NCAR: NEURAL COLLAPSE ALIGNMENT REGULARIZATION FOR MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning aims to remove the influence of requested training data while efficiently approximating the solution induced by retraining on the retained data. Existing gradient-based methods primarily optimize predictive or output-level objectives for forgetting and retention, yet a discrepancy in the retained representation–classifier structure can remain even when predictive behavior is close to retraining outcomes. To address this structural discrepancy, we propose Neural Collapse Alignment Regularization (NCAR), which uses the idealized Neural Collapse terminal geometry of supervised classification as a structural prior. NCAR constructs a canonical simplex over the remaining classes, aligns it with the current classifier, and adaptively fits its translation and scale to retained features to obtain a classifier-aligned structural target. The resulting retain-side regularizer couples within-class concentration and simplex organization of class means with feature–classifier alignment. NCAR can be incorporated into existing gradient-based unlearning objectives while preserving the base method's forgetting-loss construction and parameter-update restrictions. We evaluate NCAR across random-subset and full-class unlearning, multiple datasets, architectures, and base methods, analyzing predictive behavior and retained representation structure together.

1 INTRODUCTION

Machine unlearning aims to remove the influence of training data requested for deletion while efficiently approximating the solution induced by retraining on the remaining data (Cao & Yang, 2015; Bourtoule et al., 2021; Guo et al., 2020; Golatkar et al., 2020). Repeating the original training procedure on the retained data from its prescribed initialization, referred to as *exact retraining*, provides a natural reference for evaluating unlearning, but doing so for every deletion request incurs substantial computational cost. Approximate unlearning reduces this cost by updating a trained model with limited computation. Among these approaches, gradient-based methods commonly combine a forgetting objective on the deleted data with supervision, distillation, or other retention objectives on the remaining data (Kurmanji et al., 2023; Fan et al., 2024; Huang et al., 2024; Di et al., 2026).

Existing gradient-based methods differ in how they construct forgetting signals, select trainable parameters, and preserve retained predictions. Many of these methods optimize forgetting and retention through losses defined on labels, logits, or predictive distributions. These objectives directly supervise predictions but do not explicitly prescribe the geometry of retained representations. Retraining on the retained data, in contrast, induces a new representation together with a classifier structured around it. An unlearned model can therefore achieve the desired forgetting behavior and retained performance while remaining structurally different from the retrained model (Seo et al., 2025; Gao et al., 2026). Using the retrained model's structure as a target, however, would require performing exact retraining, reintroducing the computational cost that unlearning seeks to reduce. This motivates a structural prior that can be constructed without access to retraining outcomes.

Neural Collapse (NC) characterizes the class-conditional representation–classifier geometry that emerges during the terminal phase of supervised classification (Papyan et al., 2020). NC captures within-class concentration (NC1), simplex equiangular tight frame (ETF) organization of class means (NC2), alignment between class means and classifier directions (NC3), and the corresponding nearest-class-center decision structure (NC4). Retained-data retraining is itself a super-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

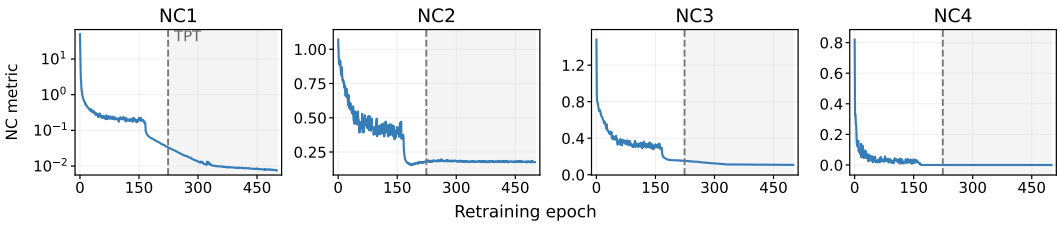


Figure 1: **Neural Collapse during exact retraining.** Evolution of the NC1–NC4 metrics during exact retraining of ResNet-18 on CIFAR-10 after 10% random-subset deletion. Lower values indicate closer agreement with the corresponding ideal NC property. The shaded region marks the terminal phase of training (TPT), beginning when retained-set training accuracy reaches 100%.

vised classification process, and the NC terminal geometry provides an idealized description of the representation–classifier structure induced by this process. Figure 1 shows a representative trajectory of the NC1–NC4 metrics during exact retraining of ResNet-18 on the retained CIFAR-10 subset after 10% random deletion.

Motivated by this observation, we propose *Neural Collapse Alignment Regularization* (NCAR), which uses the idealized Neural Collapse terminal geometry of supervised classification as a structural prior. NCAR constructs a simplex target for the remaining-class means according to NC2, aligns its orientation with the current classifier according to NC3, and fits the resulting class targets to the retained feature space. Retained features are then regularized toward their corresponding class targets, inducing the within-class concentration associated with NC1. In this way, NCAR jointly regularizes the structural properties corresponding to NC1–NC3 within a single retain-side objective.

NCAR provides a plug-in structural regularizer for gradient-based machine unlearning. It constructs its targets without a retrained model or retrained features, preserves the forgetting objective and parameter-update restrictions of the underlying method, and applies to both random-subset and full-class deletion through the same formulation. Across datasets, architectures, and base methods, the reported results show gains in both predictive agreement and retained-feature similarity to exact retraining.

2 Related Work

Machine unlearning. Exact retraining provides a reference for unlearning, and approximate methods reduce repeated training costs through model updates (Cao & Yang, 2015; Bourtoule et al., 2021; Guo et al., 2020; Golatkar et al., 2020). SCRUB uses teacher–student objectives (Kurmanji et al., 2023); Bad Teacher uses competent and incompetent teachers (Chundawat et al., 2023); SalUn restricts updates through weight saliency (Fan et al., 2024); and selective synaptic dampening modifies influential parameters (Foster et al., 2024). UGradSL introduces smoothed-label forgetting objectives (Di et al., 2026). Huang et al. (2024) unify gradient-based updates and incorporate the geometry of the output probability space. Gradient conflicts provide another perspective on the interaction between forgetting and retention (Patel & Qiu, 2025).

Representation-based unlearning. LAF aligns retained representations with those of the original model without label supervision (Shen et al., 2024). Residual Feature Alignment uses low-rank adaptation (LoRA) to preserve intermediate features on retained data while modifying those of forget data (Qin et al., 2026); Contrastive Unlearning operates on retain and forget embeddings (Lee et al., 2025); and Dimensional Alignment constrains latent eigenspaces (Seo et al., 2025). Deep Unlearning uses activation-derived subspaces for class forgetting (Kodge et al., 2024). NCAR complements this research direction with a classifier-conditioned NC target for retained features that can be incorporated into existing unlearning objectives.

Neural Collapse. Neural Collapse (NC) characterizes the simplex-ETF geometry that emerges in terminal supervised training (Papyan et al., 2020), with geometric analyses under unconstrained-

feature models (Zhu et al., 2021). Markou et al. (2024) fit a simplex ETF to feature means and use it to define an implicit classifier. In unlearning, POUR removes target-class directions through projection or projection-guided distillation (Le et al., 2026), while CMF-based methods construct the classifier from class-mean features to enforce feature–classifier alignment (Gao et al., 2026). NCAR instead uses the *current classifier* to orient a simplex target for retained features while retaining the base method's parameter-update rules. The same construction applies to both random-subset and full-class unlearning.

3 METHOD

3.1 PROBLEM FORMULATION

Let $D = \{(x_i, y_i)\}$ be the development training set, with $y_i \in \{1, \dots, C\}$, partitioned as $D = R \dot{\cup} F$ into retained examples R and examples requested for deletion F. The model consists of a feature extractor f_θ with parameters θ and feature dimension d, and a linear classifier $W = [w_1, \dots, w_C] \in \mathbb{R}^{d \times C}$. For an input x_i, its last-layer representation and logits are

$$h_i = f_\theta(x_i) \in \mathbb{R}^d, \qquad z_i = W^\top h_i + b. \tag{1}$$

The optional bias $b \in \mathbb{R}^C$ is zero for a bias-free head. Let $\mathcal{C}_R$ denote the remaining classes represented in R and $C_R = |\mathcal{C}_R|$. Target columns follow a fixed mapping from $\mathcal{C}_R$ to $\{1, \dots, C_R\}$. We write $\mathbf{1}_k$ and I_k for the all-ones vector and identity matrix, $\mathbf{1}[\cdot]$ for an indicator, and $\|\cdot\|_2$ and $\|\cdot\|_F$ for Euclidean and Frobenius norms.

A gradient-based unlearning method with an objective representation supplies

$$\mathcal{L}_{\text{base}}(\theta, W, b; R, F), \tag{2}$$

which may include forgetting, retention, distillation, and method-specific weighting, together with parameter-update restrictions. NCAR adds a retain-side structural regularizer coupling within-class concentration (NC1), simplex organization (NC2), and classifier alignment (NC3). NC4 characterizes the associated nearest-class-center decisions and is evaluated separately rather than directly imposed. Exact retraining provides the selection and evaluation reference, not an input to target construction. Table 6 summarizes the notation.

3.2 NC2: REMAINING-CLASS SIMPLEX GEOMETRY

NC2 characterizes the simplex organization of centered class means. We represent the relative geometry of the classes in R by a fixed canonical simplex equiangular tight frame (ETF) $V = [v_1, \dots, v_{C_R}] \in \mathbb{R}^{d \times C_R}$ satisfying

$$V\mathbf{1}_{C_R} = 0, \qquad V^\top V = \frac{C_R}{C_R - 1}\left(I_{C_R} - \frac{\mathbf{1}_{C_R}\mathbf{1}_{C_R}^\top}{C_R}\right). \tag{3}$$

Thus $\|v_c\|_2 = 1$ and $v_c^\top v_k = -1/(C_R - 1)$ for $c \neq k$. We assume $2 \leq C_R \leq d+1$, the dimensional condition for a nondegenerate simplex with C_R vertices.

Equation 3 fixes the equal class-direction norms and pairwise angles associated with NC2. The remaining orientation is selected using the current classifier.

3.3 NC3: CLASSIFIER-ALIGNED REMAINING-CLASS GEOMETRY

NCAR uses the current classifier to orient the simplex while preserving its NC2 geometry. Collect the classifier columns corresponding to $\mathcal{C}_R$ in $W_R \in \mathbb{R}^{d \times C_R}$ and center them as

$$\bar{w} = \frac{1}{C_R}\sum_{c \in \mathcal{C}_R} w_c, \qquad \bar{W}_R = [w_c - \bar{w}]_{c \in \mathcal{C}_R}. \tag{4}$$

With the numerical floor $\epsilon = 10^{-12}$, normalize the columns as

$$\widehat{w}_c = \frac{w_c - \bar{w}}{\max(\|w_c - \bar{w}\|_2, \epsilon)}, \qquad \widehat{W}_R = [\widehat{w}_c]_{c \in \mathcal{C}_R}. \tag{5}$$

Zero centered columns remain zero. With the same class ordering in V and $\widehat{W}_R$, solve the orthogonal Procrustes problem (Schönemann, 1966)

$$Q_W \in \arg\min_{Q\in\mathbb{R}^{d\times d},\, Q^\top Q=I_d} \|QV - \widehat{W}_R\|_F^2, \qquad A_W = Q_W V. \tag{6}$$

A singular value decomposition (SVD) $\widehat{W}_R V^\top = U_W \Sigma_W^{\mathrm{svd}} Z_W^\top$ gives the solution $Q_W = U_W Z_W^\top$. The equivalent reduced formulation in Appendix C.1 avoids a full $d \times d$ factorization. Orthogonality preserves

$$A_W^\top A_W = V^\top V, \tag{7}$$

so A_W preserves the simplex relations while conditioning its orientation on the classifier. Its columns define the retained-feature target directions.

3.4 NC1: Retained-Feature Targets and Within-Class Concentration

Let $a_c = (A_W)_c$ be the direction for remaining class c. A target prototype has the form $\mu + sa_c$, with translation $\mu \in \mathbb{R}^d$ and common radius $s \geq 0$. For a retained minibatch B_R of size $m = |B_R|$, define

$$\bar{h} = \frac{1}{m}\sum_{i\in B_R} h_i, \qquad \bar{a} = \frac{1}{m}\sum_{i\in B_R} a_{y_i}, \qquad d_c = a_c - \bar{a}. \tag{8}$$

Although the full simplex is centered, $\bar{a}$ depends on the minibatch class composition. The ideal translation–scale fit minimizes

$$\min_{\mu\in\mathbb{R}^d,\, s\geq 0} \sum_{i\in B_R} \|h_i - \mu - sa_{y_i}\|_2^2. \tag{9}$$

For fixed s, the optimal intercept is $\bar{h} - s\bar{a}$. Define

$$N = \sum_{i\in B_R} \langle h_i - \bar{h}, d_{y_i}\rangle, \qquad D_a = \sum_{i\in B_R} \|d_{y_i}\|_2^2. \tag{10}$$

With $[t]_+ = \max(t, 0)$, the implementation uses the denominator floor $\epsilon = 10^{-12}$:

$$\boxed{s^\star = \left[\frac{N}{\max(D_a, \epsilon)}\right]_+, \qquad t_c = \bar{h} + s^\star d_c.} \tag{11}$$

Equivalently, $t_c = \mu^\star + s^\star a_c$ with $\mu^\star = \bar{h} - s^\star\bar{a}$. This agrees with the least-squares solution when $D_a \geq \epsilon$; the floor stabilizes small denominators. A single-class minibatch gives $N = D_a = 0$ and $s^\star = 0$.

For a fixed classifier, let H stack minibatch features as rows. The target family is

$$\mathcal{T}_{\mathrm{NC}}(W) = \{H \in \mathbb{R}^{m\times d} : h_i = \mu + s(A_W)_{y_i},\ \mu \in \mathbb{R}^d,\ s \geq 0\}. \tag{12}$$

Within each minibatch, samples of the same class share a target. Their concentration toward these targets promotes NC1, while a shared translation and positive scale preserve the class-relative simplex geometry.

3.5 Joint NC1–NC3 Structural Regularizer

Using the fitted prototypes, NCAR defines the retain-side structural regularizer

$$\boxed{\Omega_{\mathrm{NCAR}}(B_R, W) = \frac{1}{m}\sum_{i\in B_R} \|h_i - t_{y_i}\|_2^2.} \tag{13}$$

This is the residual at the stabilized fit; when $D_a \geq \epsilon$, it equals the squared distance to $\mathcal{T}_{\mathrm{NC}}(W)$ divided by m. For a single-class minibatch it reduces to scatter around $\bar{h}$. The dependence on the current feature extractor is implicit.

For interpretation, let $n_c = |\{i \in R : y_i = c\}|$ and $\mu_c = n_c^{-1} \sum_{i \in R: y_i = c} h_i$ be the retained class count and mean. With translation and scale fitted on the full retained set, the residual decomposes as

$$\Omega_{\mathrm{NCAR}} = \underbrace{\frac{1}{|R|} \sum_{i \in R} \|h_i - \mu_{y_i}\|_2^2}_{\text{within-class concentration}} + \underbrace{\frac{1}{|R|} \sum_{c \in \mathcal{C}_R} n_c \|\mu_c - \mu^\star - s^\star (A_W)_c\|_2^2}_{\text{classifier-aligned simplex fitting}}. \tag{14}$$

The first term promotes within-class concentration (NC1), while the second fits the retained class means to the classifier-aligned simplex, coupling NC2 geometry with NC3 alignment. Together, they provide a single retained-feature penalty for NC1–NC3. This full-data decomposition explains the regularizer; during optimization, translation and scale are fitted on each retained minibatch.

At zero full-data residual and positive fitted radius, the RMS NC3 error is $E_3 = r_W / \sqrt{C_R}$ for unit normalized classifier directions, where $r_W = \|A_W - \widehat{W}_R\|_F$ measures the classifier's deviation from simplex geometry. Appendix C gives the derivations and limiting analysis.

3.6 Plug-and-Play Integration with Gradient-Based Unlearning

Given the base objective in Equation 2, NCAR defines

$$\boxed{\mathcal{L}_{\mathrm{base+NCAR}} = \mathcal{L}_{\mathrm{base}} + \beta \Omega_{\mathrm{NCAR}}, \qquad \beta \geq 0.} \tag{15}$$

The coefficient β controls the regularization strength. The base forgetting loss, distillation terms, saliency masks, and parameter-update restrictions are retained. The added gradient is applied through the parameters exposed by the base procedure. Setting $\beta = 0$ recovers the base update under identical sampling and update rules.

At each step, NCAR recomputes A_W from the current classifier, fits translation and scale to the retained minibatch, and adds $\beta \Omega_{\mathrm{NCAR}}$ to the base loss. The targets therefore follow the evolving model.

For methods already consuming retained minibatches, the same batch can be reused. Methods using only forget examples require additional labeled retained-data access. The construction uses the existing feature extractor and classifier and introduces no retrained feature bank or additional trainable network.

Random-subset unlearning. When all original classes remain represented in R, $C_R = C$. The canonical simplex and classifier alignment use all original classes in the original label order.

Full-class unlearning. When entire classes are deleted, $\mathcal{C}_R$ contains only the remaining labels. NCAR rebuilds the simplex with C_R vertices and uses the corresponding classifier columns in the same label order. The target therefore follows the post-deletion class set used in retraining.

4 Experiments

4.1 Experimental Setup

Datasets, models, and unlearning settings. We evaluate NCAR on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) and TinyImageNet (Le & Yang, 2015), using ResNet-18 (He et al., 2016) and ViT-Small/8 (Dosovitskiy et al., 2021). We consider two settings: 10% random-subset unlearning and full-class unlearning. The main text reports both settings for CIFAR-10 with ResNet-18, along with random-subset results for TinyImageNet with ResNet-18 and CIFAR-10 with ViT-Small/8. Additional dataset–architecture–unlearning combinations are provided in Appendix D. Original training and exact retraining follow the same architecture-specific reference-training protocol, with re-

Table 1: **CIFAR-10 with ResNet-18: random-subset unlearning.** 10% random-subset deletion.

Method	UA	MIA	RA	TA	Avg. Gap↓	CKA↑
Retrain	4.20±0.00	75.51±0.00	100.00±0.00	94.82±0.00	0.00±0.00	1.000±0.000
FT	2.49±0.20	84.85±1.07	98.41±0.25	91.73±0.04	2.13±0.04	0.934±0.006
+NCAR	3.78±0.37	84.34±1.22	98.20±0.44	91.39±0.48	**1.89±0.21**	**0.944±0.005**
NegGrad+	2.87±0.44	84.71±1.07	97.97±0.78	91.32±0.73	2.29±0.41	0.929±0.012
+NCAR	2.94±0.74	85.10±0.35	98.52±0.58	91.83±0.99	**1.91±0.29**	**0.948±0.006**
SalUn	10.44±4.03	65.76±3.46	90.86±4.03	86.83±3.41	7.79±3.81	0.767±0.035
+NCAR	4.25±0.19	73.56±1.13	97.66±0.38	91.35±0.62	**1.98±0.33**	**0.905±0.009**
SCRUB	3.52±0.36	84.12±0.61	99.70±0.15	93.04±0.32	0.92±0.15	0.968±0.007
+NCAR	3.19±0.22	84.29±0.58	99.81±0.06	93.38±0.18	**0.88±0.10**	**0.971±0.007**
Bad Teacher	3.84±1.09	41.55±6.48	99.95±0.00	93.90±0.22	0.62±0.20	0.902±0.021
+NCAR	4.10±0.45	39.47±2.17	99.97±0.02	94.01±0.03	**0.38±0.10**	**0.949±0.001**
RL	4.09±0.20	57.34±0.94	99.03±0.27	92.03±0.27	1.31±0.11	0.834±0.063
+NCAR	3.76±0.78	55.69±1.61	99.46±0.05	92.91±0.11	**0.99±0.26**	**0.941±0.002**
UGradSL+	2.65±0.34	67.61±0.58	99.82±0.07	93.36±0.25	**1.06±0.03**	0.895±0.010
+NCAR	2.66±0.37	67.52±0.71	99.81±0.07	93.36±0.26	**1.06±0.03**	**0.896±0.010**

training initialized from scratch for ResNet-18 or from the same ImageNet initialization for ViT-Small/8 and trained only on the retained set R. Detailed data splitting, optimization, preprocessing, architecture, and selection settings are given in Appendix A.

Compared methods. We compare fine-tuning (FT), NegGrad+ (Kurmanji et al., 2023), SalUn (Fan et al., 2024), SCRUB (Kurmanji et al., 2023), Bad Teacher (Chundawat et al., 2023), random labeling (RL) (Graves et al., 2021), and UGradSL+ (Di et al., 2026), each with and without NCAR. Retrain denotes the model trained on R using the architecture-specific reference-training protocol.

Evaluation, selection, and reporting. Unlearning accuracy (UA) is the percentage of forget examples classified incorrectly; retained accuracy (RA) and test accuracy (TA) are percentage accuracies on R and the original test set T, respectively. Avg. Gap is the mean absolute deviation from Retrain over UA, RA, and TA, measured in percentage points. Centered kernel alignment (CKA) (Kornblith et al., 2019) measures retained-feature similarity to Retrain. We report the membership-inference attack (MIA) score (Shokri et al., 2017) separately as the percentage of forget examples classified as members by the entropy-based logistic-regression attack described in Appendix B. Results are mean ± sample standard deviation over three unlearning seeds, except for the single-seed component ablation and single-run overhead profiles; Avg. Gap is computed per seed before aggregation.

For each method and seed, we select the base hyperparameters and checkpoint by minimizing Avg. Gap. The +NCAR variant fixes the selected base hyperparameters and tunes only β. MIA, CKA, and NC1–NC4 are evaluated at the selected checkpoints but are not used for model selection. Details of model selection and evaluation metrics are provided in Appendices A and B, respectively.

4.2 Unlearning Results

CIFAR-10 with ResNet-18. Tables 1 and 2 report results under 10% random-subset and full-class unlearning, respectively. In the random-subset setting, adding NCAR consistently increases CKA while reducing or matching Avg. Gap across the evaluated base methods. Under full-class unlearning, NCAR lowers Avg. Gap and increases CKA for all evaluated methods. Together, the two settings show that the retain-side structural regularization improves agreement with retraining in both predictive behavior and retained representation structure, although the magnitude of the improvement varies across the underlying unlearning procedures.

Table 2: **CIFAR-10 with ResNet-18: full-class unlearning.** Class 0 removed.

Method	UA	MIA	RA	TA	Avg. Gap↓	CKA↑
Retrain	100.00±0.00	14.53±0.00	100.00±0.00	85.76±0.00	0.00±0.00	1.000±0.000
FT	100.00±0.00	17.14±2.25	98.04±0.20	83.02±0.43	1.57±0.21	0.912±0.007
+NCAR	100.00±0.00	8.24±1.20	98.78±0.18	83.59±0.12	**1.13±0.09**	**0.954±0.001**
NegGrad+	100.00±0.00	46.41±7.82	97.56±0.05	82.35±0.14	1.95±0.03	0.834±0.055
+NCAR	100.00±0.00	44.50±3.93	98.72±0.62	83.46±0.43	**1.19±0.35**	**0.947±0.007**
SalUn	100.00±0.00	1.53±0.35	99.95±0.02	85.41±0.15	0.13±0.05	0.965±0.003
+NCAR	100.00±0.00	0.46±0.28	99.99±0.01	85.68±0.11	**0.04±0.03**	**0.980±0.003**
SCRUB	100.00±0.00	17.02±5.75	99.92±0.13	85.29±0.35	0.18±0.16	0.749±0.037
+NCAR	100.00±0.00	20.35±7.04	99.98±0.03	85.68±0.39	**0.10±0.09**	**0.964±0.012**
Bad Teacher	99.96±0.04	0.04±0.04	100.00±0.00	86.14±0.04	0.14±0.03	0.977±0.001
+NCAR	99.94±0.10	0.06±0.10	100.00±0.00	86.06±0.02	**0.12±0.03**	**0.985±0.001**
RL	99.78±0.26	0.03±0.01	98.99±0.10	83.43±0.07	1.19±0.10	0.935±0.007
+NCAR	99.84±0.27	0.02±0.04	99.66±0.49	84.77±0.96	**0.50±0.47**	**0.968±0.009**
UGradSL+	97.95±0.16	60.14±0.82	97.25±0.23	81.52±0.38	3.02±0.16	0.916±0.007
+NCAR	97.78±0.10	59.76±0.44	98.17±0.16	82.56±0.16	**2.42±0.07**	**0.949±0.005**

Table 3: **TinyImageNet with ResNet-18.** 10% random-subset unlearning.

Method	UA	MIA	RA	TA	Avg. Gap↓	CKA↑
Retrain	38.52±0.00	23.33±0.00	99.98±0.00	61.87±0.00	0.00±0.00	1.000±0.000
FT	41.38±0.95	54.99±0.44	70.07±2.70	46.48±0.66	16.05±1.28	0.554±0.002
+NCAR	37.83±0.90	47.76±0.67	75.60±2.60	49.51±0.28	**12.48±0.85**	**0.599±0.007**
NegGrad+	40.05±4.05	54.31±1.35	70.45±7.31	47.57±2.92	15.47±4.43	0.569±0.028
+NCAR	37.60±0.64	47.09±0.93	78.04±2.98	50.74±0.77	**11.33±0.90**	**0.612±0.007**
SalUn	52.53±3.18	21.59±2.80	87.02±1.45	50.40±0.58	12.81±0.71	0.605±0.006
+NCAR	50.21±4.36	18.50±5.46	90.57±2.65	51.68±0.26	**10.43±0.68**	**0.649±0.005**
SCRUB	42.57±1.61	39.94±1.00	98.96±0.30	55.50±1.24	3.81±1.04	0.744±0.008
+NCAR	42.85±1.23	39.90±1.16	99.14±0.04	55.85±0.66	**3.73±0.62**	**0.748±0.005**
Bad Teacher	38.98±0.86	0.67±0.19	99.94±0.01	56.63±0.25	1.96±0.33	0.763±0.003
+NCAR	38.77±0.43	0.12±0.07	99.98±0.00	57.42±0.26	**1.58±0.04**	**0.774±0.003**
RL	54.46±1.93	19.32±2.26	88.07±1.28	50.50±0.21	13.07±0.45	0.611±0.009
+NCAR	48.83±6.64	21.70±7.12	89.40±6.14	52.60±0.58	**10.05±0.33**	**0.649±0.011**
UGradSL+	42.10±0.45	56.30±0.22	99.38±0.07	51.62±0.31	4.81±0.27	**0.723±0.002**
+NCAR	42.20±0.43	56.07±0.13	99.41±0.07	51.70±0.27	**4.81±0.25**	**0.723±0.002**

Dataset extension. Table 3 reports 10% random-subset unlearning results on TinyImageNet with ResNet-18. Across the evaluated methods, adding NCAR reduces or matches Avg. Gap while increasing CKA, preserving the behavior observed on CIFAR-10 on a different dataset. Additional results are reported in Appendix D.

Architecture extension. Table 4 reports 10% random-subset unlearning results on CIFAR-10 with ViT-Small/8. Adding NCAR lowers Avg. Gap and increases CKA across the evaluated base methods, showing that the improvement in retraining agreement is also observed with a transformer-based vision architecture. Additional results are reported in Appendix D.

Table 4: **CIFAR-10 with ViT-Small/8.** 10% random-subset unlearning.

Method	UA	MIA	RA	TA	Avg. Gap↓	CKA↑
Retrain	5.09±0.00	87.40±0.00	99.95±0.00	95.04±0.00	0.00±0.00	1.000±0.000
FT	4.38±0.43	81.84±0.09	98.02±0.22	93.25±0.38	1.48±0.17	0.857±0.002
+NCAR	4.39±0.15	80.02±1.16	98.20±0.08	93.61±0.37	**1.29±0.19**	**0.871±0.002**
NegGrad+	4.92±0.32	82.67±0.81	98.27±0.16	93.22±0.36	1.24±0.11	0.891±0.002
+NCAR	5.18±0.19	83.33±0.18	98.60±0.18	93.67±0.18	**0.96±0.10**	**0.923±0.002**
SalUn	3.30±0.11	67.99±0.41	97.17±0.33	93.86±0.25	1.92±0.16	0.843±0.007
+NCAR	3.05±0.05	68.59±0.42	97.34±0.13	94.04±0.06	**1.88±0.08**	**0.865±0.004**
SCRUB	3.53±0.29	89.61±0.40	98.73±0.28	94.43±0.13	1.13±0.05	0.920±0.009
+NCAR	3.59±0.29	90.14±0.14	98.85±0.30	94.49±0.38	**1.05±0.11**	**0.922±0.005**
Bad Teacher	1.71±0.37	31.90±0.82	99.42±0.18	94.67±0.18	1.43±0.00	0.850±0.004
+NCAR	1.74±0.30	33.07±0.70	99.40±0.15	94.73±0.20	**1.40±0.02**	**0.859±0.006**
RL	4.76±0.08	35.63±3.34	98.60±0.26	93.84±0.23	0.96±0.13	0.723±0.066
+NCAR	4.57±0.24	36.10±2.46	98.72±0.09	94.03±0.03	**0.92±0.11**	**0.734±0.070**
UGradSL+	5.08±0.47	39.41±1.22	99.14±0.08	93.97±0.05	0.74±0.06	0.768±0.006
+NCAR	5.12±0.24	39.99±1.14	99.15±0.03	94.02±0.04	**0.66±0.04**	**0.804±0.003**

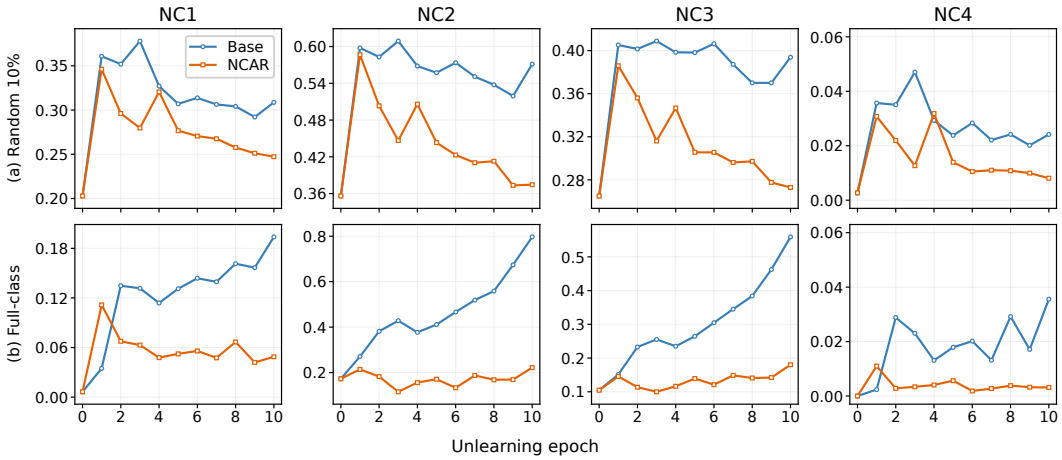


Figure 2: **Neural Collapse during unlearning.** NC1–NC4 trajectories for NegGrad+ with and without NCAR on CIFAR-10. (a) ViT-Small/8 with 10% random-subset unlearning. (b) ResNet-18 with full-class unlearning. Values closer to zero indicate closer agreement with the corresponding ideal Neural Collapse property.

4.3 Structural Analysis

Figure 2 tracks NC1–NC4 for NegGrad+ with and without NCAR on CIFAR-10: 10% random-subset unlearning with ViT-Small/8 in panel (a), and full-class unlearning with ResNet-18 in panel (b). The metrics initially increase as unlearning begins. The base trajectories remain displaced from the ideal-NC reference, particularly under full-class deletion, whereas subsequent trajectories with NCAR lie closer to zero. This is consistent with reduced structural drift during the combined unlearning update, rather than evidence isolating the forgetting loss as its cause. Appendix E complements these trajectories with NC diagnostics measured at the Avg. Gap-selected checkpoints.

4.4 Ablation Study

We remove within-class concentration (WC), simplex ETF geometry (ETF), or classifier alignment (CA) from Full NCAR while retaining the other two components. Table 5 reports these leave-

Table 5: **NCAR component ablation on CIFAR-10 with ViT-Small/8.** NegGrad+ is the common base objective; full-class unlearning removes class 0.

	Components			Random 10%				Full-class			
Variant	WC	ETF	CA	RA	TA	Avg. Gap↓	CKA↑	RA	TA	Avg. Gap↓	CKA↑
NegGrad+	×	×	×	98.29	93.26	1.17	0.889	93.92	80.73	3.68	0.784
−WC	×	✓	✓	98.37	93.24	1.30	0.903	94.02	81.00	3.55	0.755
−ETF	✓	×	✓	98.77	93.42	0.95	0.909	93.76	80.65	3.77	0.781
−CA	✓	✓	×	98.38	93.03	1.29	0.922	93.64	80.52	3.85	0.780
Full NCAR	✓	✓	✓	98.81	93.84	**0.85**	**0.923**	96.71	82.63	**2.17**	**0.925**

one-out comparisons using NegGrad+ on CIFAR-10 with ViT-Small/8 under both deletion settings; Appendix F defines the variants.

Full NCAR achieves the lowest Avg. Gap and highest CKA in both settings. Removing any component substantially weakens the full-class gains. Under random-subset deletion, removing CA worsens Avg. Gap while leaving CKA nearly unchanged, showing that the component's effect is not fully captured by feature similarity. These leave-one-out results support the complete construction.

5 Discussion

The results suggest that the gap between approximate unlearning and retraining is not fully characterized by predictive metrics alone. NCAR addresses this gap from the retained representation side by regularizing class-conditional features toward a classifier-aligned Neural Collapse geometry while leaving the base forgetting objective unchanged. Across the reported experiments, this structural constraint improves representation similarity to retraining without degrading the retraining-relative predictive criterion, indicating that predictive agreement and retained representation structure can be improved together.

The structural analyses further clarify the role of the proposed regularization. NCAR does not treat ideal Neural Collapse as a substitute for retraining; instead, the NC geometry provides an idealized structural prior constructed without a retrained model. The trajectories show changes in the representation–classifier geometry during unlearning, with subsequent values closer to the ideal-NC reference when NCAR is applied. The ablation results further indicate that within-class concentration, ETF geometry, and classifier alignment influence different aspects of the retained representation, motivating their joint use in the full objective.

The current formulation is limited to supervised classification with labeled retained data and a linear classification head. Applying NCAR also requires access to retained samples during unlearning and introduces additional computation for classifier alignment and retained-feature regularization. Moreover, because NCAR preserves the forgetting mechanism of the base method, its structural metrics should not be interpreted as a certificate of forgetting; forgetting quality must still be assessed using the behavioral and privacy metrics. Runtime and memory overhead are reported in Appendix G.

6 Conclusion

We introduced Neural Collapse Alignment Regularization (NCAR), a retain-side structural regularizer for gradient-based machine unlearning. NCAR uses classifier-aligned Neural Collapse geometry to constrain retained representations while preserving the forgetting mechanism of the underlying unlearning method. Our results show that explicitly regularizing the retained representation–classifier structure can bring approximate unlearning closer to retraining in both predictive behavior and representation structure. NCAR therefore complements existing forgetting and retention objectives with an explicit structural target for retained representations.

REPRODUCIBILITY STATEMENT

We provide the implementation and evaluation scripts in the supplementary code package. The appendix documents the reference-training setup, model-selection procedure, evaluation metrics, ablation objectives, and computational overhead.

AI USE STATEMENT

Generative AI was used solely for language assistance, including English translation, grammar correction, and stylistic refinement of the manuscript. It was not used to generate experimental results or to make methodological or experimental decisions. All AI-assisted text was reviewed by the authors. The authors take full responsibility for the final content of the manuscript.

REFERENCES

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine Unlearning. In *IEEE Symposium on Security and Privacy*, 2021. URL https://arxiv.org/abs/1912.03817.

Yinzhi Cao and Junfeng Yang. Towards Making Systems Forget with Machine Unlearning. In *IEEE Symposium on Security and Privacy*, 2015. URL https://www.cs.columbia.edu/~junfeng/papers/unlearning/.

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, pp. 7210–7217, 2023. doi: 10.1609/aaai.v37i6.25879. URL https://ojs.aaai.org/index.php/AAAI/article/view/25879.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

Zonglin Di, Zhaowei Zhu, Jinghan Jia, Jiancheng Liu, Zafar Takhirov, Bo Jiang, Yuanshun Yao, Sijia Liu, and Yang Liu. Label Smoothing Improves Machine Unlearning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2d19debb2ea6a4620f7fe67372cca800-Abstract-Conference.html.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.11929.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ec4d2e436794d1bf55ca83f5ebb31887-Abstract-Conference.html.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast Machine Unlearning without Retraining through Selective Synaptic Dampening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12043–12051, 2024. doi: 10.1609/aaai.v38i11.29092. URL https://ojs.aaai.org/index.php/AAAI/article/view/29092.

Yichen Gao, Altay Unal, Akshay Rangamani, and Zhihui Zhu. An Illusion of Unlearning? Assessing Machine Unlearning Through Internal Representations. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 4114–4122. PMLR, 2026. URL https://proceedings.mlr.press/v300/gao26e.html.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. URL https://arxiv.org/abs/1911.04933.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac Machine Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11516–11524, 2021. doi: 10.1609/aaai.v35i13.17371. URL https://ojs.aaai.org/index.php/AAAI/article/view/17371.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified Data Removal from Machine Learning Models. In *Proceedings of the International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 2020. URL https://proceedings.mlr.press/v119/guo20c.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. URL https://openaccess.thecvf.com/CVPR2016.py.

Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified Gradient-Based Machine Unlearning with Remain Geometry Enhancement. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0831. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2e622ac74f66df03b686a12e2e0e4424-Abstract-Conference.html.

Sangamesh Kodge, Gobinda Saha, and Kaushik Roy. Deep Unlearning: Fast and Efficient Gradient-free Class Forgetting. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=BmI5p6wBi0.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of Neural Network Representations Revisited. In *Proceedings of the International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019. URL https://proceedings.mlr.press/v97/kornblith19a.html.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/cifar.html.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards Unbounded Machine Unlearning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/062d711fb777322e2152435459e6e9d9-Abstract-Conference.html.

Anjie Le, Can Peng, Yuyuan Liu, and J. Alison Noble. POUR: A Provably Optimal Method for Unlearning Representation via Neural Collapse. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10273–10282, 2026. URL https://openaccess.thecvf.com/CVPR2026?day=2026-06-05.

Ya Le and Xuan Yang. Tiny ImageNet Visual Recognition Challenge. Cs231n course project report, Stanford University, 2015. URL https://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf.

Hong kyu Lee, Qiuchen Zhang, Carl Yang, Jian Lou, and Li Xiong. Contrastive Unlearning: A Contrastive Approach to Machine Unlearning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 7464–7472, 2025. doi: 10.24963/ijcai.2025/830. URL https://www.ijcai.org/proceedings/2025/830.

Evan Markou, Thalaiyasingam Ajanthan, and Stephen Gould. Guiding Neural Collapse: Optimising Towards the Nearest Simplex Equiangular Tight Frame. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1121. URL https://papers.nips.cc/paper_files/paper/2024/hash/3eb660055cdcdc9a545a0b16c1eff80d-Abstract-Conference.html.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020. doi: 10.1073/pnas.2015509117. URL https://doi.org/10.1073/pnas.2015509117.

Gaurav Patel and Qiang Qiu. Learning to Unlearn while Retaining: Combating Gradient Conflicts in Machine Unlearning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4211–4221, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Patel_Learning_to_Unlearn_while_Retaining_Combating_Gradient_Conflicts_in_Machine_ICCV_2025_paper.html.

Laiqiao Qin, Tianqing Zhu, Linlin Wang, and Wanlei Zhou. Machine Unlearning on Pre-trained Models by Residual Feature Alignment Using LoRA. *IEEE Transactions on Dependable and Secure Computing*, 2026. doi: 10.1109/TDSC.2026.3658545. URL https://arxiv.org/abs/2411.08443.

Peter H. Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1):1–10, 1966. doi: 10.1007/BF02289451. URL https://doi.org/10.1007/BF02289451.

Seonguk Seo, Dongwan Kim, and Bohyung Han. Revisiting Machine Unlearning with Dimensional Alignment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3206–3215, 2025. URL https://openaccess.thecvf.com/content/WACV2025/html/Seo_Revisiting_Machine_Unlearning_with_Dimensional_Alignment_WACV_2025_paper.html.

Shaofei Shen, Chenhao Zhang, Yawen Zhao, Alina Bialkowski, Weitong Chen, and Miao Xu. Label-Agnostic Forgetting: A Supervision-Free Unlearning in Deep Models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/ee1f0da706829d7f198eac0edaacc338-Abstract-Conference.html.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks against Machine Learning Models. In *IEEE Symposium on Security and Privacy*, 2017. URL https://arxiv.org/abs/1610.05820.

Zhihui Zhu, Tianyu Ding, Jinxin Zhou, Xiao Li, Chong You, Jeremias Sulam, and Qing Qu. A Geometric Analysis of Neural Collapse with Unconstrained Features. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29820–29834, 2021. URL https://proceedings.nips.cc/paper_files/paper/2021/hash/f92586a25bb3145facd64ab20fd554ff-Abstract.html.

A Implementation and Selection Details

A.1 Notation

Table 6 summarizes the notation used in the implementation and evaluation appendices.

A.2 Reference Training and Data Splits

For CIFAR-10, CIFAR-100, and TinyImageNet, a class-stratified 10% validation partition is first held out from the training data; it is not used for unlearning hyperparameter tuning, checkpoint selection, or reported metrics, and is reserved only for sanity checks of the original and reference models. The resulting development set D is used to construct the deletion request. Random-subset unlearning selects 10% of D using the split seed, while full-class unlearning assigns all examples of the designated class in D to F and all other examples to R.

Original models are trained on D, whereas exact-retraining reference models are trained only on R. Reference training uses 500 epochs, batch size 256, SGD with momentum 0.9, weight decay 5×10^{-4}, and MultiStepLR with milestones at epochs 166 and 333 and decay factor 0.1. ResNet-18

Table 6: **Notation used in the method and appendix.**

Symbol	Meaning
D	Development training set after holding out the validation partition
R, F	Retained and forget subsets of D
T	Original test set
$\mathcal{C}_R, C_R$	Remaining class set and its cardinality
B_R, m	Retained minibatch and its size
$h_i \in \mathbb{R}^d$	Feature representation of sample i used by the linear classifier
W, W_R	Linear classifier and its remaining-class columns
V	Canonical simplex ETF used by NCAR
$M_{h,t}, M_{\mathrm{rt}}$	Candidate checkpoint model and exact-retraining reference model
$\mathcal{H}, \mathcal{B}$	Evaluated base-method configurations and NCAR coefficient values
$\mathcal{K}_h, \mathcal{K}_\beta$	Evaluated checkpoint epochs for the base and NCAR stages
G_{acc}	Retraining-relative Avg. Gap selection criterion
C_{base}	Per-update computational cost of the base unlearning method
β	NCAR regularization coefficient
ϵ	Numerical floor 10^{-12} for normalization and scale fitting

is trained from scratch with learning rate 0.05. The ResNet-18 implementation uses a CIFAR-style 3×3 stride-1 stem without max pooling and a 512-dimensional penultimate feature.

The second architecture is ViT-Small/8, implemented as a 12-block Vision Transformer with embedding dimension 384, six attention heads, multilayer-perceptron (MLP) expansion ratio 4, and patch size 8. It is initialized from an ImageNet-pretrained ViT-Small/8 checkpoint (Deng et al., 2009); positional embeddings are resized to the target image resolution and the classification head is reinitialized. Reference training uses learning rate 10^{-4} with the same SGD, batch-size, weight-decay, and MultiStepLR settings described above.

CIFAR images use random cropping with four-pixel padding and random horizontal flipping during training. TinyImageNet uses 64×64 images with the same random-crop padding and horizontal flip. Dataset-specific channel normalization is applied inside the model. Evaluation uses deterministic, non-augmented inputs.

A.3 NCAR Optimization

At each unlearning update, NCAR computes the classifier-aligned target from the current classifier, fits the shared translation and scale using a labeled retained minibatch, and adds $\beta\Omega_{\mathrm{NCAR}}$ to the base objective. For Full NCAR, gradients propagate through the target construction and the retained-feature residual. The retained minibatch used by the evaluated base method is reused for the NCAR computation.

Unlearning candidates are run for the configured epoch budget, with checkpoints evaluated after each epoch for selection. The implementation supports up to 10 unlearning epochs per candidate. The optimization seed controls unlearning stochasticity independently of the split seed that defines R and F.

A.4 Candidate Selection

For a base method, let $\mathcal{H}$ denote its evaluated hyperparameter configurations, $\mathcal{K}_h$ the evaluated epochs under configuration h, and $M_{h,t}$ the corresponding model at epoch t. With M_{rt} denoting the reference model, the selected base checkpoint is

$$(h^*, t^*) \in \arg\min_{h \in \mathcal{H},\, t \in \mathcal{K}_h} G_{\mathrm{acc}}(M_{h,t}, M_{\mathrm{rt}}), \tag{16}$$

where G_{acc} is defined in Equation 19. Ties are resolved in favor of the earlier epoch.

Let $\mathcal{B}$ be the candidate NCAR coefficients, $\mathcal{K}_\beta$ the epochs evaluated for β, and $M_{h^*,\beta,t}$ the corresponding model. The +NCAR stage fixes h^* and selects

$$(\beta^*, t^*_\beta) \in \arg\min_{\beta\in\mathcal{B},\, t\in\mathcal{K}_\beta} G_{\mathrm{acc}}(M_{h^*,\beta,t}, M_{\mathrm{rt}})\,. \tag{17}$$

Both stages use the same retraining-relative selection criterion. MIA, CKA, and NC diagnostics are not used for selection.

B Evaluation Metrics

B.1 Behavioral Metrics and Average Gap

Let $\widehat{y}_i$ denote the model prediction. The reported percentages are

$$\mathrm{UA} = \frac{100}{|F|}\sum_{i\in F}\mathbf{1}[\widehat{y}_i \neq y_i], \qquad \mathrm{RA} = \frac{100}{|R|}\sum_{i\in R}\mathbf{1}[\widehat{y}_i = y_i], \tag{18}$$

with TA defined analogously on the complete original test set T. In full-class experiments, TA still uses the original test population and the original classifier output space.

For MIA, the implementation uses predictive entropy as a one-dimensional attack feature. A class-balanced logistic-regression attack is fitted with retained examples R as members and test examples from the remaining classes, $T_R = \{(x, y) \in T : y \in \mathcal{C}_R\}$, as nonmembers. The reported MIA value is 100 times the fraction of forget examples F classified as members. The same attack construction is used for the candidate and retraining reference; the attack random seed is fixed independently of the unlearning seed.

For a candidate model M and reference M_{rt}, the formal definition is

$$G_{\mathrm{acc}}(M, M_{\mathrm{rt}}) = \frac{1}{3}\sum_{q\in\{\mathrm{UA},\mathrm{RA},\mathrm{TA}\}} |q(M) - q(M_{\mathrm{rt}})|. \tag{19}$$

Avg. Gap is measured in percentage points and excludes MIA. It is computed within each paired seed before aggregation across seeds.

B.2 Retained-Sample Similarity

Let $n = |R|$ and let $H, H^{\mathrm{rt}} \in \mathbb{R}^{n\times d}$ contain features of the same ordered retained examples as rows. Set $X = J_n H$ and $Y = J_n H^{\mathrm{rt}}$, where $J_n = I_n - \mathbf{1}_n\mathbf{1}_n^\top/n$. Linear CKA (Kornblith et al., 2019) is

$$\mathrm{CKA}(X, Y) = \frac{\|X^\top Y\|_F^2}{\|X^\top X\|_F\,\|Y^\top Y\|_F}. \tag{20}$$

The implementation evaluates CKA in double precision on the complete retained set using deterministic evaluation inputs. A numerically zero denominator returns zero by implementation convention; this is a degenerate case.

B.3 Neural Collapse Diagnostics

Following the NC properties characterized by Papyan et al. (2020), we compute the diagnostics below on the complete retained set over the remaining classes. Let μ_c denote the class mean and

$$\mu_G = \frac{1}{C_R}\sum_{c\in\mathcal{C}_R}\mu_c, \tag{21}$$

and define

$$\Sigma_W = \frac{1}{|R|}\sum_{i\in R}(h_i - \mu_{y_i})(h_i - \mu_{y_i})^\top, \qquad \Sigma_B = \frac{1}{C_R}\sum_{c\in\mathcal{C}_R}(\mu_c - \mu_G)(\mu_c - \mu_G)^\top. \tag{22}$$

The NC1 diagnostic is

$$E_1 = \frac{1}{C_R}\,\mathrm{tr}\left(\Sigma_W\Sigma_B^\dagger\right), \tag{23}$$

Here Σ_W is sample-weighted, whereas μ_G and Σ_B are class-balanced. The pseudoinverse retains eigenvalues $\lambda_j > \max(10^{-12}, 10^{-10}\lambda_{\max})$, where $\lambda_{\max}$ is the largest eigenvalue of Σ_B.

For NC2 and NC3, use the same class ordering as in the method and define the row-stacked matrices

$$M_R = [(\mu_c - \mu_G)^\top]_{c\in\mathcal{C}_R}, \qquad W_{R,\mathrm{row}} = \bar{W}_R^\top, \qquad P_R = I_{C_R} - \frac{\mathbf{1}_{C_R}\mathbf{1}_{C_R}^\top}{C_R}. \tag{24}$$

Both M_R and $W_{R,\mathrm{row}}$ lie in $\mathbb{R}^{C_R\times d}$. For nonzero A, define $\mathcal{N}_F(A) = A/\|A\|_F$. NC2 is the scale-free feature-Gram error

$$E_2 = \|\mathcal{N}_F(M_R M_R^\top) - \mathcal{N}_F(P_R)\|_F. \tag{25}$$

This normalizes the complete Gram matrix, not each class mean, and therefore retains relative class-mean norm differences.

Let $\mathcal{N}_r(\cdot)$ denote row-wise ℓ_2 normalization. NC3 uses the root-mean-square (RMS) convention:

$$E_3 = \frac{1}{\sqrt{C_R}}\|\mathcal{N}_r(M_R) - \mathcal{N}_r(W_{R,\mathrm{row}})\|_F. \tag{26}$$

This is the root mean square of the class-wise feature–classifier directional discrepancies. The same metric is analyzed in Appendix C.3.

Finally, NC4 is the disagreement rate, on a 0–1 scale, between the remaining-class linear head and nearest-class-center predictions:

$$E_4 = \frac{1}{|R|}\sum_{i\in R}\mathbf{1}\left[\arg\max_{c\in\mathcal{C}_R} z_{ic} \neq \arg\min_{c\in\mathcal{C}_R}\|h_i - \mu_c\|_2^2\right]. \tag{27}$$

C Geometric Characterization

This section provides the geometric justification for the structural construction in Section 3. We first show that the canonical simplex can be embedded in the retained feature space and aligned with the classifier without changing its NC2 geometry. We then derive the translation–scale fit and the residual decomposition used by NCAR. Finally, using the diagnostics in Appendix B.3, we characterize the conditional zero-residual limit of NC1–NC3.

C.1 Simplex Existence and Classifier Alignment

A unit simplex with C_R vertices has rank

$$r = C_R - 1, \tag{28}$$

so the condition $C_R \le d + 1$ in Section 3.2 is sufficient to embed the simplex in the d-dimensional feature space.

Write

$$V = O_0 S_0, \tag{29}$$

where $O_0 \in \mathbb{R}^{d\times r}$ has orthonormal columns, $S_0 \in \mathbb{R}^{r\times C_R}$ satisfies $S_0\mathbf{1}_{C_R} = 0$, and

$$S_0^\top S_0 = V^\top V. \tag{30}$$

Every orthogonal transformation of the simplex can be represented by some $P \in \mathbb{R}^{d\times r}$ as

$$QV = PS_0, \qquad P^\top P = I_r, \tag{31}$$

and conversely any such P can be extended to an orthogonal matrix Q satisfying $QO_0 = P$. Hence the classifier-alignment problem in Equation 6 is equivalent to

$$\min_{P^\top P = I_r}\left\|PS_0 - \widehat{W}_R\right\|_F^2. \tag{32}$$

If

$$\widehat{W}_R S_0^\top = U_r\Sigma_r Z_r^\top \tag{33}$$

is a thin SVD, a reduced Procrustes solution is $P = U_r Z_r^\top$ (Schönemann, 1966).

Because the alignment is orthogonal,

$$A_W^\top A_W = V^\top V. \tag{34}$$

Therefore, classifier alignment changes only the orientation of the canonical simplex in feature space. The equal class-direction norms and pairwise angles defining the NC2 geometry are preserved exactly.

C.2 Translation–Scale Fit and Residual Decomposition

For fixed directions $a_{y_i} = (A_W)_{y_i}$, consider the ideal least-squares problem in Equation 9. For fixed s, the optimal intercept is $\mu = \bar{h} - s\bar{a}$. Substitution gives

$$\begin{aligned}\sum_{i \in B_R} &\|(h_i - \bar{h}) - s(a_{y_i} - \bar{a})\|_2^2 \\ &= \sum_{i \in B_R} \|h_i - \bar{h}\|_2^2 - 2sN + s^2 D_a,\end{aligned} \tag{35}$$

where N and D_a are defined in Equation 10. For $D_a > 0$, the exact nonnegative minimizer is $[N/D_a]_+$. The implemented fit in Equation 11 instead uses $\max(D_a, \epsilon)$, with $\epsilon = 10^{-12}$; it agrees with the exact minimizer when $D_a \geq \epsilon$. This is a denominator floor, not an additive ridge term. For $D_a = 0$, the centered directions vanish, $N = 0$, and the implementation returns $s^\star = 0$.

The residual decomposition holds for any shared class prototypes, including those from the stabilized fit. For a fit on the complete retained set, let $p_c = \mu^\star + s^\star (A_W)_c$. Expanding

$$h_i - p_{y_i} = (h_i - \mu_{y_i}) + (\mu_{y_i} - p_{y_i}) \tag{36}$$

and using $\sum_{i \in R: y_i = c} (h_i - \mu_c) = 0$ removes the cross term, yielding

$$\begin{aligned}\Omega_{\mathrm{NCAR}}(R, W) = &\frac{1}{|R|} \sum_{i \in R} \|h_i - \mu_{y_i}\|_2^2 \\ &+ \frac{1}{|R|} \sum_{c \in \mathcal{C}_R} n_c \|\mu_c - \mu^\star - s^\star (A_W)_c\|_2^2.\end{aligned} \tag{37}$$

This recovers Equation 14: the first term is within-class scatter, and the second measures class-mean fitting to the classifier-aligned simplex.

C.3 Neural Collapse Geometry in the Zero-Residual Limit

We analyze the full-data zero-residual limit using Equation 37. Fix the retained sample set, the current classifier, and its selected alignment A_W. Assume all remaining classes occur, normalized centered classifier directions have unit norm, and the fitted radius satisfies $s^\star \geq s_{\min} > 0$ for a fixed $s_{\min}$. These conditions define the full-data limiting analysis. Consider

$$\Omega_{\mathrm{NCAR}} \to 0. \tag{38}$$

The within-class component of the residual then satisfies

$$\frac{1}{|R|} \sum_{i \in R} \|h_i - \mu_{y_i}\|_2^2 \to 0, \tag{39}$$

so the within-class covariance vanishes. The class-mean fitting term also vanishes; together with the radius lower bound, this keeps the nonzero between-class eigenvalues separated from zero. The pseudoinverse in NC1 therefore remains bounded, giving

$$E_1 \to 0. \tag{40}$$

At the same time, the class-mean fitting component satisfies

$$\frac{1}{|R|} \sum_{c \in \mathcal{C}_R} n_c \, \|\mu_c - \mu^\star - s^\star (A_W)_c\|_2^2 \to 0. \tag{41}$$

Thus the retained class means approach a translated and scaled copy of the classifier-aligned simplex. Since the orthogonal alignment preserves

$$A_W^\top A_W = V^\top V, \tag{42}$$

the normalized class-mean Gram matrix approaches the canonical simplex Gram matrix, giving

$$E_2 \to 0. \tag{43}$$

The same limit makes the normalized centered class-mean directions approach the columns of A_W. Applying the RMS NC3 definition in Equation 26, and using invariance of the Frobenius norm under transposition between the row and column conventions, gives

$$E_3 \to \frac{1}{\sqrt{C_R}} \left\| A_W - \widehat{W}_R \right\|_F = \frac{r_W}{\sqrt{C_R}}, \qquad r_W = \|A_W - \widehat{W}_R\|_F. \tag{44}$$

Because Q_W is selected by orthogonal Procrustes alignment,

$$r_W = \min_{Q^\top Q = I} \|QV - \widehat{W}_R\|_F, \tag{45}$$

so r_W is the smallest discrepancy between the normalized remaining-class classifier directions and an orthogonal orientation of the canonical simplex. The NC phenomenon motivates this target (Papyan et al., 2020); r_W quantifies the remaining discrepancy determined by the current classifier geometry.

Under these conditions, as the full-data NCAR residual approaches zero,

$$E_1 \to 0, \qquad E_2 \to 0, \qquad E_3 \to \frac{r_W}{\sqrt{C_R}}, \tag{46}$$

with the remaining NC3 discrepancy correspondingly small when the current classifier is close to the Neural Collapse simplex geometry.

D Additional Experimental Results

D.1 Dataset Extension

Table 7 extends the dataset comparison to CIFAR-100 with ResNet-18 under 10% random-subset unlearning. The original-model and Retrain checkpoints and the data split are fixed across runs.

D.2 Architecture Extension

Table 8 extends the transformer evaluation to full-class unlearning on CIFAR-10 with ViT-Small/8. Across both additional settings, the reported mean Avg. Gap decreases and CKA increases for all tested base methods, although the improvement varies by method. These comparisons extend the evidence for the same retain-side structural construction beyond the main settings.

E Extended Neural Collapse Analysis

Table 9 complements the trajectories in Figure 2 with NC1–NC4 measured at the Avg. Gap-selected checkpoints for CIFAR-10 10% random-subset unlearning with ResNet-18 and ViT-Small/8. The selection protocol is given in Section 4.1. Lower values indicate closer agreement with the corresponding Neural Collapse property.

Across the two architectures and evaluated base methods, adding NCAR generally reduces the NC diagnostics, indicating that its structural effect extends beyond the representative trajectories in Figure 2. The results show that NCAR moves the retained representation–classifier geometry at these checkpoints toward the Neural Collapse structure targeted by the regularizer, while the magnitude of the change depends on the underlying unlearning procedure.

Table 7: **CIFAR-100 with ResNet-18.** 10% random-subset unlearning.

Method	UA	MIA	RA	TA	Avg. Gap↓	CKA↑
Retrain	23.80±0.00	40.24±0.00	99.98±0.00	76.15±0.00	0.00±0.00	1.000±0.000
FT	32.39±3.95	59.15±0.78	71.07±4.60	52.48±3.35	20.39±3.83	0.551±0.019
+NCAR	25.93±4.06	54.21±2.67	81.28±4.88	60.26±2.24	**12.64±3.17**	**0.636±0.051**
NegGrad+	27.21±3.34	60.66±0.79	92.07±1.40	65.94±0.26	7.17±0.78	0.706±0.010
+NCAR	24.70±0.81	53.46±0.50	94.52±1.22	67.71±0.59	**4.94±0.47**	**0.738±0.003**
SalUn	24.18±1.65	21.21±0.89	97.95±0.17	67.21±0.20	4.12±0.12	0.727±0.005
+NCAR	21.16±1.87	16.54±1.90	99.03±0.12	68.96±0.25	**3.59±0.51**	**0.758±0.000**
SCRUB	23.74±0.96	53.77±1.01	98.81±0.14	70.26±0.63	2.61±0.28	0.802±0.005
+NCAR	23.21±1.04	51.01±3.36	98.79±1.55	71.14±1.87	**2.40±1.03**	**0.816±0.035**
Bad Teacher	24.96±2.71	35.76±6.96	94.53±2.00	64.96±1.02	6.12±0.26	0.692±0.019
+NCAR	24.56±0.90	23.92±7.84	97.36±2.17	68.27±2.05	**3.79±1.22**	**0.749±0.019**
RL	37.96±3.56	34.14±5.60	82.94±11.13	59.81±5.76	15.85±6.81	0.607±0.071
+NCAR	29.96±0.67	29.73±7.99	89.49±8.82	64.67±4.10	**9.37±4.39**	**0.698±0.060**
UGradSL+	21.90±0.99	68.38±0.06	98.33±0.12	67.96±0.17	3.91±0.40	0.768±0.001
+NCAR	21.61±0.71	69.01±0.57	98.37±0.06	68.24±0.22	**3.90±0.32**	**0.769±0.001**

Table 8: **ViT-Small/8 on CIFAR-10: full-class unlearning.** Class 0 removed.

Method	UA	MIA	RA	TA	Avg. Gap↓	CKA↑
Retrain	100.00±0.00	30.53±0.00	99.93±0.00	85.75±0.00	0.00±0.00	1.000±0.000
FT	92.49±1.64	17.73±3.99	92.03±0.87	81.20±0.62	6.66±0.93	0.788±0.008
+NCAR	99.93±0.10	20.39±1.91	98.21±0.14	84.61±0.14	**0.98±0.10**	**0.900±0.002**
NegGrad+	100.00±0.00	99.12±0.30	93.73±0.28	80.85±0.11	3.70±0.07	0.775±0.010
+NCAR	99.94±0.10	71.19±24.83	95.34±1.22	81.70±0.81	**2.90±0.63**	**0.869±0.085**
SalUn	99.95±0.03	0.07±0.04	99.35±0.05	85.57±0.06	0.27±0.01	0.914±0.006
+NCAR	99.96±0.01	0.07±0.03	99.31±0.06	85.72±0.02	**0.23±0.02**	**0.920±0.003**
SCRUB	99.21±0.79	33.51±3.42	98.53±0.92	85.19±0.32	0.92±0.64	0.926±0.028
+NCAR	98.82±1.18	31.77±1.79	98.96±0.66	85.37±0.18	**0.84±0.67**	**0.939±0.019**
Bad Teacher	97.10±2.20	0.00±0.00	99.72±0.05	85.99±0.27	1.13±0.79	0.867±0.003
+NCAR	98.85±1.07	0.00±0.00	99.50±0.03	85.71±0.13	**0.56±0.35**	**0.896±0.003**
RL	99.72±0.30	0.01±0.01	98.09±0.45	84.04±0.34	1.28±0.31	0.874±0.005
+NCAR	99.92±0.12	0.01±0.01	98.23±0.27	84.41±0.27	**1.04±0.21**	**0.896±0.005**
UGradSL+	100.00±0.00	95.27±7.11	93.17±1.83	80.47±1.54	4.01±1.12	0.780±0.012
+NCAR	100.00±0.00	98.82±0.88	93.72±1.16	80.92±0.91	**3.68±0.69**	**0.805±0.013**

F Ablation Objective Definitions

All variants use NegGrad+ (Kurmanji et al., 2023). Let CE_R and CE_F denote the mean cross-entropy losses on the retained and forget minibatches used by the base update, respectively, and let $\alpha \in [0, 1]$. The base and regularized objectives are

$$\mathcal{L}_{\mathrm{NegGrad+}} = \alpha \mathrm{CE}_R - (1 - \alpha)\mathrm{CE}_F, \tag{47}$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{NegGrad+}} + \beta\Omega. \tag{48}$$

The variants change the retain-side structural regularizer Ω, not the forgetting objective. WC, ETF, and CA have the same meanings as in Section 4.4.

Table 9: **Extended Neural Collapse analysis.** NC1–NC4 measured at the Avg. Gap-selected checkpoints for CIFAR-10 10% random-subset unlearning with ResNet-18 and ViT-Small/8. Lower values indicate closer agreement with the corresponding NC property.

Architecture	Method	NC1↓	NC2↓	NC3↓	NC4↓
ResNet-18	FT	0.074±0.003	0.253±0.012	0.166±0.004	0.004±0.001
	+NCAR	**0.041±0.007**	**0.100±0.023**	**0.099±0.011**	**0.002±0.001**
	NegGrad+	0.082±0.014	0.259±0.007	0.173±0.002	0.005±0.002
	+NCAR	**0.051±0.010**	**0.145±0.032**	**0.123±0.020**	**0.002±0.002**
	SalUn	0.256±0.070	0.552±0.018	0.518±0.017	0.049±0.023
	+NCAR	**0.093±0.007**	**0.265±0.030**	**0.285±0.023**	**0.006±0.002**
	SCRUB	0.021±0.003	0.233±0.011	0.158±0.007	0.001±0.001
	+NCAR	**0.019±0.001**	**0.216±0.025**	**0.151±0.008**	**0.001±0.000**
	Bad Teacher	0.026±0.002	0.330±0.031	0.116±0.019	0.000±0.000
	+NCAR	**0.015±0.000**	**0.123±0.015**	**0.080±0.011**	**0.000±0.000**
	RL	0.058±0.004	0.438±0.108	0.175±0.018	0.004±0.003
	+NCAR	**0.031±0.001**	**0.172±0.033**	**0.108±0.018**	**0.001±0.000**
	UGradSL+	**0.040±0.003**	0.244±0.012	0.151±0.006	**0.000±0.000**
	+NCAR	0.041±0.003	**0.240±0.011**	**0.148±0.006**	0.001±0.000
ViT-Small/8	FT	0.309±0.006	0.587±0.017	0.430±0.014	0.025±0.002
	+NCAR	**0.155±0.003**	**0.190±0.005**	**0.171±0.004**	**0.007±0.001**
	NegGrad+	0.306±0.003	0.543±0.008	0.386±0.005	0.022±0.001
	+NCAR	**0.252±0.005**	**0.367±0.014**	**0.272±0.007**	**0.009±0.002**
	SalUn	0.314±0.011	0.535±0.020	0.284±0.033	0.040±0.003
	+NCAR	**0.287±0.014**	**0.499±0.014**	**0.271±0.014**	**0.031±0.002**
	SCRUB	0.176±0.006	0.386±0.012	0.287±0.008	0.011±0.002
	+NCAR	**0.173±0.006**	**0.334±0.010**	**0.285±0.008**	**0.008±0.002**
	Bad Teacher	0.215±0.001	0.268±0.016	0.166±0.009	0.012±0.002
	+NCAR	**0.210±0.002**	**0.253±0.014**	**0.164±0.008**	**0.010±0.001**
	RL	0.212±0.027	0.274±0.040	**0.111±0.005**	0.015±0.004
	+NCAR	**0.206±0.027**	**0.257±0.032**	0.111±0.008	**0.014±0.001**
	UGradSL+	0.264±0.004	0.220±0.017	0.086±0.001	0.012±0.000
	+NCAR	**0.241±0.001**	**0.192±0.011**	**0.084±0.002**	**0.009±0.000**

F.1 Common Translation and Scale Fit

For B_R with $m = |B_R|$, let n_c^B be the number of examples from class c and define $\mathcal{C}_B = \{c \in \mathcal{C}_R : n_c^B > 0\}$. Write

$$\bar{h} = \frac{1}{m} \sum_{i \in B_R} h_i, \qquad \pi_c = \frac{n_c^B}{m}. \tag{49}$$

Given class directions u_c, define

$$\bar{u} = \sum_{c \in \mathcal{C}_R} \pi_c u_c, \qquad d_c = u_c - \bar{u}. \tag{50}$$

All four structural configurations use the same denominator-floored fit as Full NCAR:

$$N = \sum_{i \in B_R} \langle h_i - \bar{h}, d_{y_i} \rangle, \qquad D_a = \sum_{i \in B_R} \|d_{y_i}\|_2^2, \tag{51}$$

$$s^\star = \left[\frac{N}{\max(D_a, \epsilon)}\right]_+, \qquad t_c = \bar{h} + s^\star d_c, \qquad \epsilon = 10^{-12}. \tag{52}$$

For Full NCAR, $-$ETF, and $-$CA, the common instance-level residual is

$$\Omega_{\text{inst}} = \frac{1}{m} \sum_{i \in B_R} \|h_i - t_{y_i}\|_2^2. \tag{53}$$

The $-$WC variant instead uses the class-mean residual defined below.

F.2 Full NCAR: WC, ETF, and CA

Full NCAR uses $u_c = (A_W)_c = Q_W v_c$, with Q_W defined by Equation 6, and

$$\Omega_{\text{Full}} = \Omega_{\text{inst}}. \tag{54}$$

The directions are a classifier-aligned simplex, and the instance residual promotes within-class concentration.

F.3 Removing Within-Class Concentration ($-$WC)

The $-$WC variant uses the same directions and targets as Full NCAR. For observed classes $c \in \mathcal{C}_B$, let $\mu_c^B = (n_c^B)^{-1} \sum_{i \in B_R : y_i = c} h_i$. Its regularizer is

$$\Omega_{-\text{WC}} = \frac{1}{m} \sum_{c \in \mathcal{C}_B} n_c^B \|\mu_c^B - t_c\|_2^2. \tag{55}$$

The decomposition

$$\Omega_{\text{Full}} = \frac{1}{m} \sum_{i \in B_R} \|h_i - \mu_{y_i}^B\|_2^2 + \Omega_{-\text{WC}} \tag{56}$$

shows that this removes the explicit within-class term while retaining classifier-aligned simplex fitting of class means.

F.4 Removing ETF Projection ($-$ETF)

The $-$ETF variant omits Procrustes projection onto the simplex family and directly uses the current centered, normalized classifier directions:

$$u_c = \widehat{w}_c = \frac{w_c - \bar{w}}{\max(\|w_c - \bar{w}\|_2, \epsilon)}, \qquad \Omega_{-\text{ETF}} = \Omega_{\text{inst}}. \tag{57}$$

This retains instance-level concentration and classifier conditioning without explicitly imposing simplex geometry. The directions change as the classifier is updated.

F.5 Removing Classifier Alignment ($-$CA)

The $-$CA variant retains the simplex but fits its orientation to the retained minibatch class means rather than the classifier. Let $\bar{v} = \sum_{c \in \mathcal{C}_R} \pi_c v_c$. Define

$$Q_H \in \arg \min_{Q \in \mathbb{R}^{d \times d},\, Q^\top Q = I_d} \sum_{c \in \mathcal{C}_B} \pi_c \|(\mu_c^B - \bar{h}) - Q(v_c - \bar{v})\|_2^2, \tag{58}$$

$$u_c = Q_H v_c, \qquad \Omega_{-\text{CA}} = \Omega_{\text{inst}}. \tag{59}$$

This preserves instance-level concentration and simplex geometry while removing explicit classifier conditioning.

F.6 Component Summary

The unregularized baseline has $\Omega = 0$. Each leave-one-out variant removes one specified construction component while retaining the other two. In a single-class minibatch, $N = D_a = 0$ and $s^\star = 0$: the instance residual is scatter around $\bar{h}$, whereas the $-$WC residual is zero.

G Computational Overhead

Let C_{base} denote the per-update cost of the base method, m the retained minibatch size, d the feature dimension, and C_R the number of remaining classes. With the reduced Procrustes formulation in Appendix C.1, classifier alignment costs $O(dC_R^2)$ under $C_R - 1 \leq d$. Translation and scale fitting and residual evaluation cost $O(md)$. When the required retained-data forward and backward passes are already part of the base update, the total per-update cost is

$$O\big(C_{\text{base}} + d(m + C_R^2)\big)\,. \tag{60}$$

Otherwise, additional retained-data network computation must also be included.

Table 10: **Computational overhead of NCAR.** CIFAR-10/ResNet-18, 10% random-subset unlearning. Complexity uses the reduced alignment described above.

Method	**Variant**	**Time complexity**	Time/step	Total time	Peak memory
SalUn	Base	$O(C_{\text{base}})$	130.1 ms	0.76 min	1.46 GB
	+NCAR	$O(C_{\text{base}} + d(m + C_R^2))$	189.9 ms	0.90 min	1.46 GB
NegGrad+	Base	$O(C_{\text{base}})$	210.5 ms	0.94 min	2.55 GB
	+NCAR	$O(C_{\text{base}} + d(m + C_R^2))$	254.7 ms	0.92 min	2.55 GB
UGradSL+	Base	$O(C_{\text{base}})$	257.2 ms	0.93 min	2.55 GB
	+NCAR	$O(C_{\text{base}} + d(m + C_R^2))$	324.2 ms	1.16 min	2.55 GB

Timing protocol. Table 10 reports single-run, seed-0 profiles over one epoch with batch size 256, using identical hardware, data pipelines, and precision. Time/step is the CUDA-synchronized online epoch time divided by the number of optimizer updates. Total time additionally includes setup, method preparation (including SalUn mask construction), checkpoint-selection evaluation, and checkpoint saving, but excludes hyperparameter search and downstream diagnostics. Peak memory is measured using `torch.cuda.max_memory_allocated`. Time/step more directly reflects the incremental online cost of NCAR, whereas total time also includes setup and checkpoint-selection evaluation.