models fine-tuned with and without specific data, enables efficient and scalable unlearning. Ortiz-Jimenez *et al*. [50] further addressed weight disentanglement within task vectors via model linearization in the tangent space. However, utilizing individual/few task vectors neglects the models' variability due to diverse fine-tuning. Thus, we propose an efficient ensemble of an infinite number of functions of task vectors captured by the task simplex.

Parameter-level Model Merging can enhance performance by combining parameters from multiple models [64, 32, 22]. A simple strategy involves parameter averaging across fine-tuned models, improving task-specific performance without extra computational overhead during inference [56]. Apart from averaging, refined merging of coefficients can be obtained from Fisher information matrices [46] or linear regression [31]. Yadav *et al*. [59] explored interference among model parameters when merging, addressing it by pruning low-magnitude parameter modifications. Yang *et al*. [60] proposed to adaptively learn merging coefficients in an unsupervised scheme. Related are also Parameter-efficient Fine-tuning (PEFT) models, *e.g*., LoRA [27], PACE [48], CrossSpectra [66] and BiLoRA [67] whose fine-tuned parameter residuals are task vectors. Distilled/adapted parameters can be also merged [40, 11]. We focus on function-level ensembles rather than parameter-level merging.

Function-level Ensemble Learning aggregates the function outputs of diverse models, and improves both accuracy and calibration [8, 62]. Simple averaging at the prediction (function) level has been extended to deep learning to enhance generalization across diverse domains under distribution shift [36, 51]. Gontijo-Lope *et al*. [24] empirically showed that diverse training configurations induce distinct generalization behaviors characterized by uncorrelated errors, thereby improving ensemble performance. Rodriguez-Opazo *et al*. [49] investigated cross-backbone ensemble learning in VLMs based on re-weighting predicted logits. In contrast, we focus on VLM unlearning and lowering the prediction variance by the ensemble of infinite number of functions from the task simplex.

3 Proposed Method

Below, we provide background and introduce a closed-form aggregation of the infinite number of functions (VLMs with classifier) formed from task vectors within the task simplex. We also propose how to distill such an ensemble into one task vector (VLM with no ensemble).

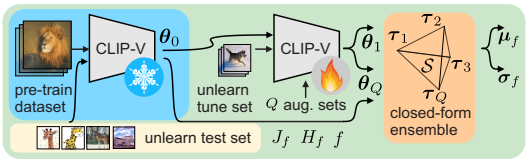

Figure 2: The task simplex based unlearning. Parameters $\boldsymbol{\theta}_0$ are given from pre-trained model. For Q augmentation ways, we get $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_Q$ on the unlearning set and task vectors $\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_Q$ which form slimplex $\mathcal{S}$ for a function-level ensemble.

Background. Foundation VLMs, such as CLIP [52] and its recent variants [39, 41] are pre-trained on extensive datasets collected from diverse internet sources. Formally, a VLM with a classifier can be represented as a function $f: \mathcal{X} \times \theta \rightarrow [0,1]^C$, parameterized by a set of parameters $\boldsymbol{\theta}_0 \in \theta \equiv \mathbb{R}^m$, which maps an input $\boldsymbol{x} \in \mathcal{X}$ to predicted probabilities across C categories. To adapt a pre-trained VLM to a specific downstream task i using a single *task vector* [29], characterized by its dataset $\mathcal{D}_i$ (*e.g*., ImageNet), the pre-trained parameters $\boldsymbol{\theta}_0$ are fine-tuned, resulting in task-specific parameters $\boldsymbol{\theta}_i$, $i > 0$. Following fine-tuning, the associated task vector is defined as the parameter difference between the fine-tuned parameters and the original pre-trained parameters: $\boldsymbol{\tau}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_0$. Depending on the adaptation type, i may refer to a specific dataset or the i-th aug. variant of data.

3.1 Problem Formulation

Figure 1 shows that aggregating more task vectors of different fine-tuning configurations on the unlearning set can lower the prediction variance and enhance unlearning performance. We thus propose to capture the space of all possible interpolations of Q task vectors by a $(Q-1)$-*task simplex* whose vertices are given by $(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_Q)$. Figure 2 is an overview of our method, detailed below.

Problem definition. We investigate machine unlearning in the context of VLMs, aiming to remove knowledge of a so-called *forget set*, while preserving performance across the rest of the data *i.e.*, the *retain set*. Leveraging task vector arithmetic, unlearning can be efficiently achieved by performing task vector negation. Subtracting the scaled task vector from the original parameters (pre-training), *i.e.*, $\boldsymbol{\theta}_i^- = \boldsymbol{\theta}_0 - \lambda\,\boldsymbol{\tau}_i$, where λ is a scaling factor obtained on a held-out validation set to ensure the model's accuracy reduction on the forget set while maintaining accuracy on the retain set. However, given sample $\boldsymbol{x} \in \mathcal{X}$, we are interested in an ensemble of the infinite number of functions (VLM with a classifier) of task vectors uniformly sampled from $(Q-1)$-simplex $\Delta_{\boldsymbol{\theta}}$:

$$\boldsymbol{\mu}_f(\boldsymbol{x}) = \frac{1}{|\Delta_{\boldsymbol{\theta}}|}\int_{\Delta_{\boldsymbol{\theta}}} f(\boldsymbol{x}, \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau})\,d\boldsymbol{\tau} \;\in\; [0,1]^C. \tag{1}$$

Task vectors have large dimension, *i.e.*, $m = 86M$ dim. (CLIP ViT-B-16). Given low number of vectors, *e.g.*, $Q = 30 \ll m = 86M$), due to the curse of dimensionality there exist no obvious way to reliably estimate what distribution type the task vectors follow. However, we choose to model Eq. (1) with the Dirichlet distribution as even for very large m, it facilitates: (i) ease to define its Probability Density Function (PDF) with task vectors as its support parameters, (ii) compact support which does not "exceed" observed minimum or maximum values of individual parameter coefficients, (iii) efficient sampling through our Taylor formulation based on the closed-form 0th, 1st, and 2nd order moments.

Closed-form Aggregation of Functions from Task Simplex. To facilitate efficient aggregation in Eq. (1), we employ a second-order Taylor expansion of the VLM function $f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau})$ around the original parameters $\boldsymbol{\theta}_0$. Let $J_f(\boldsymbol{\theta}_0) \in \mathbb{R}^{C\times m}$ denote the Jacobian matrix of $f(\boldsymbol{x}; \boldsymbol{\theta}_0)$ for a given input $\boldsymbol{x}$. Let class-specific Hessian matrices be $\big(H_f(\boldsymbol{\theta}_0)\big)_1, \ldots, \big(H_f(\boldsymbol{\theta}_0)\big)_C \in \mathbb{R}^{m\times m}$. For sufficiently small scaling factors λ constraining $\|\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}\|_2$ to a small $\epsilon' > 0$ ball, we have the following second-order approximation: $f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) = f(\boldsymbol{x}; \boldsymbol{\theta}_0) - \lambda\, J_f(\boldsymbol{\theta}_0)\,\boldsymbol{\tau} + \frac{1}{2}\left[\lambda^2\,\boldsymbol{\tau}^\top H_f(\boldsymbol{\theta}_0)\boldsymbol{\tau}\right]_{c=1}^{C} + \mathcal{O}\big(\lambda^3\|\boldsymbol{\theta_0}\|^3\big)$. Thus, for a task simplex $\Delta_{\boldsymbol{\theta}}$, the aggregated function-f ensemble is given as:

$$\boldsymbol{\mu}_f(\boldsymbol{x}) = \underbrace{\frac{1}{|\Delta_{\boldsymbol{\theta}}|}\sum_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}_0}} f(\boldsymbol{x}; \boldsymbol{\theta} - \lambda\boldsymbol{\tau})}_{\frac{1}{|\Delta_{\boldsymbol{\theta}}|}\int_{\Delta_{\boldsymbol{\theta}}} d\boldsymbol{\tau} \;\text{ if uncountable set } \Delta_{\boldsymbol{\theta}}} \approx f(\boldsymbol{x}; \boldsymbol{\theta}_0) - \lambda\Big\langle J_f(\boldsymbol{\theta}_0), \underbrace{\frac{1}{|\Delta_{\boldsymbol{\theta}}|}\sum_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\boldsymbol{\tau}}_{\boldsymbol{\mu}_\tau}\Big\rangle + \frac{1}{2}\sum_{c=1}^{C}\lambda^2\underbrace{\Big\langle H_f(\boldsymbol{\theta}_0)_c, \frac{1}{|\Delta_{\boldsymbol{\theta}}|}\sum_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\boldsymbol{\tau}\boldsymbol{\tau}^\top\Big\rangle}_{\omega_c(\boldsymbol{\theta}_0;\Delta_{\boldsymbol{\theta}})}. \tag{2}$$

After simplifying the right-hand part of Eq. (2), $\omega_c(\boldsymbol{\theta}_0; \Delta_{\boldsymbol{\theta}}) = \frac{1}{|\Delta_{\boldsymbol{\theta}}|}\sum_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\boldsymbol{\tau}^\top H_f(\boldsymbol{\theta}_0)_c\boldsymbol{\tau}$. However, ω in Eq. (2) contains a sum over task vector samples from $\Delta_{\boldsymbol{\theta}}$. Thus, below we derive a closed-form ω.

Theorem 1. *Let $\mathcal{S} \subset \mathbb{R}^m$ be a $(Q-1)$-simplex with vertices $\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_Q \in \mathbb{R}^m$. Let matrix $\boldsymbol{H} \in \mathbb{R}^{m\times m}$ (in our case, a Hessian matrix). Consider an i.i.d. uniformly drawing an infinite number of vectors $\boldsymbol{\tau}$ contained in $\mathcal{S}$. Compute an expected value $\mathbb{E}[\boldsymbol{\tau}^\top\boldsymbol{H}\boldsymbol{\tau}]$ which leads to a closed-form expression*[2]*:*

$$\omega = \frac{1}{Q(Q+1)}\Big(\sum_{i=1}^{Q}\boldsymbol{\tau}_i^\top\boldsymbol{H}\boldsymbol{\tau} + \Big(\sum_{i=1}^{Q}\boldsymbol{\tau}_i\Big)^\top\boldsymbol{H}\Big(\sum_{i=1}^{Q}\boldsymbol{\tau}_i\Big)\Big). \tag{3}$$

Proof. Every vector $\boldsymbol{\tau}$ enjoys barycentric coordinates $\boldsymbol{\tau} = \sum_{i=1}^{Q}\alpha_i\,\boldsymbol{\tau}_i$ with $\alpha_i \geq 0$ and $\sum_{i=1}^{Q}\alpha_i = 1$. Under the uniform measure on $\mathcal{S}$, the vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_Q)$ is distributed as $Dir(\mathbf{1}_Q)$. Thus:

$$\mathbb{E}[\alpha_i] = \tfrac{1}{Q}, \qquad \mathbb{E}[\alpha_i^2] = \tfrac{2}{Q(Q+1)}, \qquad \mathbb{E}[\alpha_i\alpha_j] = \tfrac{1}{Q(Q+1)}\ (\text{iff } i \neq j). \tag{4}$$

Now we expand $\boldsymbol{\tau}^\top\boldsymbol{H}\boldsymbol{\tau} = \sum_{i=1}^{Q}\sum_{j=1}^{Q}\alpha_i\alpha_j\,\boldsymbol{\tau}_i^\top\boldsymbol{H}\boldsymbol{\tau}_j$, substitute Eq. (4) and compute expectations:

$$\mathbb{E}[\boldsymbol{\tau}^\top\boldsymbol{H}\boldsymbol{\tau}] = \sum_{i=1}^{Q}\mathbb{E}[\alpha_i^2]\,\boldsymbol{\tau}_i^\top\boldsymbol{H}\boldsymbol{\tau}_i + \sum_{\substack{i,j=1\\ i\neq j}}^{Q}\mathbb{E}[\alpha_i\alpha_j]\,\boldsymbol{\tau}_i^\top\boldsymbol{H}\boldsymbol{\tau}_j = \tfrac{2}{Q(Q+1)}\sum_{i=1}^{Q}\boldsymbol{\tau}_i^\top\boldsymbol{H}\boldsymbol{\tau}_i + \tfrac{1}{Q(Q+1)}\sum_{\substack{i,j=1\\ i\neq j}}^{Q}\boldsymbol{\tau}_i^\top\boldsymbol{H}\boldsymbol{\tau}_j. \tag{5}$$

□

[2] See also an expectation over the outer product of vectors on $\mathcal{S}$ [14].

Corollary 1. *Let vertices be assigned importance weights,* i.e., $\big(w_1,\ldots,w_Q:\sum_{i=1}^Q w_i=1\big)$. *Let* w *be a linear interpolation of these weights at location* $\boldsymbol{\tau}$. *Extending Theorem 1 to* $\mathbb{E}[w\boldsymbol{\tau}^\top \boldsymbol{H}\boldsymbol{\tau}]$ *yields:*

$$\omega' = \frac{1}{Q(Q+1)(Q+2)}\left(\sum_{i=1}^{Q}(2+4\,w_i)\,\boldsymbol{\tau}_i^\top \boldsymbol{H}\boldsymbol{\tau}_i + \sum_{\substack{i,j=1\\ i\neq j}}^{Q}\big(1+w_i+w_j\big)\,\boldsymbol{\tau}_i^\top \boldsymbol{H}\boldsymbol{\tau}_j\right). \tag{6}$$

Proof. See Appendix G.1. □

Theorem 1 and Corollary 1 provide closed-form ω for Eq. (2) under uniform and importance weighting of task vectors, respectively. Weights of importance weighting (or task vectors themselves) may be optimized to lower the variance associated with Eq. (2).

Computing Variance. Using Eq. (2), the variance of ensemble of the infinite number of functions is given as:

$$\boldsymbol{\sigma}^2(\boldsymbol{x}) = \boldsymbol{\mu}_{f^2}(\boldsymbol{x}) - \big(\boldsymbol{\mu}_f(\boldsymbol{x})\big)^2. \tag{7}$$

Advanced Aggregation Scheme. Up to this point, we discussed the average pooling aggregator in Eq. (1) and (2). Another popular aggregator is maximum pooling over a set of votes (outputs of classifier-level functions) [34]. However, the maximum operation cannot be applied over an infinite number of voters. Therefore, we leverage the probability mass function (PMF) of the Poisson Binomial distribution. Firstly, we ask a question: *what is the probability of zero successes in* Q *trials?* This question is asked under independent Bernoulli trials that are not necessarily identically distributed, *i.e.*, we treat an output of each function in the ensemble as one probability of success, denoted p_i^c. For Q trials and class c, we readily obtain $\prod_{i=1}^Q(1-p_i^c)$ and we readily deduce that $1-\prod_{i=1}^Q(1-p_i^c) \geq \max_{i=1,\ldots,Q} p_i^c$ is *the probability of at least one successes in* Q *trials* and an upper bound of maximum pooling. Now it remains to extend this result to an ensemble of the infinite number of functions from the task simplex.

Theorem 2. *Consider task vector* $\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}$ *sampled from task simplex* $\Delta_{\boldsymbol{\theta}}$. *Let* $g(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau}) = \log\big(1-f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})\big)$, *where* $g:\mathbb{R}^m\to(-\infty,0]^C$ *converts the prediction into the log space. Notice that* $|\Delta_{\boldsymbol{\theta}}|\boldsymbol{\mu}_g(\boldsymbol{x}) = \sum_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}} g(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})$ *and thus we have:*

$$\underbrace{\boldsymbol{\phi}_f(\boldsymbol{x})}_{\in[0,1]^C} = 1-\exp\big(|\Delta_{\boldsymbol{\theta}}|\boldsymbol{\mu}_g(\boldsymbol{x})\big) = 1-\prod_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\big(1-f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})\big) \geq \max_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}} f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau}), \tag{8}$$

which is the probability of at least one successes in the infinite number of trials, each trial being an output of function $f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})$ for task vector $\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}$. Volume $|\Delta_{\boldsymbol{\theta}}|$ is defined in Theorem 3.

Proof. See Appendix G.2. □

Theorem 2 proposes an aggregator which returns a high likelihood if (i) many aggregated functions yielded at least weak but consistent belief in class activation across many functions or (ii) at least one aggregated function yielded a very strong belief in class activation. In contrast, the average pooling aggregator may not yield strong response if many voters yield consistent but relatively weak class activations.

Theorem 2 requires evaluation of volume of the task simplex $|\Delta_{\boldsymbol{\theta}}|$ which provided below.

Theorem 3. *Consider* $(Q-1)$*-simplex* $\Delta_{\boldsymbol{\theta}}$ *with vertex list* $(\boldsymbol{\tau}_1,\boldsymbol{\tau}_2,\ldots,\boldsymbol{\tau}_Q)$ *and construct a vertex list matrix offset by* $\boldsymbol{\tau}_1$, i.e., $\boldsymbol{V}=[\boldsymbol{\tau}_2-\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_Q-\boldsymbol{\tau}_1]\in\mathbb{R}^{m\times(Q-1)}$. *Then the volume of the simplex* $\Delta_{\boldsymbol{\theta}}$ *is given as:*

$$|\Delta_{\boldsymbol{\theta}}| = \frac{\sqrt{\det \boldsymbol{V}^\top\boldsymbol{V}}}{(Q-1)!}. \tag{9}$$

Proof. See Appendix G.3. □

Distillation from Ensemble. Our aggregation of the infinite number of functions requires to be implemented on top of an VLM. However, if deployment requires a VLM without ensemble, one can distill the task vector from the ensemble as follows:

$$\begin{aligned}\underset{\substack{\boldsymbol{\tau}^-\\ \text{(opt.)}\boldsymbol{w}\text{ s.t. }\|\boldsymbol{w}\|_2=1\\ \Delta\boldsymbol{\tau}_i\text{ s.t. }\|\Delta\boldsymbol{\tau}_i\|_2\leq\epsilon,\forall i}}{\arg\min}\ &\frac{1}{|\mathcal{X}'|}\sum_{\boldsymbol{x}\in\mathcal{X}'}\Big\|f(\boldsymbol{x},\boldsymbol{\theta}_0-\boldsymbol{\tau}^-) - \boldsymbol{\mu}_f^{\texttt{stop\_grad}}\big(\boldsymbol{x};\boldsymbol{w},(\boldsymbol{\tau}_1+\Delta\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_Q+\Delta\boldsymbol{\tau}_Q)\big)\Big\|_2^2\\ &+\beta\Big\|\boldsymbol{\sigma}^2\big(\boldsymbol{x};\boldsymbol{w},(\boldsymbol{\tau}_1+\Delta\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_Q+\Delta\boldsymbol{\tau}_Q)\big)\Big\|_1,\end{aligned} \tag{10}$$

where $\mathcal{X}'$ is a distillation set used without labels, *i.e.* the training split of unlearning set, $\beta \geq 0$ controls the variance (section below explains importance of bias-variance trade-off in controlling the model complexity), `stop_grad` means we stop the gradient through computations of $\boldsymbol{\mu}_f$ and $\boldsymbol{\tau}^-$ is a single distilled task vector capturing the ensemble. Moreover, "opt." indicates one can optionally optimize over importance weights $\boldsymbol{w}$ and/or small perturbations $\Delta\boldsymbol{\tau}_i$ of vertices within ϵ radius. Notice that we slightly abuse notation of $\boldsymbol{\mu}_f$ and $\boldsymbol{\sigma}_f^2$ by passing parameters $\boldsymbol{w}$ and $\Delta\boldsymbol{\tau}_i$ into them.

Controlling Variance of Ensemble. In order to control the complexity of the unlearned model, one can reduce variance of the model in the so-called bias-variance trade-off [33]. For our ensemble, reducing variance may be achieved as follows:

$$\underset{\substack{\boldsymbol{w} \text{ s.t. } \|\boldsymbol{w}\|_2=1 \text{ and/or} \\ \Delta\boldsymbol{\tau}_i \text{ s.t. } \|\Delta\boldsymbol{\tau}_i\|_2 \leq \epsilon, \forall i}}{\arg\min} \frac{1}{|\mathcal{X}'|} \sum_{\boldsymbol{x} \in \mathcal{X}'} \Big\| \boldsymbol{\sigma}^2\big(\boldsymbol{x}; \boldsymbol{w}, (\boldsymbol{\tau}_1 + \Delta\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_Q + \Delta\boldsymbol{\tau}_Q)\big) \Big\|_1, \tag{11}$$

where optimizing requires early stopping to achieve a desired variance reduction. Notice, Mean Square Loss (MSE) over $\boldsymbol{x} \in \mathcal{X}'$ between labels $y(\boldsymbol{x})$ and the learner $f(\boldsymbol{x})$, given as $\mathbb{E}_{\mathcal{D}_0, \boldsymbol{\tau}}\Big[\big\|y(\boldsymbol{x}) - f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau})\big\|_2^2\Big]$ can readily be decomposed into the bias and variance terms:

$$\mathbb{E}_{\mathcal{D}_0, \boldsymbol{\tau}}\Big[\underbrace{\|f^*(\boldsymbol{x}) - \boldsymbol{\mu}_f(\boldsymbol{x})\|_2^2}_{(\text{Bias}_{\boldsymbol{\tau}}\, f(\boldsymbol{x}))^2} + \underbrace{\|\boldsymbol{\sigma}_f^2(\boldsymbol{x})\|_1}_{\text{Var}_{\boldsymbol{\tau}}(f(\boldsymbol{x}))}\Big] + \boldsymbol{\gamma}^2, \tag{12}$$

where $f^*(\cdot)$ is some class function we want to approximate, $y(\boldsymbol{x}) = f^*(\boldsymbol{x}) + \boldsymbol{\eta}$ where $\boldsymbol{\eta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\gamma}^2)$ is noise with variance $\boldsymbol{\gamma}^2$ a.k.a. the so-called irreducible error. See Appendix G.4 for further details.

Bias-variance trade-off. Eq. (12) shows that that our ensemble can be interpreted as bias-variance trade-off [33] under different choices of the unlearning task vectors $\boldsymbol{\tau}$. While we cannot reduce the bias, we can adjust the variance to improve generalization on the underlying unlearning set. Belkin *et al.* [1] showed that below the so-called interpolation threshold, high bias and low variance indicate low model complexity, whereas low bias and high variance indicate high model complexity. Above the interpolation threshold, lowering the variance can further increase the function complexity. In either case, variance is key in controlling the ensemble complexity.

Theoretical Analysis. Below we provide additional theoretical analysis of properties of our ensemble.

Theorem 4 (Interpolation Bound)**.** *Let $g : \mathbb{R}^m \to \mathbb{R}^{C'}$ be an unlearning function. Let $g(\cdot)$ be L-Lipschitz continuous,* i.e.*, $\|g(\boldsymbol{\theta}_i) - g(\boldsymbol{\theta}_j)\|_1 \leq L\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|_2,\ \forall i \neq j$.*

Given task vectors $(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_Q)$, diameter $\eta = \max_{i \neq j} \|\boldsymbol{\tau}_i - \boldsymbol{\tau}_j\|_2$, and any convex combination $\boldsymbol{\tau} = \sum_{i=1}^Q \alpha_i \boldsymbol{\tau}_i$ where $\sum_{i=1}^Q \alpha_i = 1, \alpha_i \geq 0$, we obtain:

$$\Big\| g(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \sum_{i=1}^Q \alpha_i\, g(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i) \Big\|_1 \leq L\lambda\eta. \tag{13}$$

Proof. See Appendix G.5. □

This bound is small if (i) task vectors differ by a small diameter η, λ is small and $g(\cdot)$ is smooth with low Lipschitz constant L. Indeed, these conditions are met, *e.g.*, we have low η due to sparse change between task vectors as explained in Appendix H.2, and $\lambda \leq 1$. Theorem 4 says that any interpolated choice of $(\alpha_1, \ldots, \alpha_Q)$ deviates from the expected convex combination by at most $L\lambda\eta$.

Next, below we compare Taylor expansions of (i) Vector Uniform Merge (Uniform Soup) [56], (ii) Function Ensemble [56, 8] and (iii) our infinite number of functions drawn from the task simplex:

$$\text{Vector Uniform Merge:}\quad f\Big(\mathbf{x}; \boldsymbol{\theta}_0 - \tfrac{\lambda}{Q}\sum_{i=1}^Q \boldsymbol{\tau}_i\Big) \approx \underbrace{f(\mathbf{x}; \boldsymbol{\theta}_0) - \lambda\langle J_f(\boldsymbol{\theta}_0), \boldsymbol{\mu}_\tau\rangle}_{linear} + \underbrace{\lambda^2\, \boldsymbol{\mu}_\tau^\top H_f(\boldsymbol{\theta}_0)\boldsymbol{\mu}_\tau}_{quadratic\_VUM}. \tag{14}$$

$$\text{Function Ensemble:}\quad \tfrac{1}{Q}\sum_{i=1}^Q f(\mathbf{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i) \approx \ linear\ + \underbrace{\lambda^2 \tfrac{1}{Q}\sum_{i=1}^Q \boldsymbol{\tau}_i^\top H_f(\boldsymbol{\theta}_0)\boldsymbol{\tau}_i}_{quadratic\_FE}. \tag{15}$$

$$\text{Ours:}\quad \tfrac{1}{|\Delta_{\boldsymbol{\theta}}|}\int_{\Delta_{\boldsymbol{\theta}}} f(\boldsymbol{x}; \boldsymbol{\theta} - \lambda\boldsymbol{\tau})\, d\boldsymbol{\tau} \approx \ linear\ + \underbrace{\rho\ quadratic\_VUM + \zeta\ quadratic\_FE}_{\text{Interpolation of } \textit{Vector Uniform Merge} \text{ and } \textit{Func. Ensemble}}. \tag{16}$$

Table 1: **Standard unlearning** of CLIP models measured by the average accuracy (%) on the forget datasets (Stanford Cars, DTD, EuroSAT, GTSRB, MNIST, RESISC45, SUN397, and SVHN) and the accuracy (%) on the retain dataset (ImageNet) across diverse CLIP architectures. Note that the retain set accuracy is required by the protocol to be fixed around 95% of pre-trained model by cross-val. λ.

Method	ViT-Base/32		ViT-Base/16		ViT-Large/14	
	Forget (↓)	Retain (↑)	Forget (↓)	Retain (↑)	Forget (↓)	Retain (↑)
Pre-trained Model	48.09	63.33	55.46	68.33	65.22	75.53
Standard Task Arithmetic [29]	24.23	60.74	20.57	64.77	16.66	72.10
Best Model on Val. Set [56]	22.69	60.54	19.86	64.43	16.16	71.64
Vector Uniform Merge [56]	23.01	60.64	20.28	64.66	17.30	72.08
Vector Greedy Merge [56]	22.14	60.46	19.28	64.51	16.77	71.67
Vector TIES-Merging [59]	23.94	60.27	20.19	64.56	17.83	72.50
Vector EMR-Merging [28]	21.83	60.34	19.10	64.52	15.67	71.89
Function Ensemble [56, 8]	22.75	60.32	19.89	64.48	16.32	71.91
Ours	**15.20**	60.58	**12.17**	**64.93**	**9.98**	72.17
Ours (Distillation)	15.66	**60.79**	12.70	64.72	10.63	**72.59**

The 0^{th} and 1^{st} order terms of Taylor expansions are identical for Vector Uniform Merge, Function Ensemble and our approach, and are related to the Neural Tangent Kernel of $f(\cdot)$.

However, Eq. (14) and (15) differ in their second-order terms of Taylor expansions. Eq. (16) shows that our approach in fact interpolates between quadratic terms of Vector Uniform Merge and Function Ensemble. For arbitrary concentration parameter α of the Dirichlet distribution, we obtain an interpolation with $\rho = \frac{\alpha Q}{\alpha Q+1}$ and $\zeta = \frac{1}{\alpha Q+1}$. For $\alpha = 1$, $\rho = \frac{Q}{Q+1}$ and $\zeta = \frac{1}{Q+1}$.

4 Experiments

4.1 Experimental Setup

Datasets. Following prior works [29, 50], we adopt the identical experimental setup. The unlearning evaluation of CLIP is performed on eight datasets designated as the forget set. To assess performance of retained knowledge, we use ImageNet [7] as the retain set. Further details are in Appendix A.1.

Compared Methods & Evaluations. In addition to standard task vector-based unlearning [29] for VLMs, we also adapt the notion of task vectors to extend existing model merging approaches [54] for unlearning. Specifically, we reinterpret Uniform Merge [56], Greedy Merge [56], TIES-Merging [59], and EMR-Merging [28] as parameter-level task vector merging strategies for unlearning comparisons. As our proposed method emphasizes function-level ensemble unlearning, we also consider the vanilla function-level ensemble as a baseline. Detailed formulations of these methods within the context of vision-language model (VLM) unlearning are provided in Appendix A.2. Unlearning is assessed on both the forget set and the retain set, where lower accuracy on the forget set indicates better unlearning, and accuracy on the retain set is mandated by the protocol to reach a fixed retain accuracy.

Implementation Details. Unless otherwise stated, we adopt the standard CLIP fine-tuning protocol for generating the task vectors on each forget dataset, in line with prior task arithmetic studies [29, 50]. Specifically, we adopt the AdamW optimizer [44] with a peak learning rate of 1×10^{-5}, momentum $(0.9, 0.999)$ along with a cosine annealing scheduler and a weight decay of 0.1. During fine-tuning, the CLIP text encoder is frozen to retain the integrity of the pre-trained textual representations for the classification head via zero-shot prompt embeddings. To ensure diversity of task vectors, we generate a pool of 30 fine-tuned CLIP models by varying the data augmentation configurations, *e.g.*, using different hyper-parameters of RandAugment [6]. All VLM unlearning experiments are conducted across diverse CLIP backbones [52], including ViT-Base/32, ViT-Base/16, and ViT-Large/14. For each unlearning scenario, the task vector negation coefficient λ is selected on a small held-out subset of the training data. Unless otherwise specified, the task simplex is constructed using 30 vertices, and the distillation variance penalty is set to $\beta = 2.0$. More details are in Appendix A.3.

4.2 Main Results

Unlearning with Task Vectors. Below, we compare our proposed method and its distilled single task vector variant, against prior task vector based unlearning approaches across three CLIP backbones:

Table 2: **Linear task vector unlearning** of CLIP models measured by the average accuracy (%) on the forget datasets (Stanford Cars, DTD, EuroSAT, GTSRB, MNIST, RESISC45, SUN397, SVHN) and the accuracy (%) on the retain dataset (ImageNet) across diverse CLIP architectures. The protocol mandates the retain set accuracy to be fixed around 95% of pre-trained model by cross-val. λ.

Method	ViT-Base/32		ViT-Base/16		ViT-Large/14	
	Forget (↓)	Retain (↑)	Forget (↓)	Retain (↑)	Forget (↓)	Retain (↑)
Pre-trained Model	48.09	63.33	55.46	68.33	65.22	75.53
Standard Task Arithmetic [29]	11.40	60.53	8.48	64.87	7.76	72.15
Best Model on Val. Set [56]	10.98	60.46	8.17	**65.10**	7.40	72.26
Vector Uniform Merge [56]	10.62	60.37	7.90	64.68	7.48	72.04
Vector Greedy Merge [56]	10.12	60.42	7.55	64.66	7.14	71.93
Vector TIES-Merging [59]	11.08	60.25	7.82	64.72	7.43	72.28
Vector EMR-Merging [28]	9.53	60.51	7.09	64.35	6.51	72.29
Function Ensemble [56, 8]	9.97	60.40	7.41	64.62	7.02	72.12
Ours	**7.88**	60.43	**6.46**	64.59	**5.70**	72.38
Ours (Distillation)	8.13	**60.83**	6.79	65.08	5.94	**72.56**

Table 3: **Incremental unlearning** measured by sequential-average forget accuracy (%, lower is better) after each step across eight benchmark datasets using CLIP ViT-Base/32.

Method	Cars	+DTD	+EuroSAT	+GTSRB	+MNIST	+RESISC45	+SUN397	+SVHN
Standard Task Arithmetic [29]	34.8	34.4	28.0	24.3	23.8	26.2	31.0	29.2
Best Model on Val. Set [56]	32.0	32.6	26.3	22.9	22.1	24.5	29.4	27.7
Vector Uniform Merge [56]	32.6	33.1	26.6	22.8	22.7	24.3	29.1	28.0
Vector Greedy Merge [56]	32.0	32.5	26.0	22.5	21.4	23.9	28.7	27.1
Vector TIES-Merging [59]	33.6	34.2	27.4	23.8	23.3	25.8	30.7	28.9
Vector EMR-Merging [28]	31.0	31.6	25.4	22.0	21.5	23.9	28.4	26.8
Function Ensemble [56, 8]	32.3	32.8	26.3	22.8	22.3	24.7	29.4	27.7
Ours	**28.7**	**28.3**	**21.0**	**17.3**	**15.2**	**16.4**	**21.5**	**20.1**
Ours (Distillation)	29.2	29.0	21.8	18.3	17.2	17.6	22.8	22.5

ViT-Base/32, ViT-Base/16, and ViT-Large/14. Table 1 lists the average accuracy on both the forget set and the retain set, computed by individually unlearning each of the eight target datasets. One can observe that the classification accuracy on the retain set remains consistently high, preserving ~95% of the original performance of the pre-trained model on all methods. Compared to *function ensemble* our closed-form solution achieves 7.55%, 7.72% and 6.34% improvement in unlearning on ViT-Base/32, ViT-Base/16 and ViT-Large/14, respectively. Our method also outperforms alternatives when scaled to larger-scale CLIP architectures (*e.g.*, ViT-Large/14), effectively removing targeted knowledge while retaining general capabilities. The distilled variant of our method achieves unlearning accuracy similar to that of our function-level ensemble, highlighting benefits of distilling function-level task simplex ensemble into a single task vector for improved portability. Unlearning performance w.r.t. each dataset is presented in Appendix B.

Unlearning with Linear Task Vectors. Model linearization, grounded in the Neural Tangent Kernel (NTK) theory [30], enjoys strong task arithmetic capabilities [50]. Thus, we leverage linearized task vectors in various model merging strategies and evaluate their effectiveness in unlearning tasks. Details of the task vector linearization are provided in Appendix C. Table 2 shows that the linearized task vectors consistently yield substantial reductions in the forget set accuracy across multiple merging methods and CLIP architectures, while maintaining fixed accuracy on the retain set relative to their standard (non-linearized) counterparts in Table 1. Notably, both our method and its distilled variant continue to outperform other baselines, achieving the best unlearning efficacy while preserving most of the retained knowledge on ImageNet.

Incremental Unlearning. While prior literature on unlearning in VLMs focus on single-dataset removal, we propose a more realistic and challenging scenario: unlearning datasets incrementally. This setup lets us assess the cumulative unlearning ability and interferences across distribution shifts from the perspective of task vectors. To this end, we fine-tune CLIP ViT-B/32 on each forget dataset, construct task vectors, and sequentially use them in each method to remove datasets one by one: *Cars* $\rightarrow$ *DTD* $\rightarrow$ *EuroSAT* $\rightarrow \ldots \rightarrow$ *SVHN*. After each removal step, we report the average accuracy on all unlearned (forget) datasets so far. For our distillation, after each step we distill the task vector $\boldsymbol{\tau}^-$ and set $\boldsymbol{\theta}_0 := \theta_0 - \lambda\boldsymbol{\tau}^-$ to accumulate parameters across unlearning steps. Table 3 shows our method enjoys the lowest average forget accuracy across all steps. See Appendix D for more details.

VQA Task.

Below we apply our method on multi-modal CLEAR benchmark [21] (fictional author profiles with face images–captions pairs). Average accuracy (VQA task) with LLaVA-1.5-7B, CLIP ViT-L/14 as vision encoder. Task vectors are derived by fine-tuning on the corresponding target data. Task specificity: forgetting is measured in terms of classification accuracy. This experiment demonstrates that our approach is not limited to classification tasks. Fine-tuning can be performed on image-text pairs, and the CLIP loss or any other loss can be used for fine-tuning to obtain task vectors.

Table 4: CLEAR VQA (multi-modal) results.

Method	VQA Acc. Forget (↓)	VQA Retain (↑)
Pre-trained Model	69.2	55.7
Standard Task Arithmetic	42.7	49.4
Uniform Merge	37.9	50.5
TIES-Merging	36.1	49.7
EMR-Merging	34.8	49.3
Function Ensemble	35.6	50.0
Ours	**31.5**	**50.2**

4.3 Analysis

Below provide further analyses of each component in our closed-form ensemble unlearning.

Ablation Studies. Table 5 compares four key components of our method in unlearning efficacy: (1) Vanilla ensemble of functions from task vectors (first row), (2) Our *Task Simplex* (*Simplex*), (3) *Advanced Aggregation* (Theorem 2) denoted *Adv. Aggregator*, and (4) Variance reduction by *Vertex Importance Weighting* (Corollary 1+Eq. (11)) denoted as *Weighting*, and (5) Variance reduction by *Optimization of Vertices* $\Delta\boldsymbol{\tau}_i$ (Eq. (11)) denoted as *Vertices Opt.* We report both the average forget-set accuracy across 8 datasets and the average retain-set accuracy on ImageNet based on CLIP ViT-Base/32. Table 5 shows that the task simplex significantly enhances the unlearning efficacy over function ensembling of limited task vectors. Advanced aggregation further reduces the forget accuracy while preserving most of the retained knowledge on ImageNet. The unlearning improvement obtained by variance reduction by vertex importance weighting justifies our motivation that suppressing prediction variance helps unlearning performance.

Table 5: Ablation study of key components in our method for average forget-set and retain-set accuracy (%).

	Simplex	Adv. Aggregator	Weighting	Vertices Opt.	Forget (↓)	Retain (↑)
1					22.75	60.32
2	✓				17.89	60.43
3	✓	✓			16.87	60.52
4	✓		✓		16.72	60.45
5	✓	✓		✓	15.67	60.30
6	✓	✓	✓		**15.20**	**60.58**

The Impact of the Number of Task Vector Vertices of the Simplex. Figure 1 shows that ensembling a larger number of functions from task vectors reduces prediction variance, and improves unlearning efficacy. Figure 3 further validates this claim within the framework of the task simplices. We analyze our closed-form simplex aggregation under varying numbers of task vectors (*i.e.*, simplex vertices). Unlike ensembling functions from limited task vectors, our closed-form solution approximates infinite sampling over the simplex, resulting in consistently lower prediction variance and reduced forget-set accuracy. Furthermore, we observe a clear correlation between unlearning performance and prediction-level variance: modeling higher-dimensional task vector simplices by aggregating more task vectors yields enhanced unlearning effectiveness, further validating the robustness of our method.

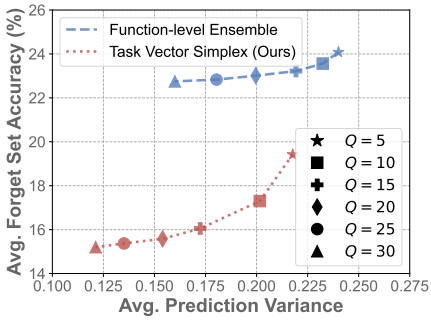

Figure 3: Comparison between function-level ensembles from task vectors and our task simplex w.r.t. the number of task vectors Q.

Importance Weighting Strategies. Motivated by the observed correlation between unlearning efficacy and prediction variance, below we study different importance weighting strategies for simplex vertices with the goal of enhancing unlearning.

In addition to importance weighting mechanism from Corollary 1+Eq. (11), we also investigate a weighting strategy that measures the importance of each task vector vertex based on the Cross-Entropy (CE) loss on the forget dataset (formulated in Appendix E). Table 6 evaluates

Table 6: Average accuracy (%) comparing variants of importance weighting of simplex vertices.

Weighting strategy	Forget (↓)	Retain (↑)
Uniform weighting	16.87	60.52
CE-guided weighting	16.37	60.28
Variance-reducing weighting (**Ours**)	**15.20**	**60.58**

both forget-set and retain-set accuracy and shows that the variance reduction by importance weighting consistently outperforms the CE-guided strategy in terms of the forget-set performance. We attribute this improvement to learning weights that reduce variance of ensemble by limiting the impact of functions associated with noisy task vectors.

Comparison of Distillation Schemes. Our distillation in Eq. (10) distills single task vector from the ensemble to facilitate easier deployment of unlearned-set model. Table 7 studies the effectiveness of different task vector distillation schemes. Specifically, it compares different prediction alignment metrics (*e.g.*, KL Divergence and the ℓ_2 distance) and evaluates their variants with or without variance tuning. Without variance tuning, the use of the ℓ_2 distance already demonstrates improved unlearning efficacy over the KL Div., potentially due to its stronger impact on logits and smoother gradient behavior. Incorporating importance weighting for variance tuning enhances the effectiveness of both metrics by suppressing high-variance predictions during distillation.

Table 7: Average performance (%) of our method w.r.t. different task vector **distillation** schemes.

Prediction Alignment Metric	Variance Tuning	Forget (↓)	Retain (↑)
KL Div.	no	17.28	60.47
ℓ_2 distance	no	16.83	60.52
KL Div.	yes	16.09	60.64
ℓ_2 distance	yes	**15.66**	**60.79**

Concentration Parameter α. The Dirichlet distribution may follow a non-uniform density if the concentration parameter $\alpha \neq 1$. Up to this point, we employed $\alpha = 1$. However, Eq. (16) shows that α controls the level of interpolation between quadratic terms of Vector Greedy Merge and Function Ensemble in our method. Table 8 (CLIP ViT-B/16) shows that the best unlearning performance is achieved for $\alpha = 0.8$. Intuitively, $\alpha < 1$ indicates higher sampling density toward simplex vertices (away from the simplex center) which are task vectors. Thus, the function ensemble puts more emphasis on contributions closer to Q task vectors.

Table 8: Effect of Dirichlet concentration α.

α	Forget (↓)	Retain (↑)
0.1	14.65	64.49
0.5	12.83	64.26
0.8	**12.05**	**64.68**
1.2	12.48	64.85
1.5	12.70	64.92
2.0	13.29	64.81
1.0	**12.17**	**64.93**

Sampling Task Vectors *vs*. closed-form Solution. Our function ensemble enjoys closed-form solution as an efficient alternative to naive task vector sampling from the simplex. Figure 4 compares our closed-form task vector aggregation with explicit task vector sampling for function-level ensembling. As the number of sampled vectors increases, the forget-set accuracy gradually improves but saturates near 300 samples, beyond which additional gains are marginal. However, this improvement comes at the cost of significantly higher unlearning time, up to $4\times$ longer. In contrast, our closed-form solution achieves similar performance with a dramatically lower time cost (*e.g.*, 0.8 hours *vs*. 6.3 hours), offering a scalable and efficient unlearning. The circle radius indicates the number of sampled task vectors used in unlearning, with our method effectively sampling the infinite number of task vector interpolations between vertices of the task simplex.

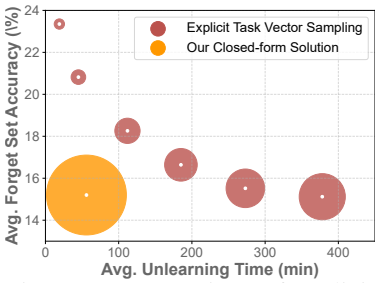

Figure 4: Comparison of explicit task vector sampling *vs*. our closed-form solution. The circle radius denotes the number of sampled task vectors, while ours represents an infinite sampling.

5 Conclusions

We have investigated the function-level ensemble perspective of task vector–based unlearning in VLMs, revealing an inherent connection between unlearning effectiveness and prediction-level variance. Through empirical and theoretical analysis, we have shown that aggregating larger number of functions from several task vectors obtained on unlearning set improves unlearning performance by reducing prediction variance, consistent with the Bienaymé principle. To address the scalability limitations of explicitly generating and ensembling an infinite number of task vectors, we introduce a novel framework based on a high-dimensional task simplex, where each vertex represents a task vector derived from distinct fine-tuning strategy on the unlearning set. We then derive a closed-form ensemble over an infinite number of interpolated task vectors uniformly sampled from this simplex. This formulation enables efficient unlearning via function-level aggregation while capturing the bias-variance trade-off. Our framework supports the distillation of unlearned model parameters from the ensemble, ensuring compatibility with the original VLM when required.

Acknowledgments

Piotr Koniusz and Hao Zhu are supported by CSIRO's Science Digital. This research is supported in part by National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative, partly supported through the the under its AI Centre of Excellence for Manufacturing (AIMfg) (Award W25MCMF014), the Centre for Frontier Artificial Intelligence Research, Institute of High Performance Computing, A*Star, and the College of Computing and Data Science at Nanyang Technological University. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, and Infocomm Media Development Authority.

References

[1] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[2] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, pages 141–159. IEEE, 2021.

[3] 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.

[4] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7210–7217, 2023.

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[6] 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.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[8] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[9] Junhao Dong, Piotr Koniusz, Junxi Chen, and Yew-Soon Ong. Adversarially robust distillation by reducing the student-teacher variance gap. In *European Conference on Computer Vision*, pages 92–111. Springer, 2024.

[10] Junhao Dong, Piotr Koniusz, Junxi Chen, Z Jane Wang, and Yew-Soon Ong. Robust distillation via untargeted and targeted intermediate adversarial samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28432–28442, 2024.

[11] Junhao Dong, Piotr Koniusz, Junxi Chen, Xiaohua Xie, and Yew-Soon Ong. Adversarially robust few-shot learning via parameter co-distillation of similarity and class concept learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28535–28544, June 2024.

[12] Junhao Dong, Piotr Koniusz, Liaoyuan Feng, Yifei Zhang, Hao Zhu, Weiming Liu, Xinghua Qu, and Yew-Soon Ong. Robustifying zero-shot vision language models by subspaces alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21037–21047, October 2025.

[13] Junhao Dong, Piotr Koniusz, Xinghua Qu, and Yew-Soon Ong. Stabilizing modality gap & lowering gradient norms improve zero-shot adversarial robustness of vlms. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, page 236–247, New York, NY, USA, 2025. Association for Computing Machinery.

[14] Junhao Dong, Piotr Koniusz, Yifei Zhang, Hao Zhu, Weiming Liu, Xinghua Qu, and Yew-Soon Ong. Improving zero-shot adversarial robustness in vision-language models by closed-form alignment of adversarial path simplices. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 14061–14078. PMLR, 13–19 Jul 2025.

[15] Junhao Dong, Jiao Liu, Xinghua Qu, and Yew-Soon Ong. Confound from all sides, distill with resilience: Multi-objective adversarial paths to zero-shot robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 624–634, October 2025.

[16] Junhao Dong, Seyed-Mohsen Moosavi-Dezfooli, Jianhuang Lai, and Xiaohua Xie. The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24678–24687, June 2023.

[17] Junhao Dong, Yuan Wang, Jian-Huang Lai, and Xiaohua Xie. Improving adversarially robust few-shot image classification with generalizable representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9025–9034, 2022.

[18] Junhao Dong, Yuan Wang, Xiaohua Xie, Jianhuang Lai, and Yew-Soon Ong. Generalizable and discriminative representations for adversarially robust few-shot learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(3):5480–5493, 2024.

[19] Junhao Dong, Lingxiao Yang, Yuan Wang, Xiaohua Xie, and Jianhuang Lai. Toward intrinsic adversarial robustness through probabilistic training. *IEEE Transactions on Image Processing*, 32:3862–3872, 2023.

[20] Junhao Dong, Cong Zhang, Xinghua Qu, Zejun MA, Piotr Koniusz, and Yew-Soon Ong. Robust superalignment: Weak-to-strong robustness generalization for vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[21] Alexey Dontsov, Dmitrii Korzh, Alexey Zhavoronkin, Boris Mikheev, Denis Bobkov, Aibek Alanov, Oleg Y Rogov, Ivan Oseledets, and Elena Tutubalina. Clear: Character unlearning in textual and visual modalities. *arXiv preprint arXiv:2410.18057*, 2024.

[22] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020.

[23] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

[24] Raphael Gontijo-Lopes, Yann Dauphin, and Ekin Dogus Cubuk. No one representation to rule them all: Overlapping features of training methods. In *The Twelfth International Conference on Learning Representations, ICLR*, 2022.

[25] Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 3832–3842. PMLR, 2020.

[26] 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.

[27] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

[28] Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. Emr-merging: Tuning-free high-performance model merging. *Advances in Neural Information Processing Systems*, 37:122741–122769, 2024.

[29] Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

[30] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[31] Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations, ICLR*, 2023.

[32] Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. When do flat minima optimizers work? *Advances in Neural Information Processing Systems*, 35:16577–16595, 2022.

[33] Ron Kohavi and David Wolpert. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, ICML'96, page 275–283, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[34] Piotr Koniusz and Hongguang Zhang. Power normalizations in fine-grained image, few-shot image and graph classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):591–609, 2021.

[35] 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.

[36] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[37] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[38] Jiaqi Li, Qianshan Wei, Chuanyi Zhang, Guilin Qi, Miaozeng Du, Yongrui Chen, Sheng Bi, and Fan Liu. Single image unlearning: Efficient machine unlearning in multimodal large language models. *Advances in Neural Information Processing Systems*, 37:35414–35453, 2024.

[39] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[40] Zhibin Li, Piotr Koniusz, Lu Zhang, Daniel Edward Pagendam, and Peyman Moghadam. Exploiting field dependencies for learning on categorical data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(11):13509–13522, November 2023.

[41] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[42] Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14, 2025.

[43] Zheyuan Liu, Guangyao Dou, Xiangchi Yuan, Chunhui Zhang, Zhaoxuan Tan, and Meng Jiang. Modality-aware neuron pruning for unlearning in multimodal large language models. *arXiv preprint arXiv:2502.15910*, 2025.

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations, ICLR*, 2019.

[45] Chengzhi Mao, Scott Geng, Junfeng Yang, Xin Wang, and Carl Vondrick. Understanding zero-shot adversarial robustness for large-scale models. In *The Eleventh International Conference on Learning Representations,ICLR*, 2023.

[46] Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

[47] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 4. Granada, 2011.

[48] Yao Ni, Shan Zhang, and Piotr Koniusz. PACE: marrying the generalization of PArameter-efficient fine-tuning with consistency regularization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[49] Cristian Rodriguez Opazo, Ehsan Abbasnejad, Damien Teney, Edison Marrese-Taylor, Hamed Damirchi, and Anton van den Hengel. Synergy and diversity in clip: Enhancing performance through adaptive backbone ensembling. In *The Twelfth International Conference on Learning Representations, ICLR*, 2025.

[50] 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*, 36:66727–66754, 2023.

[51] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[52] 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.

[53] Protection Regulation. General data protection regulation. *Intouch*, 25:1–5, 2018.

[54] Wei Ruan, Tianze Yang, Yifan Zhou, Tianming Liu, and Jin Lu. From task-specific models to unified systems: A review of model merging approaches. *arXiv preprint arXiv:2503.08998*, 2025.

[55] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[56] 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.

[57] Jing Wu and Mehrtash Harandi. Munba: Machine unlearning via nash bargaining. *arXiv preprint arXiv:2411.15537*, 2024.

[58] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[59] 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*, 36:7093–7115, 2023.

[60] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations, ICLR*, 2024.

[61] Tianyu Yang, Lisen Dai, Zheyuan Liu, Xiangqi Wang, Meng Jiang, Yapeng Tian, and Xiangliang Zhang. Cliperase: Efficient unlearning of visual-textual associations in clip. *arXiv preprint arXiv:2410.23330*, 2024.

[62] Yongquan Yang, Haijun Lv, and Ning Chen. A survey on ensemble learning under the era of deep learning. *Artificial Intelligence Review*, 56(6):5545–5589, 2023.

[63] Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. Machine unlearning of pre-trained large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*, pages 8403–8419, 2024.

[64] Xu-Cheng Yin, Kaizhu Huang, Chun Yang, and Hong-Wei Hao. Convex ensemble learning with sparsity and diversity. *Information Fusion*, 20:49–59, 2014.

[65] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[66] Yifei Zhang, Hao Zhu, Junhao Dong, Haoran Shi, Ziqiao Meng, and Han Yu Piotr Koniusz. CrossSpectra: exploiting cross-layer smoothness for parameter-efficient fine-tuning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[67] Hao Zhu, Yifei Zhang, Junhao Dong, and Piotr Koniusz. BiLoRA: almost-orthogonal parameter spaces for continual learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25613–25622, 2025.

Machine Unlearning *via* Task Simplex Arithmetic (*Supplementary Material*)

Junhao Dong[1,2], Hao Zhu[3], Yifei Zhang[1], Xinghua Qu[3], Yew-Soon Ong[1,2*], and Piotr Koniusz[4,5,6*]
[1]Nanyang Technological University, [2]CFAR, IHPC, A*STAR, [3]Bytedance,
[4]Data61♥CSIRO, [5]University of New South Wales, [6]Australian National University

A Further Experimental Configurations

Below, we provide details of our experimental setups of Vision-Language Model (VLM) unlearning in terms of dataset description, formulation of compared approaches, and implementation details.

A.1 Dataset Setups

Following prior task vector-based unlearning works [29, 50], we evaluate the unlearning efficacy of VLMs across eight datasets that span a diverse range of recognition tasks, collectively forming the **forget set**. The **retain set** is fixed as the ImageNet dataset [7], which serves to assess the preservation of general knowledge during unlearning. The eight datasets used as forget sets are categorized by their recognition scenario as follows:

1. **Fine-grained classification**: Stanford Cars [35].
2. **Texture recognition**: Describable Textures Dataset (DTD) [5].
3. **Remote sensing and aerial imagery**: EuroSAT [26], and Remote Sensing Image Scene Classification (RESISC) [3].
4. **Traffic and digit recognition**: German Traffic Sign Recognition Benchmark (GTSRB) [55], MNIST [37], and Street View House Numbers (SVHN) [47].
5. **Scene recognition**: SUN397 [58].

For each dataset (eight datasets in total) in the forget set, we obtain the task vector by fine-tuning the CLIP model on its training split. Unlearning evaluation is then performed using the ImageNet validation set, as its test labels are not publicly available. Note that the task vector is obtained through fine-tuning on the training set of each forget set (eight datasets in total), while the unlearning performance evaluation is conducted on the validation set of ImageNet, as the ground-truth labels for its test set are not publicly available. To promote diversity in the task vector space, we vary the data augmentation configuration during fine-tuning by applying RandAugment [6] with different hyperparameter settings. Specifically, we vary the number of sequential augmentation transformations from 1 to 3, and the transformation magnitude from 1 to 10. This results in a total of 30 fine-tuned CLIP models per dataset for each architecture, yielding a rich set of task vectors for subsequent unlearning experiments.

A.2 Detailed Formulations of Compared Methods

To comprehensively evaluate the effectiveness of our proposed function-level ensemble unlearning framework, we compare it with several representative baselines. These include standard task vector–based unlearning [29], diverse task vector–adapted model merging strategies [56, 59, 28], and a vanilla function-level ensemble baseline [8]. Note that the model merging is adapted to the task

*Corresponding authors. PK also in charge of theory & derivations.

vector domain by simply treating the task vector as a unique set of network parameters. Below, we outline how each method is formulated and applied in the context of vision-language model (VLM) unlearning.

1. **Standard Task Arithmetic** [29]. The unlearned model is obtained by subtracting a single task vector from the fine-tuned model, *i.e.*, $f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau})$. This method is retraining-free and efficient, but may suffer from instability due to sensitivity to fine-tuning configurations.
2. **Best Model on the Validation Set** [56]. The unlearned task vector is selected based on the best unlearning efficacy (the lowest forget set accuracy) from the validation set. The unlearned VLM is obtained via subtracting the selected task vector, *i.e.*, $f(\boldsymbol{x}; \arg\min_i \mathrm{ValAcc}(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i))$.
3. **Uniform Merge** [56]. First, compute the average of multiple task vectors, and then subtract the resulting average task vector from the base model to obtain the unlearned model, *i.e.*, $f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\frac{1}{Q}\sum_{i=1}^{Q}\boldsymbol{\tau}_i)$. This can also be regarded as a parameter-level ensemble.
4. **Greedy Merge** [56]. Iteratively select and merge task vectors based on a greedy criterion (*i.e.*, minimizing unlearning score) to optimize unlearning performance.
5. **TIES-Merging** [59]. Trim low-magnitude changes in the values of task vectors and then resolve sign disagreements across the task vectors being merged.
6. **EMR-Merging** [28]. First, elect a unified task vector from all the task vectors and then generate their corresponding modulators, including masks and rescalers (coefficients), to align the direction and magnitude between the unified task vector and each specific task vector, respectively.
7. **Function ensemble** [8]. To isolate the benefit of our proposed function-level closed-form solution, we consider a vanilla ensemble of models whose parameters are fine-tuned with different forget datasets. The predictions from each unlearned model are averaged at inference time, *i.e.*, $\frac{1}{Q}\sum_{i=1}^{Q} f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)$. This ensemble does not require task vector/model merging.

A.3 Implementation Details

In accordance with previous task arithmetic approaches [29, 50], we employ the standard CLIP fine-tuning protocol to generate task vectors for each forget dataset, unless specified otherwise. For each forget dataset, we fine-tune the CLIP model using the AdamW optimizer [44] with a cosine learning rate schedule. The peak learning rate is set to 1×10^{-5} with momentum coefficients $(\beta_1, \beta_2) = (0.9, 0.999)$ and a weight decay of 0.1. The CLIP text encoder is frozen throughout to preserve the integrity of the pre-trained textual embeddings, which serve as zero-shot classification prompts. The vision encoder is updated exclusively. Accordingly, the number of training epochs is set based on dataset characteristics: 70 epochs for Stanford Cars, 100 epochs for DTD, 40 epochs for EuroSAT, GTSRB, RESISC45, and SUN397, and 30 epochs for MNIST and SVHN. We evaluate unlearning performance across three widely used CLIP variants released in [52]: ViT-B/32, ViT-B/16, and ViT-L/14. All models are initialized from pre-trained OpenAI weights and kept fixed for fair comparison across methods.

Unless otherwise defined, the task vector simplex is constructed using $Q = 30$ vertices (i.e., 30 diverse task vectors per forget dataset). The coefficient λ controlling the magnitude of task vector subtraction is selected per forget dataset based on unlearning efficacy measured on a small held-out split, *i.e.*, $\lambda \in \{0.0, 0.05, 0.1, \ldots, 1.0\}$, mimicking a tuning-free practical scenario. We choose λ that achieves the lowest forget set accuracy while the unlearned VLM still preserves at least 95% of the accuracy of the pre-trained VLM on the retain dataset (ImageNet). For the closed-form function-level ensemble and its distillation, the bias-variance weighting factor is fixed to $\beta = 2.0$, which we find consistently balances forget and retain performance across datasets. All the experiments in this paper are conducted based on eight NVIDIA H100 GPUs with 80GB of memory.

B Standard Unlearning Performance for Individual Datasets

Unlearning w.r.t. CLIP ViT-Base/32 backbone. We here provide the VLM unlearning results across eight datasets using the architecture of CLIP ViT-Base/32 in Table 9. We can easily observe that our method and its distillation variant generally achieve great unlearning efficacy on the forget datasets while preserving most of the retained knowledge on other datasets. Furthermore, our

Table 9: **CLIP ViT-Base/32 unlearning** measured by the individual accuracy (%) on each forget dataset and its corresponding accuracy (%) on the retain dataset (ImageNet).

Method	Cars		DTD		EuroSAT		GTSRB		MNIST		RESISC45		SUN397		SVHN	
	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)
Pre-trained Model	59.6	63.3	44.4	63.3	44.7	63.3	32.6	63.3	48.3	63.3	60.3	63.3	63.2	63.3	31.6	63.3
Standard Task Arithmetic [29]	34.8	60.8	29.6	60.5	12.0	60.9	8.8	60.1	16.4	60.6	31.9	61.1	52.1	60.9	8.3	61.1
Best Model on Val. Set [56]	32.0	60.4	28.8	60.5	10.7	60.6	8.1	60.4	13.6	61.3	29.7	60.3	51.1	60.3	7.7	60.6
Vector Uniform Merge [56]	32.6	60.7	29.2	60.8	10.5	60.4	7.1	60.7	16.5	60.8	26.0	60.3	50.1	60.4	12.2	61.1
Vector Greedy Merge [56]	32.0	60.3	28.6	60.5	10.0	60.3	7.4	60.4	11.5	60.7	29.9	60.6	49.8	60.3	7.7	60.6
Vector TIES-Merging [59]	33.6	60.3	30.5	60.3	10.7	60.1	8.4	60.3	15.8	60.6	31.5	60.0	52.6	60.4	8.4	60.3
Vector EMR-Merging [28]	31.0	60.1	27.8	60.4	9.8	60.2	7.5	60.4	13.9	60.5	29.2	60.4	47.9	59.9	7.5	60.8
Function Ensemble [56, 8]	32.3	60.1	29.0	60.2	10.2	60.1	7.8	60.4	14.5	60.6	30.4	60.3	49.9	60.0	7.8	60.7
Ours	28.7	60.7	23.6	60.5	3.2	60.4	1.8	60.6	1.2	60.9	16.0	60.4	44.2	60.7	2.0	60.8
Ours (Distillation)	29.2	60.9	23.9	60.7	3.6	60.6	2.5	60.8	1.4	61.1	16.6	60.6	45.5	60.9	2.6	61.0

Table 10: **CLIP ViT-Base/16 unlearning** measured by the individual accuracy (%) on each forget dataset and its corresponding accuracy (%) on the retain dataset (ImageNet).

Method	Cars		DTD		EuroSAT		GTSRB		MNIST		RESISC45		SUN397		SVHN	
	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)
Pre-trained Model	64.7	68.3	44.7	68.3	55.3	68.3	43.4	68.3	51.7	68.3	66.4	68.3	65.5	68.3	51.9	68.3
Standard Task Arithmetic [29]	27.8	64.7	24.1	64.3	12.4	65.0	7.6	64.5	9.4	65.3	27.3	64.7	49.6	64.8	6.4	64.8
Best Model on Val. Set [56]	30.0	63.9	23.1	64.2	9.9	64.6	7.6	64.3	5.4	65.1	27.7	64.9	48.6	63.9	6.6	64.5
Vector Uniform Merge [56]	28.6	64.6	21.7	63.3	10.6	64.9	8.4	64.9	13.2	65.5	24.0	64.5	49.2	64.7	6.5	64.8
Vector Greedy Merge [56]	30.0	63.9	23.1	64.0	10.2	64.8	7.6	64.3	4.1	65.3	23.8	64.5	49.1	64.6	6.4	64.7
Vector TIES-Merging [59]	28.3	64.6	25.7	64.6	9.0	64.4	7.1	64.6	13.3	64.9	26.6	64.3	44.4	64.7	7.1	64.6
Vector EMR-Merging [28]	26.8	64.5	24.3	64.6	8.5	64.1	6.7	64.7	12.6	64.9	25.2	64.1	42.0	64.5	6.7	64.7
Function Ensemble [56, 8]	27.9	64.5	25.3	64.5	8.9	64.3	7.0	64.5	13.1	64.8	26.2	64.2	43.7	64.6	7.0	64.5
Ours	20.3	64.9	14.7	64.9	1.4	64.7	0.9	64.9	0.8	65.3	16.1	64.6	42.0	65.1	1.3	64.9
Ours (Distillation)	21.5	64.7	15.5	64.7	1.7	64.5	1.0	64.7	0.9	65.1	16.7	64.4	43.1	64.8	1.2	64.7

Table 11: **CLIP ViT-Large/14 unlearning** measured by the individual accuracy (%) on each forget dataset and its corresponding accuracy (%) on the retain dataset (ImageNet).

Method	Cars		DTD		EuroSAT		GTSRB		MNIST		RESISC45		SUN397		SVHN	
	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)	F (↓)	R (↑)
Pre-trained Model	77.9	75.5	55.4	75.5	63.6	75.5	50.5	75.5	76.3	75.5	71.3	75.5	68.2	75.5	58.4	75.5
Standard Task Arithmetic [29]	16.9	70.8	21.8	70.7	9.5	71.9	7.7	71.9	3.4	74.6	16.9	71.8	50.5	71.8	6.7	73.3
Best Model on Val. Set [56]	21.7	71.3	22.3	70.5	5.4	72.2	5.2	71.2	3.4	73.9	14.1	70.9	50.4	71.3	6.7	71.9
Vector Uniform Merge [56]	21.7	71.4	23.2	71.8	9.7	72.0	5.3	71.7	8.7	74.0	13.3	71.7	49.8	71.7	6.7	72.3
Vector Greedy Merge [56]	20.6	70.6	22.8	71.1	10.4	72.2	6.8	71.3	5.3	73.4	11.0	70.8	50.4	71.1	6.9	72.9
Vector TIES-Merging [59]	22.4	72.6	23.9	72.5	10.0	72.3	5.5	72.4	9.0	73.7	13.7	71.3	51.3	72.6	6.9	72.5
Vector EMR-Merging [28]	19.7	71.8	21.0	72.0	8.8	71.7	4.8	71.8	7.9	72.0	12.0	72.0	45.1	71.7	6.1	72.1
Function Ensemble [56, 8]	20.5	71.9	21.9	72.0	9.2	71.7	5.0	71.9	8.2	72.0	12.5	71.9	47.0	71.8	6.3	72.0
Ours	14.0	72.2	11.6	72.1	0.9	72.0	0.7	72.1	0.4	72.4	9.5	72.2	41.3	72.3	1.6	72.2
Ours (Distillation)	15.1	71.9	12.5	72.5	1.0	72.7	0.8	72.6	0.5	72.8	10.0	72.3	43.3	72.7	1.8	72.8

distillation approach maintains comparable effectiveness, indicating the utility of function-level ensemble compression into a single task vector for efficiency.

Unlearning w.r.t. CLIP ViT-Base/16 backbone. Our method also shows consistent performance gains with CLIP ViT-Base/16 backbone, as shown in Table 10. Specifically, we achieve more uniformly low forget-set accuracy, which reflects stable task vector disentanglement at the function level. Moreover, we observe minimal degradation in the ImageNet accuracy, further confirming that the retain-set supervision/knowledge is well preserved.

Unlearning w.r.t. CLIP ViT-Large/14 backbone. With a stronger-capacity CLIP ViT-Large/14 backbone, our proposed method continues to outperform all other VLM unlearning approaches (see Table 11). We can easily observe that the forget-set accuracy across all the datasets is substantially reduced compared to lightweight CLIP architectures. In the meantime, the larger model capacity appears to amplify the benefits of our approach, as our closed-form solution is able to separate task-specific information from generalizable knowledge.

C Further Details of Task Vector Linearization

We have investigated an additional VLM unlearning scenario (see Table 2) regarding linearized task vectors grounded in the Neural Tangent Kernel (NTK) theory [30], which has recently demonstrated improved unlearning effectiveness. In the tangent space (linear) approach, the VLM function $f'(\boldsymbol{x};\boldsymbol{\theta}_0+\boldsymbol{\tau}')$ is linearized via a first-order Taylor expansion at $\boldsymbol{\theta}_0$. This yields a linearized VLM in which weight changes produce linear changes in the output: $f'(\boldsymbol{x};\boldsymbol{\theta}_0+\boldsymbol{\tau}') = f(\boldsymbol{x};\boldsymbol{\theta}_0)+\boldsymbol{\tau}'^{\top}\nabla_{\boldsymbol{\theta}_0} f(\boldsymbol{x};\boldsymbol{\theta}_0)$, where $\boldsymbol{\tau}'$ is defined as the linearized task vector. By simply replacing

standard task vectors with their linearized counterparts and adopting the linear approximation f', we achieve VLM unlearning in the context of tangent-space task vectors.

Thus, for each forget dataset, we linearly fine-tune a diversity of linearized task vectors as the standard unlearning cases. We then build the corresponding linearized task vector simplex for a closed-form function ensemble. During evaluations, we follow the same protocol as the standard unlearning detailed in Appendix A.3, reporting both forget-set and retain-set accuracy. More details are in [50].

D Further Details of Incremental Unlearning

To comprehensively evaluate incremental unlearning performance, we follow an unlearning protocol across a predefined sequence of diverse datasets: *Cars*, *DTD*, *EuroSAT*, *GTSRB*, *MNIST*, *RESISC45*, *SUN397*, and *SVHN* (see Table 3). Each dataset represents distinct distributional characteristics, enabling an assessment of how cumulative distribution shifts influence unlearning efficacy. We employ the CLIP ViT-B/32 model and follow the standard fine-tuning setup as detailed in the main text. Each forget dataset is individually fine-tuned to derive corresponding task vectors, subsequently subtracted sequentially from the current model to simulate incremental forgetting.

To quantify cumulative unlearning effectiveness, after each sequential subtraction of a task vector, we evaluate the VLM on all datasets that have been unlearned up to that step (forget dataset). Specifically, we compute the forget-set accuracy for each previously evaluated dataset and the current one, reporting their average as the average forget-set accuracy. This sequential-average forget accuracy clearly indicates how effectively information from multiple tasks is incrementally erased. Lower values indicate superior unlearning capability.

E Task Vector Re-weighting Guided by Cross-entropy Loss

Recall that we investigate an adaptive re-weighting alternative based on the Cross-Entropy (CE) loss evaluated on the forget dataset (see Table 6), which servers as a performance proxy for each task vector. A higher CE loss indicates that the VLM predictions are significantly deviating from the forget samples, thus reflecting better forgetting performance. Let l_i^{forget} denote the CE loss value for the forget dataset corresponding to the i-th task vector. To adaptively quantify each task vector's relative importance, we adopt a softmax normalization defined as: $\mathbf{w}_i = \exp(l_i^{\text{forget}})/\sum_j \exp(l_j^{\text{forget}})$.

While CE-guided re-weighting provides an effective alternative to prioritizing task vectors that induce stronger forgetting (*i.e.*, higher CE loss on the forget set), its effectiveness remains limited. As shown in Table 6, the CE-based strategy yields marginal improvements over uniform weighting, yet falls short of the performance achieved by our learned re-weighting scheme. This discrepancy stems from the fact that CE losses are computed independently for each task vector, lacking the capacity to model interactions or correlations across different forget directions.

In contrast, our learned re-weighting approach benefits from being jointly optimized with the overall objective, allowing it to adaptively capture complex interdependencies among task vectors. Notably, the learned weights are implicitly regularized by the prediction variance minimization objective, enabling more effective machine unlearning.

F Broader Impact and Limitations.

F.1 Broader Impact

The growing deployment of large-scale VLMs consistently brings increasing societal pressure to support machine unlearning, enabling systems to forget specific information upon request. This need arises from regulatory requirements such as the right to be forgotten and ethical concerns over data misuse or retention. Our proposed task vector simplex unlearning framework provides an efficient alternative to re-training, enabling scalable removal of targeted knowledge from pretrained VLMs with minimal computational overhead.

The task vector formulation also promotes a compositional and interpretable unlearning mechanism, where users can flexibly combine, subtract, or re-weight different task-level behaviors. This capability facilitates responsible model editing and personalized deployment.

The broader impact of our work includes:

- **Data compliance and privacy.** Our proposed method supports a data removal scheme without full retraining, providing a feasible path toward compliance with modern data privacy regulations and ethical unlearning requirements.
- **Resource-efficient unlearning.** By enabling efficient unlearning through task vector simplex, our method lowers the barrier for deploying unlearning mechanisms in the context of VLMs.
- **Modular AI systems.** The compositional structure of task vectors introduces a way toward plug-and-play model updates, supporting dynamic behavior editing in foundation models.

F.2 Limitations

Despite the strong unlearning performance demonstrated by our closed-form task vector simplex approach for VLMs, several limitations still remain, which we aim to address or mitigate throughout the paper, as outlined below:

- **Dependence on forget data availability.** Our method assumes access to representative/partial data from the target forget task to extract task vectors. While this requirement may appear restrictive, it is often reasonable in real-world scenarios where users explicitly request the removal of identifiable data or task-specific knowledge.
- **Function-level ensemble dependency.** Our main method operates at the function level by aggregating multiple task vectors through a simplex-based ensemble. While this strategy is flexible and effective, it may raise concerns about architectural compatibility or deployment constraints. To address this, we provide an extension that distills the ensemble's behavior (predictions) into a single set of model parameters, yielding a fully unlearned model that remains structurally identical to the original VLM.

G Proofs.

G.1 Corollary 1

Proof. Let each $\boldsymbol{\tau}$ be parameterized in barycentric coordinates $\boldsymbol{\tau} = \sum_{i=1}^{Q} \alpha_i \boldsymbol{\tau}_i$ with $\alpha_i \geq 0$ and $\sum_{i=1}^{Q} \alpha_i = 1$. Let each w be parameterized as $w = \sum_{i=1}^{Q} \alpha_i w_i$.

Notice $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_Q) \in \mathbb{R}^Q$ is uniformly distributed over the $(Q-1)$-simplex, *i.e.*, $Dir(\mathbf{1}_Q)$.

Moreover, we have the following expectations:

$$\begin{aligned}
\mathbb{E}[\alpha_i^3] &= \frac{6}{Q(Q+1)(Q+2)}, \\
\mathbb{E}[\alpha_i^2 \alpha_j] &= \frac{2}{Q(Q+1)(Q+2)} \text{ iff } i \neq j, \\
\mathbb{E}[\alpha_i \alpha_j \alpha_k] &= \frac{1}{Q(Q+1)(Q+2)} \text{ iff } i \neq j, i \neq k, j \neq k,
\end{aligned} \tag{17}$$

which are based on the following formula:

$$\mathbb{E}\big[\alpha_1^{p_1} \cdots \alpha_q^{p_Q}\big] = \frac{p_1! \, p_2! \cdots p_Q!}{Q(Q+1) \cdots \big(Q + (\sum_{i=1}^{Q} p_i) - 1\big)} \tag{18}$$

because we have density:

$$f(\alpha_1, \ldots, \alpha_Q) = \frac{1}{Beta(1, \ldots, 1)} \prod_{i=1}^{Q} \alpha_i^{1-1} = (Q-1)! \, \mathbf{1}_{\{\alpha_i \geq 0, \sum \alpha_i = 1\}}, \tag{19}$$

where $Beta(\alpha_1, \ldots, \alpha_Q) = \frac{\prod_{i=1}^{Q} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{Q} \alpha_i)}$ is the multivariate Beta function. In order to get raw moments, one computes:

$$\mathbb{E}\big[\alpha_1^{p_1} \cdots \alpha_Q^{p_Q}\big] = \int_{\sum \alpha_i = 1, \, \alpha_i \geq 0} \alpha_1^{p_1} \cdots \alpha_Q^{p_Q} \, \frac{1}{Beta(1, \ldots, 1)} \, d\boldsymbol{\alpha}, \tag{20}$$

where $Beta(1,\ldots,1)=\frac{\prod_i \Gamma(1)}{\Gamma(Q)}=\frac{1}{(Q-1)!}$.

$$\mathbb{E}[\alpha_1^{p_1}\cdots\alpha_Q^{p_Q}] = \frac{1}{Beta(1,\ldots,1)}\,Beta(1+p_1,\,1+p_2,\,\ldots,\,1+p_Q) \tag{21}$$

$$= (Q-1)!\,\frac{\prod_{i=1}^{Q}\Gamma(1+p_i)}{\Gamma\big(Q+\sum_{i=1}^{Q}p_i\big)} = \frac{p_1!\,p_2!\,\cdots\,p_Q!}{Q(Q+1)\,\cdots\,\big(Q+(\sum_{i=1}^{Q}p_i)-1\big)}, \tag{22}$$

which is exactly Eq. (18).

i. For the choice of three variables α_i, α_j and α_k, Eq. (18) can simply be expanded as follows $\mathbb{E}[\alpha_i\alpha_j\alpha_k]=\mathbb{E}\big[\alpha_1^0\cdots\alpha_{i-1}^0\alpha_i^1\alpha_j^1\alpha_k^1\alpha_{k+1}^0\cdots\alpha_Q^{p_Q}\big]=\frac{0!\cdots 0!\,1!\,1!\,1!\,0!\cdots 0!}{Q(Q+1)\cdots(Q+3-1)}=\frac{1}{Q(Q+1)(Q+2)}$.

ii. For the choice of two variables α_i^2 and α_j, Eq. (18) can simply be expanded as follows $\mathbb{E}[\alpha_i^2\alpha_j]=\mathbb{E}\big[\alpha_1^0\cdots\alpha_{i-1}^0\alpha_i^2\alpha_j^1\alpha_{j+1}^0\cdots\alpha_Q^{p_Q}\big]=\frac{0!\cdots 0!\,2!\,1!\,0!\cdots 0!}{Q(Q+1)\cdots(Q+3-1)}=\frac{2}{Q(Q+1)(Q+2)}$.

iii. For the choice of one variable α_i^3, Eq. (18) can simply be expanded as follows $\mathbb{E}[\alpha_i^3]=\mathbb{E}\big[\alpha_1^0\cdots\alpha_{i-1}^0\alpha_i^3\alpha_{i+1}^0\cdots\alpha_Q^{p_Q}\big]=\frac{0!\cdots 0!\,3!\,0!\cdots 0!}{Q(Q+1)\cdots(Q+3-1)}=\frac{6}{Q(Q+1)(Q+2)}$.

Now we expand $w\,\boldsymbol{\tau}^\top\boldsymbol{H}\boldsymbol{\tau}=\sum_{i=1}^{Q}\sum_{j=1}^{Q}\sum_{k=1}^{Q}\alpha_i\alpha_j\alpha_k w_i\boldsymbol{\tau}_j^\top\boldsymbol{H}\boldsymbol{\tau}_k$, substitute Eq. (17) and compute expectations:

$$\mathbb{E}[w\,\boldsymbol{\tau}^\top\boldsymbol{H}\boldsymbol{\tau}]=\frac{1}{Q(Q+1)(Q+2)}\Bigg(\sum_{i=1}^{Q}(2+4\,w_i)\,\boldsymbol{\tau}_i^\top\boldsymbol{H}\boldsymbol{\tau}_i+\sum_{\substack{i,j=1\\ i\neq j}}^{Q}\big(1+w_i+w_j\big)\,\boldsymbol{\tau}_i^\top\boldsymbol{H}\boldsymbol{\tau}_j\Bigg). \tag{23}$$

This completes the proof. □

G.2 Advanced Aggregation

Proof. Eq. (8) emerges from the following set of transitions:

$$\begin{aligned}
&1-\prod_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\big(1-f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})\big)\\
&\quad=1-\exp\Big[\log\Big(\prod_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\big(1-f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})\big)\Big)\Big]\\
&\quad=1-\exp\Big[\sum_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\log\big(1-f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})\big)\Big]\\
&\quad=1-\exp\Big[|\Delta_{\boldsymbol{\theta}}|\frac{1}{|\Delta_{\boldsymbol{\theta}}|}\sum_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\log\big(1-f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})\big)\Big]\\
&\quad=1-\exp\big[|\Delta_{\boldsymbol{\theta}}|\mu_g(\boldsymbol{x})\big].
\end{aligned} \tag{24}$$

Moreover, as $\prod_{i=1}^{Q}(1-p_i^c)$ is the probability of zero successes in Q independent Bernoulli trials that are not necessarily identically distributed (Wadycki *et al.* [71]), thus $1-\prod_{i=1}^{Q}(1-p_i^c)$ is the probability of at least one successes in Q independent Bernoulli trials that are not necessarily identically distributed.

Therefore, it follows that infinitely sampling trials from the simplex $\Delta_{\boldsymbol{\theta}}$, as in $1-\prod_{\boldsymbol{\tau}\in\Delta_{\boldsymbol{\theta}}}\big(1-f(\boldsymbol{x};\boldsymbol{\theta}_0-\lambda\boldsymbol{\tau})\big)$, yields the probability of at least one successes in the infinite number of independent Bernoulli trials that are not necessarily identically distributed, and are sampled from simplex $\Delta_{\boldsymbol{\theta}}$. □

Notice that as function f varies due to changes of task vector$\boldsymbol{\tau}$, underlying trials are not necessarily identically distributed, and that is why we employed the PMF of the Poisson binomial distribution.

G.3 Volume of Simplex

For detailed derivations of the volume formula of simplex refer to the work by Stein [70].

G.4 Bias-variance Trade-off

Proof. Let $\boldsymbol{y}(\boldsymbol{x}) = f^*(\boldsymbol{x}) + \boldsymbol{\eta}$ with $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\gamma}^2)$. Let $\boldsymbol{\tau} = \boldsymbol{\tau}(\mathcal{D})$ be a task vector learned by fine-tuning on some augmented dataset $\mathcal{D}$ derived from unlearning dataset $\mathcal{D}_0$. Let predictor be $f(\boldsymbol{x}) = f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau})$.

Consider the following expression

$$\boldsymbol{y} - f(\boldsymbol{x}) = \big(f^*(\boldsymbol{x}) + \boldsymbol{\eta}\big) - f(\boldsymbol{x}) = \big(f^*(\boldsymbol{x}) - f(\boldsymbol{x})\big) + \boldsymbol{\eta} \tag{25}$$

and observe that

$$\|\boldsymbol{y}(\boldsymbol{x}) - f(\boldsymbol{x})\|_2^2 = \big\|f^*(\boldsymbol{x}) - f(\boldsymbol{x})\big\|_2^2 + 2\big\langle f^*(\boldsymbol{x}) - f(\boldsymbol{x}), \boldsymbol{\eta}\big\rangle + \|\boldsymbol{\eta}\|_2^2. \tag{26}$$

Consider the following expectation:

$$\mathbb{E}_{\mathcal{D}_0,\boldsymbol{\tau}}\Big[\underbrace{\mathbb{E}_{\boldsymbol{\eta}}\big[\|\boldsymbol{y}(\boldsymbol{x}) - f(\boldsymbol{x})\|_2^2 \mid \mathcal{D}_0, \boldsymbol{\tau}\big]}_{\text{inner expectation}}\Big]. \tag{27}$$

Now, expand its inner expectation as follows:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\eta}}\big[\|\boldsymbol{y}(\boldsymbol{x}) - f(\boldsymbol{x})\|_2^2 \mid \mathcal{D}_0, \boldsymbol{\tau}\big] &= \mathbb{E}_{\boldsymbol{\eta}}\big[\|f^*(\boldsymbol{x}) - f(\boldsymbol{x})\|_2^2\big] + 2\,\mathbb{E}_{\boldsymbol{\eta}}\big[\langle f^*(\boldsymbol{x}) - f(\boldsymbol{x}), \boldsymbol{\eta}\rangle\big] + \mathbb{E}_{\boldsymbol{\eta}}\big[\|\boldsymbol{\eta}\|_2^2\big] \\ &= \|f^*(\boldsymbol{x}) - f(\boldsymbol{x})\|_2^2 + 0 + \|\boldsymbol{\gamma}\|_2^2. \end{aligned} \tag{28}$$

Subsequently, Eq. (27) reduces to the following expression:

$$\mathbb{E}_{\mathcal{D}_0,\boldsymbol{\tau}}\big[\|f^*(\boldsymbol{x}) - f(\boldsymbol{x})\|_2^2\big] + \|\boldsymbol{\gamma}\|_2^2, \tag{29}$$

which we expand below into the squared bias and variance. Let $\boldsymbol{\mu}_f(\boldsymbol{x}) = \mathbb{E}_{\mathcal{D}_0,\boldsymbol{\tau}}\big[f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau})\big]$ and consider that $f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau})$ changes due to changing task vector $\boldsymbol{\tau}$ and dataset $\mathcal{D}_0$, whereas $f^*(\boldsymbol{x})$ does not change w.r.t. changes of $\boldsymbol{\tau}$ or $\mathcal{D}_0$.

Substitute $\boldsymbol{u} = f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \boldsymbol{\mu}_f(\boldsymbol{x})$ and $\boldsymbol{v} = \boldsymbol{\mu}_f(\boldsymbol{x}) - f^*(\boldsymbol{x})$, and apply the following expansion:

$$\begin{aligned}\|\boldsymbol{u} + \boldsymbol{v}\|_2^2 &= \|\boldsymbol{u}\|_2^2 + 2\langle \boldsymbol{u}, \boldsymbol{v}\rangle + \|\boldsymbol{v}\|_2^2 \\ &= \big\|f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \boldsymbol{\mu}_f(\boldsymbol{x})\big\|_2^2 + 2\,\big\langle f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \boldsymbol{\mu}_f(\boldsymbol{x}), \boldsymbol{\mu}_f(\boldsymbol{x}) - f^*(\boldsymbol{x})\big\rangle + \|\boldsymbol{\mu}_f(\boldsymbol{x}) - f^*(\boldsymbol{x})\|_2^2. \end{aligned} \tag{30}$$

Incorporating expectations into Eq. (30) leads to:

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_0,\boldsymbol{\tau}}\big[\|f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - f^*(\boldsymbol{x})\|_2^2\big] &= \underbrace{\mathbb{E}_{\mathcal{D}_0,\boldsymbol{\tau}}\Big[\big\|f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \boldsymbol{\mu}_f(\boldsymbol{x})\big\|_2^2\Big]}_{=\mathbb{E}_{\mathcal{D}_0}\big[\underbrace{\|\boldsymbol{\sigma}_f^2(\boldsymbol{x})\|_1}_{\mathrm{Var}_{\boldsymbol{\tau}}(f(\boldsymbol{x}))}\big]} \\ &\quad + \underbrace{\mathbb{E}_{\mathcal{D}_0,\boldsymbol{\tau}}\Big[2\,\big\langle f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \boldsymbol{\mu}_f(\boldsymbol{x}), \boldsymbol{\mu}_f(\boldsymbol{x}) - f^*(\boldsymbol{x})\big\rangle\Big]}_{=0} \\ &\quad + \mathbb{E}_{\mathcal{D}_0}\Big[\underbrace{\mathbb{E}_{\boldsymbol{\tau}}\big[\|\boldsymbol{\mu}_f(\boldsymbol{x}) - f^*(\boldsymbol{x})\|_2^2\big]}_{\big(\mathrm{Bias}_{\boldsymbol{\tau}} f(\boldsymbol{x})\big)^2}\Big]. \end{aligned} \tag{31}$$

Including $\|\boldsymbol{\gamma}\|_2^2$ from Eq. (29) into the result in Eq. (31) completes the proof. □

The importance of the above proof is that our closed-form ensemble of infinite number of functions with task vectors sampled from task simplex $\Delta_{\boldsymbol{\theta}}$ has the associated with it closed-form variance. Such a variance term tells us how much function $f(\boldsymbol{x}; \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau})$ fluctuates around $\boldsymbol{\mu}_f(\boldsymbol{x})$ over random draws of $\boldsymbol{\tau}$ from the task simplex $\Delta_{\boldsymbol{\theta}}$ and dataset $\mathcal{D}_0$. High variance may result from task vectors overfitting to the unlearning dataset so controlling the variance controls the generalization of our unlearning algorithm.

G.5 Interpolation Bound

Proof. Let $\bar{\boldsymbol{\theta}} = \sum_{i=1}^{Q} \alpha_i(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)$ be the convex combination of perturbed parameters.

Note that:

$$\bar{\boldsymbol{\theta}} = \sum_{i=1}^{Q} \alpha_i(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i) = \boldsymbol{\theta}_0 - \lambda \sum_{i=1}^{Q} \alpha_i\boldsymbol{\tau}_i = \boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}. \tag{32}$$

Next, for $l = 1, \ldots, C'$, we write:

$$\Big|g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \sum_{i=1}^{Q} \alpha_i\, g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)\Big| = \Big|g_l(\bar{\boldsymbol{\theta}}) - \sum_{i=1}^{Q} \alpha_i\, g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)\Big|, \tag{33}$$

where $g_l(\cdot)$ is simply the l^{th} output of the multivariate-output unlearning function $g(\cdot)$.

Let $g_l(\cdot)$ be L_l-Lipschitz continuous, *i.e.*, $|g_l(\boldsymbol{\theta}_i) - g_l(\boldsymbol{\theta}_j)| \leq L_l\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|_2,\ \forall i \neq j$. Then $g(\cdot)$ is L-Lipschitz continuous with $L = \sum_l L_l$ as:

$$\Big\|g(\boldsymbol{\theta}_i) - g(\boldsymbol{\theta}_j)\Big\|_1 = \sum_l \Big|g_l(\boldsymbol{\theta}_i) - g_l(\boldsymbol{\theta}_j)\Big| \leq \sum_l L_l\Big\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\Big\|_2 = L\Big\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\Big\|_2,\ \forall i \neq j. \tag{34}$$

Next, we write:

$$\begin{aligned}
\Big|\sum_{i=1}^{Q} \alpha_i g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i) - g_l(\bar{\boldsymbol{\theta}})\Big| &= \Big|\sum_{i=1}^{Q} \alpha_i\big[g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i) - g_l(\bar{\boldsymbol{\theta}})\big]\Big| && (35)\\
&\leq \sum_{i=1}^{Q} \alpha_i\Big|g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i) - g_l(\bar{\boldsymbol{\theta}})\Big| && (36)\\
&\leq \sum_{i=1}^{Q} \alpha_i L_l\big\|(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i) - \bar{\boldsymbol{\theta}}\big\|_2 && (37)\\
&= L_l \sum_{i=1}^{Q} \alpha_i\big\|\lambda(\boldsymbol{\tau} - \boldsymbol{\tau}_i)\big\|_2 && (38)\\
&= L_l\lambda \sum_{i=1}^{Q} \alpha_i\big\|\boldsymbol{\tau} - \boldsymbol{\tau}_i\big\|_2. && (39)
\end{aligned}$$

Since $\boldsymbol{\tau} = \sum_{j=1}^{Q} \alpha_j\boldsymbol{\tau}_j$, we have:

$$\Big\|\boldsymbol{\tau} - \boldsymbol{\tau}_i\Big\|_2 = \Big\|\sum_{j=1}^{Q} \alpha_j(\boldsymbol{\tau}_j - \boldsymbol{\tau}_i)\Big\|_2 \leq \sum_{j=1}^{Q} \alpha_j\Big\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\Big\|_2 \leq \max_j \Big\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\Big\|_2. \tag{40}$$

Therefore:

$$\Big|g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \sum_{i=1}^{Q} \alpha_i g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)\Big| \leq L_l\lambda \sum_{i=1}^{Q} \alpha_i \max_j \big\|\boldsymbol{\tau}_j - \boldsymbol{\tau}_i\big\|_2 \leq L_l\lambda \max_{i,j} \big\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_j\big\|_2. \tag{41}$$

Now using Eq. (34) it follows that Eq. (41) can be further transformed as:

$$\begin{aligned}
\Big\|g(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \sum_{i=1}^{Q} \alpha_i g(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)\Big\|_1 &= \sum_l \Big|g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}) - \sum_{i=1}^{Q} \alpha_i g_l(\boldsymbol{\theta}_0 - \lambda\boldsymbol{\tau}_i)\Big|\\
&\leq \sum_l L_l\lambda \max_{i,j} \big\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_j\big\|_2 = L\lambda \max_{i,j} \big\|\boldsymbol{\tau}_i - \boldsymbol{\tau}_j\big\|_2. \qquad (42)
\end{aligned}$$

□

H Additional Analyses and Considerations

H.1 Computation of the First-order Taylor Term

For the standard expansion without weighting, we have:

$$\boldsymbol{\mu}_\tau = \frac{1}{|\Delta_\theta|} \sum_{\boldsymbol{\tau} \in \Delta_\theta} \boldsymbol{\tau} = \frac{1}{Q} \sum_{i=1}^{Q} \boldsymbol{\tau}_i, \tag{43}$$

while for Corollary 1 we have:

$$\boldsymbol{\mu}_\tau^\omega = \frac{1}{|\Delta_\theta|} \sum_{(w,\boldsymbol{\tau}) \in \Delta_w \times \Delta_\theta} w\boldsymbol{\tau} = \frac{1}{\sum_{j=1}^{Q} w_j} \sum_{i=1}^{Q} w_i \boldsymbol{\tau}_i. \tag{44}$$

H.2 Sparse Parameter Change and Active Parameter Subset

Few-epochs fine-tuning $\boldsymbol{\theta}_0$ on unlearning dataset toward $\boldsymbol{\theta}_i$ yields sparse change

$$\sum_{j=1}^{m} \mathbb{1}(|\theta_{j,0} - \theta_{j,i}| > \epsilon) \ll m \tag{45}$$

for small ϵ and $m = 86M$ (CLIP ViT-B-16). He *et al*. [68] and Zeng *et al*. [72] observe sparsity between fine-tuned and original models. For $\epsilon = 1e{-}4$ (CLIP ViT-B/16), about 4.6% parameters change.

Thus, in this work we assume that interpolating between such sparse differences in $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$, $i \neq j$ captures a parameter subset responsible for unlearning. Interpolating on such a subset of coefficients serves as choosing varying magnitudes of parameters that are active in unlearning.

H.3 Distributions with Non-compact Support

We assert that the compact support of Dirichlet distribution is beneficial as sampling from it provides a sample of parameter vector obeying the support, and thus not activating parameters irrelevant for the task.

As a counterexample, let task vectors be Normally distributed, *i.e*., $\boldsymbol{\theta}' \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. Let off-diagonal elements be set to 0 as one cannot estimate $m \times m$ dimensional covariance for very large m. Let $\boldsymbol{\mu} = \frac{1}{Q}\sum_{i=1}^{Q} \boldsymbol{\theta}_i$ and $\boldsymbol{\sigma}^2 = \frac{1}{Q-1}\sum_{i=1}^{Q}(\boldsymbol{\theta}_i - \boldsymbol{\mu})^2$ which produces parameter samples $\boldsymbol{\theta}' = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \mathbf{v}$ where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.

Table 12: The Normal distribution *vs*. the Dirichlet distribution ensemble.

Distribution	Forget (↓)	Retain (↑)
Normal (S=100)	16.80	64.32
Normal (S=300)	15.98	64.54
Dirichlet (Ours)	**12.17**	**64.93**

Table 12 shows that the Normal distribution performs worse than the Dirichlet distribution model as for the Normal distribution the slopes of PDF decay slowly toward ∞ whereas the Dirichlet distribution has compact support.

H.4 Parameter Augmentation Perspective

As averaging all task vectors gives viable unlearning (Vector Uniform Merge), and ensembling functions on individual task vectors is viable unlearning (Function Ensemble), interpolation between the mean $\frac{1}{Q}\sum_{i=1}^{Q} \boldsymbol{\tau}_i$ and individual $\boldsymbol{\tau}_j$ can be considered as a *parameter augmentation* strategy.

Let

$$p(\mathbf{x}; \Omega) = \frac{1}{|\Omega|} \int_{\boldsymbol{\tau} \in \Omega} f\big(\mathbf{x}; \boldsymbol{\theta}_0 - \boldsymbol{\tau}\big)\, d\boldsymbol{\tau}. \tag{46}$$

Based on Theorem 4, if $f(\cdot)$ changes fast in Ω_{fast} and slow in Ω_{slow} (within the same neighborhood sizes) then intuitively the entropy of prediction $Ent(p(\mathbf{x}; \Omega_{\text{slow}}))$ is lower than $Ent(p(\mathbf{x}; \Omega_{\text{fast}}))$

as smooth changes contribute coherently to the vote (ensemble), whereas chaotic changes are incoherent leading to cancellation of class peaks; $Ent(\mathrm{vec}(1/C))$ is max. Hence our method leverages smoothness of $f(\cdot)$ w.r.t. augmentation. See Izmailov *et al*. for related considerations [69].

H.5 Proximity of Task Vectors with the Simplex

Task vectors do not follow the Dirichlet distribution *per se*. However, They are located close to the simplex. We take $Q=10$ out of 30 task vectors fine-tuned on SUN397 and build a simplex. We count-sketch dimension down to $Q=d+1=11$ for tractability. The remaining 20 task vectors are within $0.9(\eta/2)$ radius of the simplex while task vectors of Cars exceed $1.6(\eta/2)$ distance (cluster around their own simplex) for diameter η defined for Theorem 4.

H.6 Varying Fine-tuning Numbers Q

In our experiments, we fine-tuned the model over mere 6–35 epochs. We can reduce $Q=30$ down to $Q=10$ to achieve faster speed if parallel fine-tuning is not permitted.

Table 13: Scaling with # task vectors Q (CLIP ViT-B/16).

Q	Forget (↓)	Retain (↑)	Fine-tuning Time	Unlearning Inference Time
10	14.06	64.77	1.9h	372s
15	13.29	64.62	2.9h	458s
30	**12.17**	**64.93**	5.7h	716s

Table 13 provides sequential fine-tuning time and unlearning inference time together with the forget and retain performance.

Table 14 provides sequential fine-tuning time and the forget and retain performance under the epochs range (6–35) reduced by 1/2 or 1/3.

Table 14: Effect of reducing fine-tuning epochs.

Fine-tuning Epoch %	Forget (↓)	Retain (↑)	Fine-tuning Time
1/3 of 6–35 epochs	15.54	65.14	2.4h
1/2 of 6–35 epochs	14.28	65.06	3.2h
5–35 epochs	**12.17**	**64.93**	5.7h

H.7 Impact of λ on Taylor Expansion

Task vectors are combined with weight $0\leq\lambda\leq 1$. Thus, the λ^2 term in the second-order part of the Taylor expansion decays quadratically, while the first-order term λ is linear. In practice, λ is chosen by cross-validation. For very low λ^2, our method reverts to the 1st-order expansion. Table 15 investigates results w.r.t. varied λ (CLIP ViT-B/16) on SUN397. Bold indicates the λ value achieving the best performance in cross-validation.

Table 15: Effect of varying λ (CLIP ViT-B/16, SUN397).

λ	Forget (↓)	Retain (↑)
0.1	61.4	67.6
0.2	57.5	66.7
0.3	53.6	66.2
0.4	49.8	65.7
0.5	45.9	64.8
0.6	**42.0**	**65.1**
0.7	40.1	62.8
0.8	39.2	60.3

H.8 Variance $\boldsymbol{\sigma}^2(\mathbf{x})$ *vs.* $\boldsymbol{\Sigma}(\mathbf{x})$

In experiments we assumed each μ_i for $i=1,\ldots,C$ is independent for simplicity and $f_i(\cdot)$ produces likelihood of the i^{th} class. As some task vectors may be noisy for some classes, we targeted limiting class-wise variance only. In the future, we will consider reducing covariance (off-diagonal terms) to decorrelate the model. This will require rethinking the Taylor expansion to produce the outer product of shape $C\times C$.

H.9 Details of the Bienaymé Formula

For a fixed sample $\mathbf{x}$, one may think of $f(\cdot)$ as a transformation of random variable (task vector sampled from the Dirichlet distribution (simplex)) into another random variable living in the class space (surface of a simplex) which follows some class distribution resulting from this transformation. The Bienaymé formula tells what happens with variance of each class (treated independently) as the number of ensembled functions grows. That number depends on Q task vectors. However, ρ is required to be low. The opposite means each Q functions are not independent. In extreme case, one could have Q identical functions which indeed cannot reduce the variance if they all are identical. To help reduce the variance, we investigated optimizing w_i or $\boldsymbol{\tau}_i$ in a small radius. Reducing off-diagonal terms of covariance could strengthen this effect.

References

[68] Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *Transactions on Machine Learning Research*, 2025.

[69] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *The Conference on Uncertainty in Artificial Intelligence*, pages 876–885. AUAI Press, 2018.

[70] P. Stein. A note on the volume of a simplex. *The American Mathematical Monthly*, 73(3):299–301, 1966.

[71] Walter J. Wadycki, B. K. Shah, P. D. Ghangurde, Edward J. Dudewicz, Nathan Mantel, Charles C. Brown, Harold J. Larson, Donald R. Barr, James W. Frane, Bernard Saperstein, I. J. Good, and Howard L. Jones. Letters to the editor. *The American Statistician*, 27(3):123–127, 1973.

[72] Siqi Zeng, Yifei He, Meitong Liu, Weiqiu You, Yifan Hao, Yao-Hung Hubert Tsai, Makoto Yamada, and Han Zhao. Task vector bases: A unified and scalable framework for compressed task arithmetic, 2025.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the main contributions and scope of the paper, providing a well-motivated rationale for pursuing machine unlearning. The presented claims are substantiated by both rigorous theoretical analysis and comprehensive experimental evaluation.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are discussed in Appendix F.2.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The paper presents thorough theoretical results, with each theorem accompanied by explicitly stated assumptions. Detailed formal proofs are provided in the appendix, while intuitive explanations are included in the main text to enhance clarity.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Detailed experimental setups are provided in Appendix A. For each experiment/anaylsis, we also provide its corresponding configuration for clarity.

Guidelines:

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

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

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

   Answer: [No]

   Justification: Detailed experimental setups for reproducing our results are provided in Appendix A. All the datasets used in the paper are publicly available.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Full details of training and evaluations are provided in Appendix A.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: We repeat each experiment five times under different random seeds and report the mean performance. Consistent trends across runs indicate the robustness of our findings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The required computing resources in our setting are provided in Appendix A.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We have thoroughly checked the NeurIPS Code of Ethics for our research.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Broader positive impacts are discussed in Appendix F.1.

    Guidelines:

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

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

11. **Safeguards**

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

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All the external resources we used are properly mentioned and credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.