SHARED AFFINE COMPENSATION FOR EXEMPLAR-FREE CLASS-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Exemplar-free class-incremental learning must preserve past knowledge while adapting representations to new classes. Class means and covariances offer a compact memory, but keeping them aligned with an evolving feature space is difficult without old images. Our key observation is that matching these low-order moments leaves more freedom than aligning individual features. This motivates a single affine map shared across classes, fitted on current-task feature pairs to update stored statistics in closed form. Our analysis characterizes affine moment matching and bounds a shared map's old-class errors in terms of feature drift and cross-class response differences. Building on this foundation, we propose Recoverable Feature Learning with Affine Compensation (ReFAC), which combines this memory update with flexible representation learning. ReFAC trains a decoder to recover previous features and their within-class covariances, preserving recoverable information while the representation adapts to new classes. Replacing the statistics updates of four representative learners with the shared affine map largely preserves or improves final accuracy while reducing update cost. Extensive experiments on standard benchmarks show that ReFAC consistently achieves competitive or superior performance compared with state-of-the-art methods.

1 INTRODUCTION

Class-incremental learning requires a model to acquire new classes while continuing to recognize all previously seen classes, without task identifiers at inference (Masana et al., 2023). In exemplar-free class-incremental learning (EFCIL), past training images cannot be retained, making it difficult to preserve old knowledge as the model learns from new data. This tension is especially pronounced in cold-start learning, where training begins from scratch with only a small set of classes (Magistri et al., 2024). The initial representation must continue to develop, yet each update changes the feature space in which earlier classes were learned.

Class means and covariances provide a compact memory of earlier classes, supporting classification without storing their images. Keeping this memory useful requires coordinating feature learning with statistics updates. Existing methods implement this coordination in different ways. EFC estimates affinity-weighted prototype shifts (Magistri et al., 2024), while DPCR combines task-wise and category-specific projections before reconstructing a classifier (He et al., 2025). AdaGauss learns adapters to update Gaussian class memories (Rypeść et al., 2024), and BiCyc couples forward and backward mappings through cycle consistency (Xu & Krawczyk, 2026). Across these designs, a shared requirement is to bring stored class statistics into alignment with the evolving representation.

We ask how much mapping complexity is needed to maintain this memory. A class's mean and covariance leave much of its feature distribution unspecified, so matching these summaries leaves freedom in how individual features are mapped. This motivates studying a common compensation rule across different learners. We fit a single affine map per task from the previous and current features of current-task images and apply it directly to stored statistics. Replacing the updates in EFC, AdaGauss, DPCR, and BiCyc, while retaining their feature-learning objectives and prediction rules, largely preserves or improves final accuracy across three datasets and two task counts (Table 1). This finding identifies a shared affine update as a practical basis for maintaining low-order class memory.

Our analysis separates class-wise moment expressivity from the transfer of a shared fitted map. For an individual class with a full-rank source covariance, an affine map can match any target mean

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.

and covariance, even when the underlying feature change is nonlinear. Once a map is given, both moments can be updated exactly from stored statistics. In the exemplar-free setting, the shared map is fitted on current-task data. Its old-class moment errors depend on feature drift, regularization, and cross-class differences in feature and update statistics, which we characterize through error bounds.

Feature learning and memory compensation operate at different levels: the former must preserve useful information in individual features while learning new classes, whereas the latter updates class-level moments. In our feature-learning comparison, directly regularizing feature changes toward a shared affine map reduces moment errors but also lowers new-class and overall accuracy (Appendix C). This motivates *Recoverable Feature Learning with Affine Compensation* (ReFAC). We augment projected distillation (Rypeść et al., 2024) with within-class covariance matching in a learned decoder's output space, preserving recoverable information while allowing the representation to adapt. After training, a shared affine map updates old-class statistics in closed form. The decoder is discarded, and inference uses the current feature extractor and a Mahalanobis classifier.

Our contributions are as follows:

- We show that task-shared affine compensation can replace the statistics updates of four EFCIL methods while largely preserving or improving final accuracy across three datasets and two task counts. On CIFAR-100 with ten tasks, it accelerates the post-training update stage of AdaGauss and BiCyc by approximately 29–$31\times$.
- We characterize class-wise affine moment expressivity and derive error bounds for a shared map fitted on current-task data, identifying the roles of feature drift and cross-class response differences in old-class moment recovery.
- We instantiate this update in ReFAC, combining projected distillation and within-class covariance matching with closed-form memory compensation. ReFAC improves final accuracy over AdaGauss and BiCyc in all eight settings on four datasets, and component ablations quantify the contributions of its training objectives.

2 Related Work

Exemplar-free class-incremental learning. Class-incremental learning requires recognizing all seen classes without task identifiers (Masana et al., 2023). Exemplar-based methods combine stored samples with rehearsal and distillation (Rebuffi et al., 2017; Hou et al., 2019; Douillard et al., 2020). Exemplar-free approaches retain past knowledge through model constraints, synthetic data, or class statistics. Generative replay and model inversion synthesize past inputs (Shin et al., 2017; Smith et al., 2021), while compact feature memories support prototype augmentation (Zhu et al., 2021a;b), representation expansion (Zhu et al., 2022), and feature translation (Petit et al., 2023). FeCAM uses class covariances for Mahalanobis classification (Goswami et al., 2023). Fixed representations also enable analytic classifier updates, as in ACIL and RanPAC (Zhuang et al., 2022; McDonnell et al., 2023). Freezing the extractor avoids drift but ties subsequent recognition to its initial features. Cold-start learning, with few initial classes and no pretraining, makes continued feature adaptation particularly important (Magistri et al., 2024).

Compensation for feature drift. Updating the feature extractor makes stored class representations outdated (Yu et al., 2020; Toldo & Ozay, 2022). SDC estimates prototype shifts from current-task feature displacements (Yu et al., 2020); ADC improves estimation through adversarially generated inputs (Goswami et al., 2024), and LDC learns a mapping between successive feature spaces (Gomez-Villa et al., 2024). EFC combines selective feature regularization with affinity-weighted prototype updates (Magistri et al., 2024). AdaGauss extends compensation to class covariances by mapping samples from stored Gaussians and re-estimating their moments (Rypeść et al., 2024). BiCyc jointly learns forward and backward mappings with cycle consistency to improve cross-task alignment (Xu & Krawczyk, 2026). DPCR combines a closed-form task-wise linear projection with category-specific subspace projection, calibrating class means and uncentered second-order statistics for ridge-regression classifier reconstruction (He et al., 2025).

Feature regularization and representation geometry. Parameter regularization protects weights important to previous tasks (Kirkpatrick et al., 2017; Zenke et al., 2017; Aljundi et al., 2018),

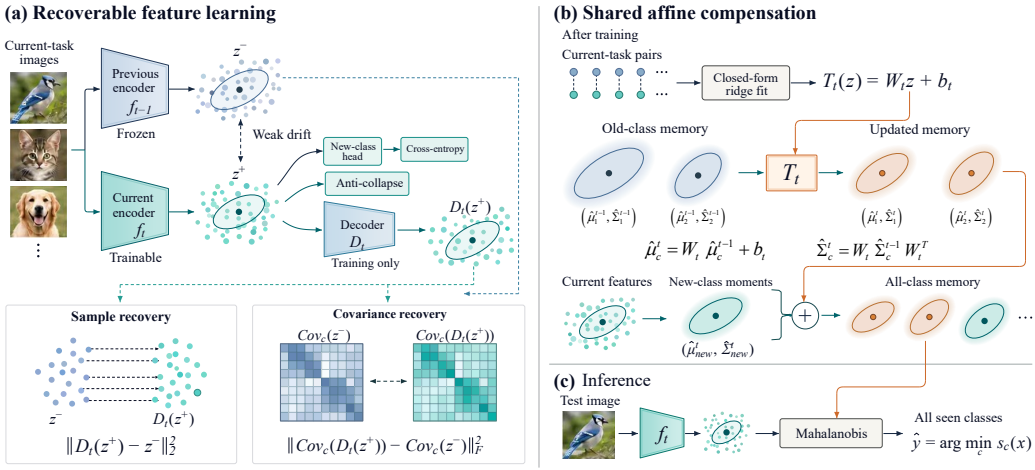

Figure 1: Overview of ReFAC. (a) A training-only decoder recovers previous features and within-class covariances while the encoder learns new classes. (b) A shared affine map fitted on current-task feature pairs updates old-class means and covariances in closed form. (c) The current encoder and all-class memory support Mahalanobis classification.

while knowledge distillation transfers previous model outputs (Hinton et al., 2015; Li & Hoiem, 2016). Projected Functional Regularization and CaSSLe use learnable mappings to recover earlier representations while allowing new features to develop (Gomez-Villa et al., 2022; Fini et al., 2022). AdaGauss combines projected distillation with an anti-collapse objective (Rypeść et al., 2024). Beyond pointwise feature matching, second-order statistics also characterize representation similarity (Kornblith et al., 2019) and support distribution alignment. Deep CORAL aligns feature covariances across domains (Sun & Saenko, 2016), while Barlow Twins and VICReg use second-order objectives to reduce redundancy and avoid representation collapse (Zbontar et al., 2021; Bardes et al., 2022).

3 Method

We study affine compensation as a common update for low-order class memory. We first characterize its class-wise moment expressivity, then fit a task-shared map and analyze the factors affecting its transfer to old classes. Finally, we instantiate this update in ReFAC with projected distillation and within-class covariance matching (Figure 1).

3.1 Learning Setup and Class Statistics

We consider exemplar-free class-incremental learning over datasets $\mathcal{D}_1, \ldots, \mathcal{D}_T$ with disjoint class sets $\mathcal{C}_1, \ldots, \mathcal{C}_T$. Task t uses only $\mathcal{D}_t$, and testing covers all seen classes $\mathcal{C}_{1:t} = \bigcup_{s=1}^{t} \mathcal{C}_s$ without a task identifier. Let $f_t : \mathcal{X} \to \mathbb{R}^d$ be the feature extractor after task t.

With few classes in the initial task, continued representation learning is essential (Magistri et al., 2024). A common memory consists of class means and covariances, denoted by $\mathcal{M}_{t-1} = \{(\widehat{\mu}_c^{t-1}, \widehat{\Sigma}_c^{t-1}) : c \in \mathcal{C}_{1:t-1}\}$. As f_t evolves, these stored statistics must be updated to the current feature space.

Existing methods implement this update differently: AdaGauss and BiCyc learn adapters for Gaussian class memories (Rypeść et al., 2024; Xu & Krawczyk, 2026); DPCR calibrates class moments before reconstructing a classifier (He et al., 2025); EFC updates class means through affinity-weighted feature displacements (Magistri et al., 2024; 2025). These different mechanisms serve the same role: updating low-order class memory as the representation evolves.

3.2 Affine Expressivity for Class Statistics

The compensation target is the class mean and covariance in the updated representation. Many feature distributions share these moments, so matching them leaves freedom in how individual features are mapped. The following result characterizes when an affine map can achieve this target.

Theorem 1 (Affine moment matching)**.** *For a class c, let P_c denote the distribution of inputs from that class, and define $z^- = f_{t-1}(x)$ and $z^+ = f_t(x)$ for $x \sim P_c$. Assume finite second moments, with means μ_c^-, μ_c^+ and covariances Σ_c^-, Σ_c^+. There exists an affine map $T_c(z) = A_c z + b_c$ satisfying*

$$\mathbb{E}[T_c(z^-)] = \mu_c^+, \qquad \mathrm{Cov}(T_c(z^-)) = \Sigma_c^+ \tag{1}$$

if and only if $\mathrm{rank}(\Sigma_c^+) \le \mathrm{rank}(\Sigma_c^-)$. *In particular, if $\Sigma_c^- \succ 0$, any target mean and positive semidefinite target covariance can be matched by*

$$A_c = (\Sigma_c^+)^{1/2}(\Sigma_c^-)^{-1/2}, \qquad b_c = \mu_c^+ - A_c\mu_c^-. \tag{2}$$

Proof sketch. The transformed mean and covariance are $A_c\mu_c^- + b_c$ and $A_c\Sigma_c^- A_c^\top$. The covariance rank cannot exceed $\mathrm{rank}(\Sigma_c^-)$, establishing necessity. For sufficiency, map source directions with positive eigenvalues onto the target eigenspace and rescale them; the rank condition ensures enough directions exist. The offset matches the mean. Equation 2 gives the positive definite case; Appendix A gives the full construction. □

Implication for low-order memory. Affine maps can match each class's mean and covariance under the rank condition for any distribution with finite second moments, even when the underlying feature changes are nonlinear. For any given affine map, these summaries can be updated exactly in closed form using only the stored mean and covariance. General nonlinear maps can require additional distributional information to compute the same summaries. Together, moment expressivity and closed-form updates from class memory motivate affine compensation.

3.3 Task-Shared Affine Compensation

The theorem constructs a separate map from each class's target moments. Since old-class target moments are unavailable under the exemplar-free constraint, we estimate a shared map $T_t(z) = W_t z + b_t$ from current-task feature pairs.

After training, freeze both extractors and collect $X = [f_{t-1}(x_i)]_{i=1}^n$ and $Y = [f_t(x_i)]_{i=1}^n$ in $\mathbb{R}^{d\times n}$ for all n inputs in $\mathcal{D}_t$. Let $\bar{x}, \bar{y}$ be their column means and $\widetilde{X} = X - \bar{x}\mathbf{1}^\top$, $\widetilde{Y} = Y - \bar{y}\mathbf{1}^\top$. We solve

$$\min_{W,b} \ \|WX + b\mathbf{1}^\top - Y\|_F^2 + \rho\|W - I\|_F^2, \tag{3}$$

which has the closed-form solution

$$W_t = (\widetilde{Y}\widetilde{X}^\top + \rho I)(\widetilde{X}\widetilde{X}^\top + \rho I)^{-1}, \qquad b_t = \bar{y} - W_t\bar{x}. \tag{4}$$

We fix $\rho = 10^{-3}$. All old classes share the resulting map:

$$\widehat{\mu}_c^t = W_t\widehat{\mu}_c^{t-1} + b_t, \qquad \widehat{\Sigma}_c^t = W_t\widehat{\Sigma}_c^{t-1}W_t^\top, \quad c \in \mathcal{C}_{1:t-1}. \tag{5}$$

Moment accuracy of the fitted map. We relate the fitted map's moment error to feature drift and cross-class differences in the feature-update response. Let P be the uniform distribution over the n fitting samples, $z = f_{t-1}(x)$, $u = f_t(x) - z$, and $\lambda = \rho/n$. For $\varepsilon > 0$ with $\mathbb{E}_P\|u\|_2^2 \le \varepsilon^2$, ridge regression matches the current-task empirical mean exactly and satisfies

$$\left\|\mathrm{Cov}_P(z+u) - W_t\,\mathrm{Cov}_P(z)W_t^\top\right\|_F \le \varepsilon^2 + \varepsilon\sqrt{\lambda}. \tag{6}$$

For an old class c, let $E_{\mu,c}$ and $E_{\Sigma,c}$ be the errors introduced by transporting its exact preceding mean and covariance. Write $k_t = \|W_t - I\|_{\mathrm{op}}$, and let δ_c measure the joint mean and covariance differences of $(z, u/\varepsilon)$ between class c and P, as defined in Appendix B. This discrepancy depends on both the class distributions and the learned feature update. Then

$$E_{\mu,c} \le (\varepsilon + k_t)\delta_c, \qquad E_{\Sigma,c} \le \varepsilon^2 + \varepsilon\sqrt{\lambda} + (2\varepsilon + \varepsilon^2 + 2k_t + k_t^2)\delta_c. \tag{7}$$

The ridge solution also satisfies $k_t \leq \varepsilon/(2\sqrt{\lambda})$. Thus, for fixed $\lambda > 0$ and uniformly bounded δ_c, both old-class moment errors vanish as $\varepsilon \to 0$. The bounds describe how feature drift and cross-class response differences jointly affect moment recovery. The direct feature penalty in Section 3.4 penalizes the current-task drift energy. Theorem 2 in Appendix B gives the complete proof and the propagation of existing memory errors. Appendix B.1 measures the cross-class discrepancy and actual old-class transport errors on CIFAR-100 with ten tasks.

3.4 Feature Learning and Inference

With shared affine compensation established, we turn to feature learning. A natural choice is to encourage each current feature to follow an affine transformation of its previous value. On CIFAR-100 with ten tasks, this strategy yields smaller compensation errors, yet its best tested final accuracy is 49.44%, compared with ReFAC's 53.31% (Appendix C). The distinction is between maintaining discriminative features and transporting their class summaries: accurate moment recovery alone is insufficient for recognition. Inspired by projected distillation in AdaGauss (Rypeść et al., 2024) and second-order representation similarity (Kornblith et al., 2019), we use a learnable decoder $D_t : \mathbb{R}^d \to \mathbb{R}^d$ to recover previous features from current representations while matching their within-class covariances. Implemented as a two-layer MLP with GELU, D_t is jointly trained with f_t, while f_{t-1} remains frozen:

$$\mathcal{L}_{\mathrm{proj}} = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \|D_t(f_t(x)) - f_{t-1}(x)\|_2^2 + \frac{1}{|\mathcal{C}_{\mathcal{B}}|} \sum_{c\in\mathcal{C}_{\mathcal{B}}} \|\mathrm{Cov}_c(D_t \circ f_t) - \mathrm{Cov}_c(f_{t-1})\|_F^2 . \quad (8)$$

Here, $\mathcal{B}$ is a minibatch, $\mathcal{C}_{\mathcal{B}}$ contains its classes with at least two samples, and Cov_c is their sample covariance under the specified mapping.

We retain AdaGauss's anti-collapse term $\mathcal{L}_{\mathrm{ac}} = -\frac{1}{d}\sum_{j=1}^{d} \min((L_{\mathcal{B}})_{jj}, \beta)$ to maintain feature variation (Rypeść et al., 2024), where $L_{\mathcal{B}}$ is the Cholesky factor of the minibatch covariance with a small diagonal offset and β caps each contribution. A weak direct penalty $\mathcal{L}_{\mathrm{fd}} = \frac{1}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}} \|f_t(x) - f_{t-1}(x)\|_2^2$ penalizes the drift energy in the moment-error bounds of Section 3.3. A temporary classifier g_t learns current classes through cross-entropy $\mathcal{L}_{\mathrm{ce}}$, giving

$$\mathcal{L} = \mathcal{L}_{\mathrm{ce}} + \lambda_{\mathrm{proj}}\mathcal{L}_{\mathrm{proj}} + \lambda_{\mathrm{fd}}\mathcal{L}_{\mathrm{fd}} + \alpha\mathcal{L}_{\mathrm{ac}}. \quad (9)$$

After training, we apply Equations 4–5 to update old-class memory, compute new-class means and covariances from current features, and discard D_t and g_t. Prediction uses the Mahalanobis distance (Mahalanobis, 1936). For $z = f_t(x)$ and numerically regularized covariance $\overline{\Sigma}_c^t$,

$$s_c(x) = (z - \widehat{\mu}_c^t)^\top (\overline{\Sigma}_c^t)^{-1}(z - \widehat{\mu}_c^t), \qquad \widehat{y} = \arg\min_{c\in\mathcal{C}_{1:t}} s_c(x). \quad (10)$$

Algorithm 1 in Appendix F summarizes the complete procedure.

4 Experiments

We first evaluate shared affine compensation by replacing the statistics updates in existing EFCIL methods, measuring classification accuracy, recovery of old-class statistics, and computational cost. We then evaluate ReFAC as a complete learner built around this update and use ablations to examine its feature-learning objectives.

4.1 Experimental Setup

Datasets and evaluation. We follow the cold-start benchmark protocol of AdaGauss (Rypeść et al., 2024) on CIFAR-100 (Krizhevsky, 2009), TinyImageNet (Le & Yang, 2015), and ImageNet-100 (Deng et al., 2009). CIFAR-100 contains 100 classes with 32×32 images; TinyImageNet contains 200 classes with 64×64 images; ImageNet-100 contains 100 classes from ImageNet. We also evaluate fine-grained recognition on CUB-200 (Wah et al., 2011), which contains 200 bird species. Each dataset is divided into $T = 10$ or $T = 20$ tasks with equal numbers of classes, including the initial task. Training uses the current task's images, and prediction is over all classes seen so far

without a task identifier. Let a_t denote the corresponding test accuracy after task t. We report final accuracy $A_{\text{last}} = a_T$ and average incremental accuracy $A_{\text{inc}} = T^{-1} \sum_{t=1}^{T} a_t$ in percent, as mean $\pm$ standard deviation over five runs.

Compared methods. We compare ReFAC with regularization-based methods EWC (Kirkpatrick et al., 2017) and LwF (Li & Hoiem, 2016); prototype and covariance-based methods PASS (Zhu et al., 2021b), FeTrIL (Petit et al., 2023), and FeCAM (Goswami et al., 2023); and drift-compensation methods SDC (Yu et al., 2020), EFC (Magistri et al., 2024), ADC (Goswami et al., 2024), LDC (Gomez-Villa et al., 2024), AdaGauss (Rypeść et al., 2024), DPCR (He et al., 2025), and BiCyc (Xu & Krawczyk, 2026). To examine affine compensation across different statistics updates, we additionally replace the update rules of EFC, AdaGauss, DPCR, and BiCyc while retaining each method's feature-learning objectives and prediction rules.

Implementation details. All experiments use a ResNet-18 backbone (He et al., 2016), trained from scratch on CIFAR-100, TinyImageNet, and ImageNet-100 and initialized from ImageNet-pretrained weights on CUB-200. For ReFAC, a linear layer reduces the 512-dimensional backbone output to 64 dimensions, and the decoder is a two-layer MLP with GELU and a hidden dimension of 2048 (32×64). On the three cold-start benchmarks, we train for 200 epochs per task with a batch size of 256, using SGD with momentum 0.9 and weight decay 5×10^{-4}. The initial learning rate is 0.1 for the first task and 0.01 for each subsequent task; it is multiplied by 0.1 at the end of epochs 60, 120, and 180 within each task. The decoder uses the same optimizer settings as the backbone. We fix $\lambda_{\text{proj}} = 2.5$, $\lambda_{\text{fd}} = 0.25$, and $\alpha = 1$. After each task, we fit the shared affine map with $\rho = 10^{-3}$ and update class statistics in closed form. All training is performed on an NVIDIA GeForce RTX 3090 Ti GPU.

4.2 Analysis of Affine Compensation

Evidence across methods. Table 1 evaluates affine replacement in AdaGauss, BiCyc, DPCR, and EFC across three datasets and two task counts. Affine compensation matches or improves AdaGauss's final accuracy in all six settings; changes for BiCyc and DPCR range from -0.59 to $+0.70$ percentage points. The EFC variant updates both means and covariances in place of its original mean update and improves accuracy in all six settings. Its updated memory also participates in subsequent feature learning. Across these different uses of class statistics, a shared affine map provides an effective common update.

Table 1: Final accuracy (%) with original statistics updates or shared affine compensation. Bold marks the better result within each method and setting, including ties.

		CIFAR-100		TinyImageNet		ImageNet-100	
Base method	Update	$T = 10$	$T = 20$	$T = 10$	$T = 20$	$T = 10$	$T = 20$
AdaGauss	Original	50.01	39.90	36.24	31.26	**51.64**	42.40
	Affine	**50.13**	**42.09**	**37.07**	**32.63**	**51.64**	**44.20**
BiCyc	Original	50.31	43.12	**37.17**	31.39	**52.34**	**44.02**
	Affine	**50.51**	**43.48**	36.94	**31.73**	52.04	43.88
DPCR	Original	49.83	39.43	**36.30**	26.41	50.46	38.46
	Affine	**50.30**	**40.13**	35.71	**26.86**	**50.56**	**38.71**
EFC	Original	45.07	33.41	34.53	27.89	49.74	38.52
	Affine	**48.22**	**38.40**	**38.97**	**32.40**	**52.50**	**42.76**

We examine moment recovery and runtime on CIFAR-100 with $T = 10$. For each base method, we compare three memory updates: *No update*, which retains each class's statistics from its arrival; *Original*, the base method's compensation; and *Affine*, the shared update in Equation 5. Each variant maintains its own recursive memory. AdaGauss, BiCyc, and DPCR share a feature-model trajectory across the three variants. EFC trains each variant independently because its memory participates in subsequent feature learning. Appendix D gives the diagnostic and timing protocols.

4.2.1 Recovery of Old-Class Statistics

Shared affine compensation reduces both mean and covariance errors relative to leaving old-class memories unchanged throughout the sequence (Figure 2). After each task, we compare stored moments with reference moments recomputed from old training images under the current feature extractor, using those images only for offline evaluation. We measure errors before classifier-specific processing. For each class, the mean error is divided by its reference within-class spread, $\sqrt{\operatorname{tr} \Sigma_c^{\star,t}}$, and the covariance error by $\|\Sigma_c^{\star,t}\|_F$; both are then averaged equally across old classes.

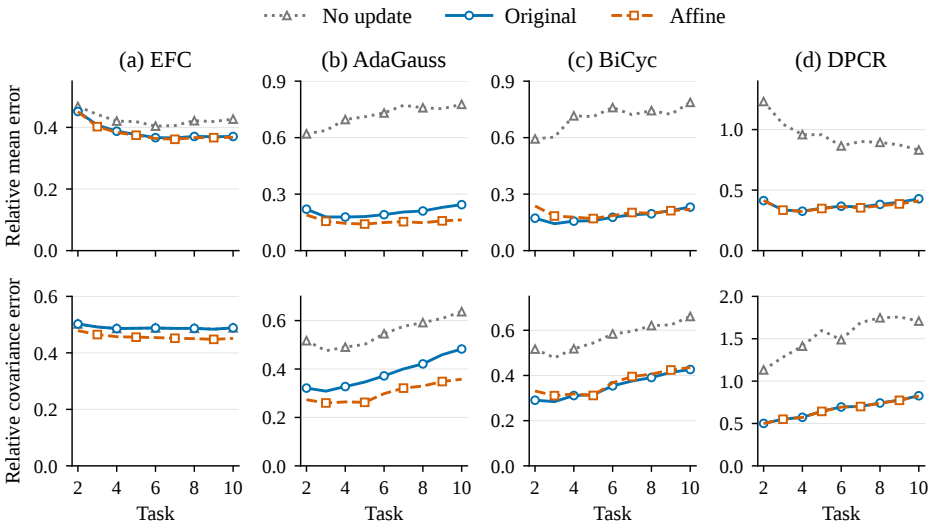

Figure 2: Relative mean (top) and covariance (bottom) errors on CIFAR-100 ($T = 10$), averaged over old classes before classifier-specific processing. Affine compensation improves over unchanged memories. Updates are recursive; No update freezes statistics at arrival. EFC uses independent training trajectories; other methods share one per column. Vertical scales vary by method and start at zero.

The affine variants of AdaGauss and EFC have lower mean and covariance errors than Original at every incremental stage. For AdaGauss, the final mean error decreases from 0.244 to 0.163, and the covariance error from 0.483 to 0.358. DPCR's affine covariance error is slightly lower at every stage, while its mean error is lower after task 2. BiCyc shows mixed differences: at the final task, affine compensation lowers the mean error from 0.231 to 0.219 and increases the covariance error from 0.426 to 0.435.

4.2.2 Computational Cost

Affine compensation provides its largest time savings when it replaces a trained adapter (Table 2). Timings on an RTX 3090 Ti cover feature extraction, statistics updates, and classifier preparation for AdaGauss, BiCyc, and DPCR. For EFC, they cover feature-pair extraction, fitting, and application. Speedup is Original/Affine.

Table 2: Cumulative update time (s) on CIFAR-100 ($T = 10$).

Method	Original	Affine	Speedup
EFC	8.990	8.302	1.08×
AdaGauss	695.824	22.470	30.97×
BiCyc	623.565	21.554	28.93×
DPCR	39.779	22.519	1.77×

Across the ten-task sequence, AdaGauss's update time decreases from 695.82 to 22.47 seconds, a $30.97\times$ speedup; BiCyc decreases from 623.57 to 21.55 seconds, a $28.93\times$ speedup. Their original adapters repeatedly extract feature pairs during optimization, whereas affine compensation collects one set of pairs and solves a regularized least-squares problem. DPCR already uses a linear projection, and replacing its update yields a smaller $1.77\times$ speedup. EFC's compensation takes 8.99 seconds

Table 3: Final (A_{last}) and average incremental (A_{inc}) accuracy on CIFAR-100 and TinyImageNet (%, mean ± std over five runs). Higher is better; column bests are bold.

	CIFAR-100				TinyImageNet			
	$T = 10$		$T = 20$		$T = 10$		$T = 20$	
Method	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
EWC	30.90±1.8	50.40±1.6	17.00±1.7	34.20±2.0	18.50±1.9	34.30±2.2	11.30±1.8	26.80±2.4
LwF	31.90±1.2	51.80±1.4	17.60±1.1	39.20±1.8	27.10±1.4	39.60±1.9	15.20±1.5	31.50±2.0
SDC	40.60±1.0	56.20±1.2	32.30±0.9	46.60±1.5	29.50±1.0	43.80±1.6	26.30±1.3	40.60±1.6
PASS	30.80±1.1	48.30±1.0	17.60±0.9	31.10±1.2	24.50±0.7	39.50±1.1	18.50±1.3	30.40±1.8
FeTrIL	34.90±0.6	51.20±1.0	23.30±1.3	37.90±1.3	31.00±0.8	45.30±1.7	25.90±1.1	39.90±1.3
FeCAM	32.40±0.6	48.70±0.8	21.10±0.9	34.50±1.2	30.90±1.0	44.90±1.3	24.90±0.9	37.90±1.5
EFC	45.07±0.9	59.43±1.1	33.41±1.0	48.57±1.4	34.53±1.0	47.42±1.4	27.89±1.1	41.56±1.5
ADC	46.50±1.3	61.40±1.5	35.10±1.3	51.70±1.7	32.30±1.4	43.00±1.8	18.10±1.7	36.00±2.0
LDC	45.40±1.5	59.50±1.8	35.50±1.8	51.90±2.2	34.20±1.2	46.80±1.7	24.90±2.1	38.20±2.6
AdaGauss	50.01±1.1	62.59±0.9	39.90±1.2	54.11±1.0	36.24±1.0	49.11±1.2	31.26±1.1	43.98±1.0
DPCR	49.83±0.8	62.72±1.0	39.43±1.1	54.72±1.0	36.30±1.6	48.47±1.0	26.41±0.8	39.62±0.7
BiCyc	50.31±1.0	63.54±1.2	43.12±1.0	56.74±1.2	37.17±0.9	49.01±1.3	31.39±1.0	43.83±1.2
ReFAC (Ours)	**53.31**±0.9	**65.04**±0.8	**44.50**±0.9	**58.59**±0.8	**39.17**±0.7	**49.95**±1.0	**32.56**±0.9	**44.03**±0.8

with its original rule and 8.30 seconds with the affine rule, a $1.08\times$ speedup. The largest reductions therefore come from eliminating iterative adapter training.

For our method, ReFAC retains AdaGauss's class-memory footprint (16.25 KiB per class in FP32 at $d = 64$) while using 98.4% fewer mapping coefficients for the statistics update; Appendix E details the parameter and storage costs.

4.3 Evaluation of ReFAC

Recognition performance. ReFAC attains the highest mean final and average incremental accuracies among the methods listed in Tables 3 and 4, across all eight settings. Relative to AdaGauss, final accuracy increases across all four datasets and both task counts, with gains of 4.60 and 2.72 percentage points on CIFAR-100 and ImageNet-100 at $T = 20$. Compared with BiCyc, the final-accuracy gains at $T = 20$ are 1.38, 1.17, 1.10, and 1.40 points on CIFAR-100, TinyImageNet, ImageNet-100, and CUB-200, respectively. At $T = 10$ on ImageNet-100, ReFAC reaches 53.46%, exceeding BiCyc's 52.34% by 1.12 points. These results show that shared affine compensation supports an effective complete learner across both cold-start and pretrained settings.

Direct affine replacement also produces competitive learners. Comparing final accuracies with Table 1, ReFAC exceeds all four affine variants on CIFAR-100 and ImageNet-100 at both task counts. On TinyImageNet, ReFAC and the strongest affine variants are close: at $T = 10$, ReFAC reaches 39.17% and EFC with affine compensation reaches 38.97%; at $T = 20$, ReFAC reaches 32.56% and AdaGauss with affine compensation reaches 32.63%.

Ablation study. Table 5 examines feature learning and memory maintenance within ReFAC on CIFAR-100 with $T = 10$ and $T = 20$. Keeping old-class statistics fixed reduces final accuracy by 9.71 and 16.35 percentage points, respectively, showing the importance of updating memory as the representation evolves. Within the recovery objective, removing covariance matching while retaining sample correspondence lowers final accuracy by 0.95 and 0.89 points. This isolates the benefit of the added second-order term. Removing the entire $\mathcal{L}_{\text{proj}}$ reduces final accuracy by 5.88 and 9.56 points, while removing the anti-collapse term $\mathcal{L}_{\text{ac}}$ costs 2.41 and 3.48 points. The direct feature penalty improves final accuracy by 0.84 and 0.58 points, while average incremental accuracy is nearly unchanged at $T = 10$ (65.04% versus 65.05%).

Hyperparameter sensitivity. Figure 3 examines the weights of the recovery objective and direct feature penalty on CIFAR-100 with $T = 10$ and $T = 20$. For λ_{proj}, final accuracy is highest among the tested values in the range 1.25–5 and decreases as the weight increases further. At $\lambda_{\text{proj}} = 2.5$, final accuracy reaches 53.31% and 44.50%, within 0.14 and 0.02 points of the best result in each

Table 4: Accuracy on ImageNet-100 and CUB-200, using the metrics and conventions of Table 3. CUB-200 uses ImageNet pretraining; dashes denote unavailable results.

	ImageNet-100				CUB-200			
	$T = 10$		$T = 20$		$T = 10$		$T = 20$	
Method	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}	A_{last}	A_{inc}
EWC	25.10±2.7	40.60±3.1	13.70±2.0	29.20±2.4	15.80±0.8	32.60±0.6	12.30±0.7	27.20±0.7
LwF	33.40±2.1	51.50±1.7	18.60±1.5	41.30±1.8	30.40±1.0	46.10±0.9	19.40±1.5	34.70±1.7
SDC	35.40±1.8	50.10±1.5	19.40±1.1	36.50±1.5	50.30±1.2	60.50±1.1	27.90±1.3	40.10±1.5
PASS	26.40±1.2	45.70±0.4	14.40±1.1	31.70±0.5	27.00±0.8	42.30±1.0	18.10±1.1	36.90±1.2
FeTrIL	36.20±1.1	52.60±0.7	26.60±1.4	42.40±2.0	36.90±0.8	48.20±0.7	34.60±0.9	45.30±1.0
FeCAM	38.70±0.9	54.80±0.6	29.00±1.2	44.60±1.9	40.20±0.9	54.90±0.9	36.20±1.0	48.90±1.2
EFC	49.74±1.0	60.94±1.1	38.52±1.3	52.91±1.4	–	–	–	–
ADC	38.30±1.1	55.50±1.4	25.10±1.2	43.40±1.6	49.50±1.0	58.80±1.2	35.40±1.3	48.30±1.5
LDC	50.49±1.1	65.40±0.7	28.50±1.6	46.50±2.5	47.50±0.8	55.70±1.2	27.20±1.2	39.80±2.0
AdaGauss	51.64±1.3	65.78±1.3	42.40±1.5	57.99±1.7	53.11±0.9	63.28±1.2	43.91±1.2	56.22±1.1
DPCR	50.46±0.9	64.37±1.0	38.46±1.5	55.18±0.8	–	–	–	–
BiCyc	52.34±1.0	65.69±1.3	44.02±1.3	57.95±1.6	52.37±0.8	62.85±0.9	43.60±1.3	55.22±1.1
ReFAC (Ours)	**53.46**±1.1	**66.13**±1.1	**45.12**±1.2	**58.44**±1.4	**53.37**±0.7	**64.65**±1.0	**45.00**±1.0	**57.22**±0.9

sequence. Increasing the weight to 20 reduces these accuracies to 45.05% and 39.24%, suggesting that an overly strong recovery constraint limits adaptation to new classes. Final accuracy varies less with λ_{fd} over the tested range from 0 to 1. Table 5 shows that the selected direct feature penalty improves final accuracy by 0.84 and 0.58 points for the two task counts. We select $\lambda_{\text{proj}} = 2.5$ and $\lambda_{\text{fd}} = 0.25$, combining a recovery weight near the peak in both sequences with a weak direct feature penalty.

Variant	$T = 10$	$T = 20$
Full method	**53.31**	**44.50**
w/o affine compensation	43.60	28.15
w/o $\mathcal{L}_{\text{proj}}$	47.43	34.94
w/o $\mathcal{L}_{\text{fd}}$	52.47	43.92
w/o $\mathcal{L}_{\text{ac}}$	50.90	41.02
w/o covariance term	52.36	43.61

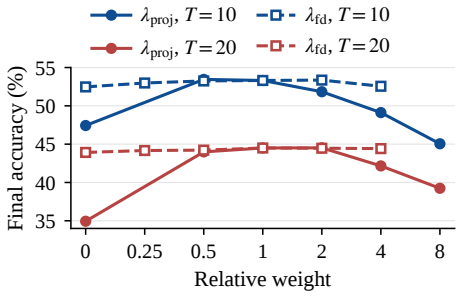

Table 5 and Figure 3: CIFAR-100 ablations (left; final accuracy, %; best in bold) and hyperparameter sensitivity (right). Relative weights divide λ_{proj} by 2.5 and λ_{fd} by 0.25.

5 Conclusion

Shared affine compensation offers a common update for low-order class memory in exemplar-free class-incremental learning. Our analysis connects affine moment expressivity with the transfer errors of a map fitted on current-task data. Across four learners, this update largely preserves or improves accuracy while reducing compensation cost. ReFAC combines it with recoverable feature learning and within-class covariance matching, achieving strong results across four datasets. These findings support learning adaptable representations while maintaining their class summaries through a shared closed-form update.

Limitations. Shared compensation depends on current-task drift being informative about old classes, so class-dependent changes can limit moment recovery. Recursive updates can propagate existing memory errors, and full covariances require storage quadratic in feature dimension. Class-dependent compensation and structured covariance memories are promising extensions.

AI USE STATEMENT

We did not use generative AI tools for any of the required disclosure tasks (e.g., generating synthetic datasets, formulating mathematical claims or proofs, proposing core hypotheses, designing the research methodology, or interpreting results). These were performed entirely by the authors.

We used generative AI tools (Codex Cli) solely for recommended tasks: polishing the writing for improved readability and grammar, and assisting with code implementation and debugging. All AI-generated text and code was reviewed, verified, and edited by the authors. We take full responsibility for the final content of this work.

REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision*, 2018. doi: 10.1007/978-3-030-01219-9_9. URL https://arxiv.org/abs/1711.09601.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2020. doi: 10.1007/978-3-030-58565-5_6. URL https://arxiv.org/abs/2004.13513.

Enrico Fini, Victor G. Turrisi da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Fini_Self-Supervised_Models_Are_Continual_Learners_CVPR_2022_paper.html.

Alex Gomez-Villa, Bartłomiej Twardowski, Lu Yu, Andrew D. Bagdanov, and Joost van de Weijer. Continually learning self-supervised representations with projected functional regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 3867–3877, 2022. URL https://openaccess.thecvf.com/content/CVPR2022W/CLVision/html/Gomez-Villa_Continually_Learning_Self-Supervised_Representations_With_Projected_Functional_Regularization_CVPRW_2022_paper.html.

Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D. Bagdanov, Bartłomiej Twardowski, and Joost van de Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *Proceedings of the European Conference on Computer Vision*, 2024. doi: 10.1007/978-3-031-72667-5_27.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost Van De Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36:6582–6595, 2023.

Dipam Goswami, Albin Soutif-Cormerais, Yuyang Liu, Sandesh Kamath, Bartłomiej Twardowski, and Joost van de Weijer. Resurrecting old classes with new data for exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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.

Run He, Di Fang, Yicheng Xu, Yawen Cui, Ming Li, Cen Chen, Ziqian Zeng, and Huiping Zhuang. Semantic shift estimation via dual-projection and classifier reconstruction for exemplar-free class-incremental learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22392–22406. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/he25d.html.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Hou_Learning_a_Unified_Classifier_Incrementally_via_Rebalancing_CVPR_2019_paper.html.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017. doi: 10.1073/pnas.1611835114. URL https://arxiv.org/abs/1612.00796.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 09–15 Jun 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.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Cs231n course report, Stanford University, 2015.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *Proceedings of the European Conference on Computer Vision*, 2016.

Simone Magistri, Tomaso Trinci, Albin Soutif, Joost van de Weijer, and Andrew D. Bagdanov. Elastic feature consolidation for cold start exemplar-free incremental learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7D9X2cFnt1.

Simone Magistri, Tomaso Trinci, Albin Soutif-Cormerais, Joost van de Weijer, and Andrew D. Bagdanov. Efc++: Elastic feature consolidation with prototype re-balancing for cold start exemplar-free incremental learning, 2025. URL https://arxiv.org/abs/2503.10439.

P. C. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2023. doi: 10.1109/TPAMI.2022.3213473. URL https://arxiv.org/abs/2010.15277.

Mark D. McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. RanPAC: Random projections and pre-trained models for continual learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 12022–12053, 2023. doi: 10.52202/075280-0526. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/2793dc35e14003dd367684d93d236847-Abstract-Conference.html.

Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. FeTrIL: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023. doi: 10.1109/WACV56688.2023.00390.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Rebuffi_iCaRL_Incremental_Classifier_CVPR_2017_paper.html.

Grzegorz Rypeść, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. Task-recency bias strikes back: Adapting covariances in exemplar-free class incremental learning. *Advances in Neural Information Processing Systems*, 37:63268–63289, 2024.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/0efbe98067c6c73dba1250d2beaa81f9-Paper.pdf.

James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9374–9384, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Smith_Always_Be_Dreaming_A_New_Approach_for_Data-Free_Class-Incremental_Learning_ICCV_2021_paper.html.

Baochen Sun and Kate Saenko. Deep CORAL: Correlation alignment for deep domain adaptation. In *Computer Vision – ECCV 2016 Workshops*, pp. 443–450, 2016. doi: 10.1007/978-3-319-49409-8_35. URL https://arxiv.org/abs/1607.01719.

Marco Toldo and Mete Ozay. Bring evanescent representations to life in lifelong class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16732–16741, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Toldo_Bring_Evanescent_Representations_to_Life_in_Lifelong_Class_Incremental_Learning_CVPR_2022_paper.html.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 dataset, 2011.

Hongye Xu and Bartosz Krawczyk. Two-way is better than one: Bidirectional alignment with cycle consistency for exemplar-free class-incremental learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=7UfZAxKo5K.

Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6982–6991, 2020.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 2017. URL https://proceedings.mlr.press/v70/zenke17a.html.

Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Class-incremental learning via dual augmentation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14306–14318, 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/77ee3bc58ce560b86c2b59363281e914-Abstract.html.

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021b.

Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9296–9305, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Zhu_Self-Sustaining_Representation_Expansion_for_Non-Exemplar_Class-Incremental_Learning_CVPR_2022_paper.html.

Huiping Zhuang, Zhenyu Weng, Hongxin Wei, Renchunzi Xie, Kar-Ann Toh, and Zhiping Lin. ACIL: Analytic class-incremental learning with absolute memorization and privacy protection. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11602–11614, 2022. doi: 10.52202/068431-0843. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4b74a42fc81fc7ee252f6bcb6e26c8be-Abstract-Conference.html.

A Proof of Theorem 1

For any random vector with finite second moments,

$$\mathbb{E}[A_c z^- + b_c] = A_c \mu_c^- + b_c, \qquad \mathrm{Cov}(A_c z^- + b_c) = A_c \Sigma_c^- A_c^\top. \tag{11}$$

Thus a necessary condition for covariance matching is $\mathrm{rank}(\Sigma_c^+) \le \mathrm{rank}(\Sigma_c^-)$. The mean can always be matched by choosing $b_c = \mu_c^+ - A_c \mu_c^-$.

To prove sufficiency, let $k = \mathrm{rank}(\Sigma_c^+) \le \mathrm{rank}(\Sigma_c^-)$. For $k > 0$, write the target eigendecomposition as $\Sigma_c^+ = \sum_{i=1}^k \gamma_i u_i u_i^\top$, where $\gamma_i > 0$ and the u_i are orthonormal. Select k positive eigenvalues η_i of Σ_c^- and their orthonormal eigenvectors v_i. Define

$$A_c = \sum_{i=1}^k \sqrt{\frac{\gamma_i}{\eta_i}} u_i v_i^\top, \qquad b_c = \mu_c^+ - A_c \mu_c^-. \tag{12}$$

Since $v_i^\top \Sigma_c^- v_j = \eta_i \mathbf{1}_{i=j}$, substitution gives

$$A_c \Sigma_c^- A_c^\top = \sum_{i=1}^k \gamma_i u_i u_i^\top = \Sigma_c^+. \tag{13}$$

When $k = 0$, choose $A_c = 0$ and $b_c = \mu_c^+$. This covers all positive semidefinite source and target covariances satisfying the rank condition. For $\Sigma_c^- \succ 0$, Equation 2 is the corresponding whitening-and-coloring construction. Finally, $\mathbb{E}[zz^\top] = \mathrm{Cov}(z) + \mathbb{E}[z]\mathbb{E}[z]^\top$, so matching the mean and covariance also matches the second raw moment.

B Moment Accuracy of Fitted Affine Compensation

Fix a task $t > 1$. Let P be the uniform empirical distribution on the n current-task fitting inputs and P_c the input distribution of an old class c. Define $z = f_{t-1}(x)$, $u = f_t(x) - z$, and $\lambda = \rho/n > 0$. Dividing Equation 3 by n gives

$$\min_{W,b} \ \mathbb{E}_P \|z + u - Wz - b\|_2^2 + \lambda \|W - I\|_F^2. \tag{14}$$

All covariances below use their distributional definition, so empirical covariances under P have denominator n. For compactness, write $W = W_t$, $b = b_t$, $B = W - I$, and $k_t = \|B\|_{\mathrm{op}}$.

Choose $\varepsilon > 0$ such that $\mathbb{E}_P \|u\|_2^2 \le \varepsilon^2$. Define the joint feature and update response $\xi = (z^\top, u^\top/\varepsilon)^\top$ and its moment discrepancy

$$\delta_c = \max\left\{ \|\mathbb{E}_{P_c}\xi - \mathbb{E}_P \xi\|_2, \ \|\mathrm{Cov}_{P_c}(\xi) - \mathrm{Cov}_P(\xi)\|_F \right\}. \tag{15}$$

This quantity compares the low-order statistics of the old representation and its response to the feature update. For the true class moments

$$\begin{aligned} \mu_c^- &= \mathbb{E}_{P_c} z, & \Sigma_c^- &= \mathrm{Cov}_{P_c}(z), \\ \mu_c^+ &= \mathbb{E}_{P_c}(z+u), & \Sigma_c^+ &= \mathrm{Cov}_{P_c}(z+u), \end{aligned} \tag{16}$$

define the errors introduced by one affine update as

$$E_{\mu,c} = \|\mu_c^+ - (W\mu_c^- + b)\|_2, \qquad E_{\Sigma,c} = \|\Sigma_c^+ - W\Sigma_c^- W^\top\|_F. \tag{17}$$

Theorem 2 (Moment accuracy of fitted affine compensation)**.** *Assume finite second moments of z and u under P and P_c. The solution of Equation 14 satisfies*

$$W\mathbb{E}_P z + b = \mathbb{E}_P(z+u) \tag{18}$$

and the covariance bound in Equation 6. For each old class c, its transport errors satisfy Equation 7, with δ_c defined by Equation 15.

Current-task mean and covariance. Let $r = z + u - Wz - b = u - Bz - b$ be the fitting residual. The first-order conditions of Equation 14 give

$$\mathbb{E}_P r = 0, \qquad \mathrm{Cov}_P(r, z) = \lambda B. \tag{19}$$

The first identity proves mean matching. Write $C_z = \mathrm{Cov}_P(z)$, $C_u = \mathrm{Cov}_P(u)$, and $C_r = \mathrm{Cov}_P(r)$. Expanding $z + u = Wz + b + r$ yields

$$\begin{aligned} \mathrm{Cov}_P(z+u) - WC_zW^\top &= C_r + W\,\mathrm{Cov}_P(z,r) + \mathrm{Cov}_P(r,z)W^\top \\ &= C_r + \lambda(WB^\top + BW^\top) \\ &= C_r + 2\lambda BB^\top + \lambda(B + B^\top). \end{aligned} \tag{20}$$

Similarly, expanding $u = Bz + b + r$ gives

$$C_u = BC_zB^\top + C_r + 2\lambda BB^\top. \tag{21}$$

Consequently, $A = C_r + 2\lambda BB^\top$ satisfies $0 \preceq A \preceq C_u$. With $v^2 = \mathrm{tr}(C_u) \le \varepsilon^2$,

$$\|A\|_F \le \mathrm{tr}(A) \le v^2. \tag{22}$$

To bound B, let Z and U contain the centered z and u samples, respectively, each divided by $\sqrt{n}$. Then $C_z = ZZ^\top$, $\|U\|_F = v$, and

$$B = UZ^\top(ZZ^\top + \lambda I)^{-1}. \tag{23}$$

If s_j are the eigenvalues of $ZZ^\top$, the singular values of $Z^\top(ZZ^\top + \lambda I)^{-1}$ are $\sqrt{s_j}/(s_j + \lambda)$. Since

$$\sup_{s \ge 0} \frac{\sqrt{s}}{s + \lambda} = \frac{1}{2\sqrt{\lambda}}, \tag{24}$$

we obtain $\|B\|_F \le v/(2\sqrt{\lambda})$ and hence $k_t \le \|B\|_F \le \varepsilon/(2\sqrt{\lambda})$. Applying this bound and Equation 22 to Equation 20 gives

$$\begin{aligned} \|\,\mathrm{Cov}_P(z+u) - WC_zW^\top\|_F &\le v^2 + 2\lambda\|B\|_F \\ &\le v^2 + v\sqrt{\lambda} \le \varepsilon^2 + \varepsilon\sqrt{\lambda}, \end{aligned} \tag{25}$$

which proves Equation 6.

Old-class mean and covariance. The blocks of ξ and its covariance in Equation 15 imply

$$\begin{aligned} \|\mathbb{E}_{P_c} z - \mathbb{E}_P z\|_2 &\le \delta_c, \\ \|\mathbb{E}_{P_c} u - \mathbb{E}_P u\|_2 &\le \varepsilon\delta_c, \\ \|\,\mathrm{Cov}_{P_c}(z) - \mathrm{Cov}_P(z)\|_F &\le \delta_c, \\ \|\,\mathrm{Cov}_{P_c}(z,u) - \mathrm{Cov}_P(z,u)\|_F &\le \varepsilon\delta_c, \\ \|\,\mathrm{Cov}_{P_c}(u) - \mathrm{Cov}_P(u)\|_F &\le \varepsilon^2\delta_c. \end{aligned} \tag{26}$$

The mean identity in Equation 19 gives $\mathbb{E}_P u = B\mathbb{E}_P z + b$. Hence

$$\mu_c^+ - (W\mu_c^- + b) = (\mathbb{E}_{P_c} u - \mathbb{E}_P u) - B(\mathbb{E}_{P_c} z - \mathbb{E}_P z), \tag{27}$$

whose norm is at most $(\varepsilon + k_t)\delta_c$.

For any distribution R, define $H_R = \mathrm{Cov}_R(z+u) - W\,\mathrm{Cov}_R(z)W^\top$. Expanding $W = I + B$ gives

$$\begin{aligned} H_R = {} & \mathrm{Cov}_R(z,u) + \mathrm{Cov}_R(u,z) + \mathrm{Cov}_R(u) \\ & - B\,\mathrm{Cov}_R(z) - \mathrm{Cov}_R(z)B^\top - B\,\mathrm{Cov}_R(z)B^\top. \end{aligned} \tag{28}$$

Subtracting the expressions for $R = P_c$ and $R = P$, and using Equation 26, yields

$$\|H_{P_c} - H_P\|_F \le (2\varepsilon + \varepsilon^2 + 2k_t + k_t^2)\delta_c. \tag{29}$$

Combining this inequality with Equation 6 proves the covariance bound in Equation 7, completing the proof.

Propagation of memory errors. For recursively maintained memory, let $e_{\mu,c}^{t-1} = \|\widehat{\mu}_c^{t-1} - \mu_c^-\|_2$ and $e_{\Sigma,c}^{t-1} = \|\widehat{\Sigma}_c^{t-1} - \Sigma_c^-\|_F$, with $e_{\mu,c}^t$ and $e_{\Sigma,c}^t$ defined analogously against the current true moments. Equation 5 and the triangle inequality give

$$\begin{aligned} e_{\mu,c}^t &\le \|W_t\|_{\mathrm{op}} e_{\mu,c}^{t-1} + E_{\mu,c}, \\ e_{\Sigma,c}^t &\le \|W_t\|_{\mathrm{op}}^2 e_{\Sigma,c}^{t-1} + E_{\Sigma,c}. \end{aligned} \tag{30}$$

Thus the per-task bounds add to the preceding memory errors after affine transport.

B.1 Cross-Class Response and Transport Accuracy

We examine the cross-class discrepancy δ_c and the accuracy of ReFAC's shared maps on CIFAR-100 with ten tasks (Table 6). Each map transports the preceding old-class statistics into the updated feature space.

Cross-class response. The taskwise median of δ_c stays between 16.66 and 19.73. For the first ten classes, it rises from 18.16 at task 2 to 23.60 at task 3, then remains between 22.92 and 24.04 through task 10. The discrepancy primarily reflects existing class geometry: the old-feature covariance block accounts for 99.0% of the squared covariance difference on average. Both summaries of δ_c remain stable after the early tasks.

Transport accuracy. Across all 450 class–task pairs, relative mean and covariance errors average 8.1% and 17.6%. Shared affine compensation lowers both errors at every incremental task, reducing their averages by 48.7% and 40.7% compared with leaving the preceding statistics unchanged. From task 2 to task 10, mean error decreases from 13.6% to 7.9%, and covariance error from 24.8% to 17.6%. Across this sequence, maps fitted on current-task inputs consistently improve old-class moment estimates.

Table 6: Cross-class discrepancy and relative transport errors on CIFAR-100 with ten tasks. Discrepancies are medians over all old classes or the fixed first ten classes. Errors (%; lower is better) are averaged over old classes, with Overall pooling all 450 class–task pairs. No update retains the preceding statistics.

	δ_c		Mean error		Covariance error	
Task	All old	First 10	No update	Affine	No update	Affine
2	18.16	18.16	19.1	13.6	35.2	24.8
3	18.83	23.60	16.5	8.6	27.2	18.3
4	19.41	23.92	15.7	8.3	28.9	17.7
5	19.73	23.57	14.9	7.8	28.0	17.0
6	17.97	23.23	16.2	8.0	29.2	17.4
7	16.66	23.69	15.8	8.4	29.8	17.8
8	18.69	24.04	16.2	7.5	28.6	16.6
9	16.82	22.92	15.0	8.1	29.4	17.6
10	17.65	23.08	16.0	7.9	32.2	17.6
Overall	–	–	15.8	**8.1**	29.7	**17.6**

C Feature Learning under Shared Affine Compensation

Shared affine compensation updates class means and covariances. We examine whether encouraging individual features to follow the same transformation family improves incremental learning. The comparison uses CIFAR-100 with ten tasks and the training schedule in Section 4.1, with a shared class order, initialization, and first-task model.

Direct affine feature regularization. We replace ReFAC's decoder-based recovery objective with a task-shared affine constraint,

$$\mathcal{L}_{\mathrm{aff}} = \frac{1}{|\mathcal{B}|} \sum_{(x,y)\in\mathcal{B}} \|f_t(x) - (A_t f_{t-1}(x) + a_t)\|_2^2, \qquad \mathcal{L}_{\mathrm{direct}} = \mathcal{L}_{\mathrm{ce}} + \eta\mathcal{L}_{\mathrm{aff}} + 0.25\mathcal{L}_{\mathrm{fd}} + \mathcal{L}_{\mathrm{ac}}. \tag{31}$$

The affine head (A_t, a_t) is initialized to $(I, 0)$ for each incremental task and trained jointly with f_t, while f_{t-1} stays frozen. This learner replaces the decoder and its covariance-matching term, retaining the direct feature and anti-collapse penalties. We test $\eta \in \{0.1, 1, 2.5, 10\}$. The affine, sample-recovery, and direct feature losses sum over feature coordinates and average over samples. After training, every learner independently fits the same ridge compensation rule in Equation 4 on current-task feature pairs; this fitted map, rather than the training head, updates its stored class statistics.

Measurements. Alongside A_{last} and A_{inc}, we report accuracy on each task's new-class test images, with predictions over all seen classes. On these held-out images $\mathcal{S}_t$, the normalized residual of the fitted compensation map is

$$R_{\text{aff}}(t) = \frac{\sum_{x \in \mathcal{S}_t} \|f_t(x) - T_t(f_{t-1}(x))\|_2^2}{\sum_{x \in \mathcal{S}_t} \|f_t(x) - \bar{z}_t\|_2^2}, \qquad \bar{z}_t = \frac{1}{|\mathcal{S}_t|} \sum_{x \in \mathcal{S}_t} f_t(x). \tag{32}$$

For each old class, we recompute its preceding and current moments from 450 training images using the evaluation view and a horizontal flip, solely for offline measurement. Transporting the preceding reference moments isolates the error introduced by one update. Mean and covariance errors are normalized by $\sqrt{\operatorname{tr} \Sigma_c^+}$ and $\|\Sigma_c^+\|_F$, respectively. New-class accuracy and all three diagnostic quantities are averaged over tasks 2–10, with old classes weighted equally within each task.

Table 7: Feature-learning strategies with the same affine compensation rule on CIFAR-100 ($T = 10$). Direct affine regularization lowers feature residuals and moment errors but yields lower recognition accuracy. Accuracies are in percent; residuals and errors are dimensionless ratios. A_{new} averages new-class accuracy over tasks 2–10. Bold marks the best value in each column.

	Accuracy ↑			Normalized error ↓		
Feature learning	A_{last}	A_{inc}	A_{new}	R_{aff}	Mean	Covariance
ReFAC	**53.31**	**65.04**	**63.42**	0.0399	0.0912	0.1859
Direct affine, $\eta = 0.1$	49.44	60.79	53.86	0.0131	0.0472	0.1022
Direct affine, $\eta = 1$	39.15	52.60	44.57	0.0019	0.0189	0.0434
Direct affine, $\eta = 2.5$	36.50	50.83	41.24	0.0013	0.0144	0.0331
Direct affine, $\eta = 10$	36.97	50.82	38.63	**0.0010**	**0.0101**	**0.0239**

Transport accuracy and new-class learning. Every direct-affine variant has lower feature residuals and smaller mean and covariance errors than ReFAC at all nine incremental stages, while also having lower new-class and overall accuracy. At $\eta = 0.1$, the best tested direct-affine variant reduces the average feature residual from 0.0399 to 0.0131, but loses 9.57 percentage points in new-class accuracy and 3.87 points in final accuracy (Table 7). Increasing the constraint weight further reduces transport errors while new-class accuracy continues to fall. Thus, making a representation easier to transport does not by itself yield better recognition. The two stages serve different objectives: feature learning must retain useful information while developing discrimination for new classes, whereas memory compensation needs to track the resulting class-level moments. ReFAC combines recovery through a learned decoder with affine updates of these summaries.

D Diagnostic and Timing Protocol

Models and reference statistics. The diagnostic runs in Section 4.2 use ResNet-18 with 64-dimensional features for AdaGauss and BiCyc and 512-dimensional features for EFC and DPCR. EFC's original update changes old-class means, while its affine variant changes both means and covariances. Each EFC variant trains with its own memory, EFM, and classifier. The other three methods use a shared feature-model trajectory with independent recursive memories and their original prediction rules.

After each task $t > 1$, an offline evaluator recomputes reference moments $(\mu_c^{\star,t}, \Sigma_c^{\star,t})$ from old training images. EFC uses one deterministic evaluation view; the other methods use this view and a horizontal flip. DPCR retains its native single-view memory construction and reconstructs the

diagnostic covariance from its stored sample count, mean, and uncentered second moment. All reference covariances use the sample denominator $n_c - 1$. Reference statistics are used solely for diagnostic comparisons.

Error aggregation. The curves measure stored statistics before classifier-specific processing. With $\eta = 10^{-12}$ for numerical protection, we report

$$E_\mu(t) = \frac{1}{|\mathcal{C}_{1:t-1}|} \sum_{c \in \mathcal{C}_{1:t-1}} \frac{\|\widehat{\mu}_c^t - \mu_c^{\star,t}\|_2}{\sqrt{\max(\operatorname{tr} \Sigma_c^{\star,t}, \eta)}},$$
$$E_\Sigma(t) = \frac{1}{|\mathcal{C}_{1:t-1}|} \sum_{c \in \mathcal{C}_{1:t-1}} \frac{\|\widehat{\Sigma}_c^t - \Sigma_c^{\star,t}\|_F}{\max(\|\Sigma_c^{\star,t}\|_F, \eta)}. \tag{33}$$

Thus each class is normalized before averaging, and errors include the history of recursive updates. Task 1 has no old classes, so the curves begin at task 2.

Timing. CUDA timing regions are synchronized. For AdaGauss, BiCyc, and DPCR, post-training update times sum tasks 1–10 and include feature extraction, compensation, new-class statistics, and classifier preparation. EFC's compensation time sums feature-pair extraction, fitting, and application over tasks 2–10. Speedup is the ratio of Original to Affine time within the corresponding scope. Checkpoint I/O and offline diagnostics are excluded.

E Parameter and Storage Costs

ReFAC uses the same ResNet-18 backbone and $512 \to 64$ linear dimensionality reduction as AdaGauss and BiCyc (Rypeść et al., 2024; Xu & Krawczyk, 2026).

Stage-I auxiliary parameters. Feature learning uses one decoder D_t to map current features to the previous feature space. Its two linear layers have dimensions $d \to md \to d$, with biases in both layers, giving

$$P_D = 2md^2 + (m+1)d = 264{,}256, \qquad d = 64, \quad m = 32. \tag{34}$$

The direct feature penalty and minibatch covariance matching introduce no additional trainable parameters or persistent class statistics. ReFAC thus has the same Stage-I auxiliary parameter count as AdaGauss. Relative to BiCyc, it omits an equally sized forward adapter, saving $264{,}256$ trainable parameters, approximately 2.4% of the ResNet-18 backbone's parameter count.

Affine update coefficients. The affine update in Equation 5 uses $W_t \in \mathbb{R}^{d \times d}$ and $b_t \in \mathbb{R}^d$, with

$$P_{\text{transport}} = d^2 + d = 4{,}160 \tag{35}$$

coefficients, occupying 16.25 KiB in FP32. This is a 98.4% reduction relative to the $264{,}256$ parameters of the MLP adapter used for statistics updates in AdaGauss or BiCyc. The map is fitted in closed form without gradient optimization and discarded after the statistics update, adding no persistent storage or inference cost.

Persistent class storage. For C learned classes, ReFAC stores one mean and one full covariance per class, as in AdaGauss. The feature extractor remains fixed in size, and inference uses these statistics without a growing classifier head. The numbers of stored floating-point values under compact and dense covariance storage are

$$M_{\text{compact}} = C\left[d + \frac{d(d+1)}{2}\right], \qquad M_{\text{dense}} = C(d + d^2). \tag{36}$$

The compact form exploits symmetry by storing the upper triangle including the diagonal. At $d = 64$, it requires $2{,}144$ values, or 8.375 KiB per class in FP32. The current implementation stores dense matrices: $4{,}160$ values and 16.25 KiB per class, approximately 1.59 MiB for 100 classes in FP32. ReFAC therefore preserves AdaGauss's class-statistics storage requirement while reducing the size of the compensation module.

F Complete Learning Procedure

Algorithm 1 ReFAC: Recoverable Feature Learning with Affine Compensation

Require: Task datasets $\mathcal{D}_1, \ldots, \mathcal{D}_T$

Initialize feature extractor f_1 and memory $\mathcal{M}_0 = \emptyset$
for $t = 1, \ldots, T$ **do**
    Initialize a temporary classifier g_t for current classes
    if $t > 1$ **then**
        Freeze f_{t-1}; initialize f_t from f_{t-1} and initialize decoder D_t
        Train f_t, g_t, and D_t on $\mathcal{D}_t$ with Equation 9
        Freeze f_t and fit (W_t, b_t) on current-task feature pairs via Equation 4
        Update old-class means and covariances via Equation 5
    else
        Train f_1 and g_1 on $\mathcal{D}_1$ with $\mathcal{L}_{\mathrm{ce}} + \alpha \mathcal{L}_{\mathrm{ac}}$
        Freeze f_1
    end if
    Compute new-class means and covariances from $f_t(\mathcal{D}_t)$; form $\mathcal{M}_t$
    Discard g_t and, if $t > 1$, D_t; retain f_t and $\mathcal{M}_t$
    Predict among all seen classes using Equation 10
end for