Subspace-Guided Continual Learning: Hessian Based Stable–Plastic Decomposition for Exemplar-Free Class-Incremental Learning

Qi Zhu Ziang Gan Wanting Zhang Libao Zhang*
School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China

Abstract

Exemplar-Free Class-Incremental Learning (EFCIL) is a challenging continual learning paradigm where a model must learn new classes sequentially without access to old data, making it susceptible to catastrophic forgetting. The core difficulty lies in balancing stability (preserving old knowledge) and plasticity (acquiring new knowledge). We propose Subspace-Guided Continual Learning (SGCL), a novel method that tackles this dilemma from a geometric perspective. SGCL decomposes the feature space into two orthogonal subspaces: a "stable subspace" containing old-decision-sensitive directions, and a "plastic subspace" where new knowledge can be learned with minimal interference. This decomposition is efficiently identified via the feature-space Hessian, whose nonzero eigendirections define the stable subspace. Building on this, SGCL introduces two synergistic components: 1) Subspace-Guided Regularization (SGR), which imposes curvature-weighted penalties on feature drifts within the stable subspace, and 2) Subspace-Guided Prototype Alignment (SGPA), which adaptively corrects the shift of old-class prototypes to recalibrate the classifier. Extensive experiments on standard benchmarks show that SGCL consistently achieves competitive or superior performance compared to existing state-of-the-art methods, offering a principled approach to mitigating forgetting through loss landscape analysis. Code is available at SGCL.

1 Introduction

Continual learning (CL) addresses the challenge of learning from a continuous stream of data without suffering from catastrophic forgetting [McCloskey and Cohen, 1989, French, 1999]—the tendency of neural networks to abruptly lose knowledge of previously learned tasks. This requires a delicate balance between model stability (preserving old knowledge) and plasticity (acquiring new knowledge), a conflict often referred to as the stability-plasticity dilemma [Grossberg, 2012, Mermillod et al., 2013]. Major CL strategies are broadly categorized into regularization-based, replay-based, and architecture-based methods [De Lange et al., 2021, Masana et al., 2022, Zhou et al., 2023]. Regularization-based methods add constraints to the loss function to prevent drastic weight changes; replay-based methods store and re-train on a small subset of past data; and architecture-based methods dynamically modify or expand the network structure to accommodate new knowledge.

A challenging yet practical paradigm is Exemplar-Free Class-Incremental Learning (EFCIL). In this setting, a model must learn new classes sequentially without storing any data exemplars from past tasks. Furthermore, during inference, privileged information such as task identifiers is unavailable. This constraint is crucial for real-world applications with strict memory budgets or data privacy regulations [Rebuffi et al., 2017, Gomez-Villa et al., 2024]. The absence of past exemplars significantly exacerbates catastrophic forgetting, making it a formidable research problem. We address

*Corresponding author: libaozhang@bnu.edu.cn.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

the cold-start [Magistri et al., 2024] EFCIL scenario, where the model is trained from scratch on an initial set of classes and incrementally updated, with classes evenly distributed across tasks.

Early EFCIL methods focused on parameter space regularization. Kirkpatrick et al. [2017] computes Fisher Information Matrix to identify important parameters, while methods like Wang et al. [2023] use gradient projection to constrain updates within orthogonal subspaces. However, the high dimensionality of parameter space (often millions of parameters) makes accurate importance estimation costly and prone to approximation errors. This has motivated recent advances toward feature space management, where the lower dimensionality enables more precise control. Methods directly operating in feature space [Magistri et al., 2024, Petit et al., 2023, Goswami et al., 2023, Rypeść et al., 2024b] implicitly or explicitly preserve crucial feature dimensions for old tasks. Building on this insight, we propose a principled theoretical framework based on the geometric structure of the classification loss landscape. Our key insight is that the feature-space Hessian of the cross-entropy loss naturally reveals which feature directions are most critical for preserving learned decision boundaries. Under the standard local quadratic model of the old-task loss, large-eigenvalue directions incur the greatest curvature-induced loss increase; we formally prove via the Courant–Fischer theorem that these are the optimal directions to preserve among *all* subspaces of a given dimension (Appendix F). Furthermore, for a K-class linear classifier, the cross-entropy Hessian has rank at most $K - 1$, providing both theoretical justification and computational efficiency for our subspace decomposition. This principled approach separates *stable* directions (nonzero curvature) from *plastic* directions (zero curvature).

Building upon this geometric insight, we propose Subspace-guided Continual Learning (SGCL), a novel EFCIL method that operationalizes the Hessian-based analysis through explicit feature space decomposition. This decomposition enables two synergistic components: a Subspace-Guided Regularization (SGR) loss that applies curvature-weighted penalties within the stable subspace and a weaker uniform penalty in its plastic complement, and a Subspace-Guided Prototype Alignment (SGPA) mechanism that uses the same subspace to modulate prototype updates for drift correction. Together, these components balance stability and plasticity in the demanding cold-start EFCIL setting.

The main contributions of this work are threefold:

- A novel Subspace-Guided Regularization (SGR) strategy that orthogonally decomposes features into stable and plastic subspaces, applying selective regularization to precisely balance stability and plasticity.
- An efficient stable subspace identification algorithm that exploits the Hessian's low-rank structure and adapts the exact decomposition to the smaller of the class and feature dimensions.
- A Subspace-Guided Prototype Alignment (SGPA) mechanism that aligns past prototypes via stability-modulated drift correction, enabling effective classifier calibration.

2 Related Work

2.1 Class-Incremental Learning Methods

Class-Incremental Learning (CIL) methods are designed to learn new classes over time. They are often grouped into three main families [De Lange et al., 2021, Masana et al., 2022, Zhou et al., 2024].

Regularization-based methods introduce additional loss terms to penalize changes to parameters or representations critical for past tasks. Seminal works like EWC [Kirkpatrick et al., 2017], SI [Zenke et al., 2017], and MAS [Aljundi et al., 2018] estimate parameter importance, while feature-level regularization, such as knowledge distillation [Hinton et al., 2015], has proven highly effective, e.g., LwF [Li and Hoiem, 2017], LUCIR [Hou et al., 2019], and PODNet [Douillard et al., 2020].

Replay-based methods store a small subset of past data (exemplars) in a memory buffer to rehearse when learning new tasks [Rebuffi et al., 2017, Castro et al., 2018, Belouadah and Popescu, 2019, Li et al., 2024a]. While highly effective, this approach is not always feasible due to memory or privacy constraints. To circumvent the need for storing real data, some methods employ generative models to create synthetic samples of past data [Shin et al., 2017, Smith et al., 2021].

Architecture-based methods dynamically adapt the model's architecture, for instance by freezing parts of the network and allocating new parameters for new tasks [Mallya and Lazebnik, 2018, Yoon et al., 2017, Rypeść et al., 2024a].

2.2 Exemplar-Free Class-Incremental Learning Methods

Parameter-Space Regularization Early EFCIL approaches focus on constraining parameter updates to preserve learned knowledge. EWC [Kirkpatrick et al., 2017] and its variants compute importance weights via Fisher Information Matrix, penalizing changes to critical parameters. Gradient projection methods [Jin et al., 2021, Wang et al., 2021, 2023, Zhao et al., 2023] take a more restrictive approach, constraining gradient updates to orthogonal subspaces to avoid interference with past tasks. While providing strong theoretical guarantees, these methods suffer from computational complexity in high-dimensional parameter spaces and can be overly restrictive, limiting plasticity and hindering beneficial knowledge transfer [Chaudhry et al., 2020].

Feature-Space Regularization Operating in lower-dimensional feature space enables more precise control over knowledge preservation. Knowledge distillation methods [Hou et al., 2019, Douillard et al., 2020] preserve feature distributions by matching outputs between old and new models. Elastic Feature Consolidation (EFC) [Magistri et al., 2024] identifies important feature directions via an Empirical Feature Matrix (EFM) and applies anisotropic regularization. SGCL uses the exact *feature-space Hessian* to separate its nonzero support from its null space, allowing independent stable–plastic regularization and reuse of this geometry for prototype alignment. Appendix E.2 compares the two regularizers under fixed calibration.

Prototype Drift Correction As feature spaces evolve during continual learning, class prototypes drift from their original positions, causing severe misclassification [Zhu et al., 2021]. FeTrIL [Petit et al., 2023] freezes the feature extractor and translates old prototypes via geometric transformations from new class features. FeCAM [Goswami et al., 2023] employs Mahalanobis distance to account for class covariance structures. LDC [Gomez-Villa et al., 2024] learns explicit mappings between old and new feature spaces, while ADC [Goswami et al., 2024] generates pseudo-samples through adversarial attacks. However, these methods tend to treat prototype drift in isolation, overlooking its intrinsic coupling with feature drift. Our method adopts a unified framework to jointly manage both feature drift and prototype drift through Hessian-based subspace decomposition.

3 Method

In this section, we first present the necessary preliminaries, we then introduce our Subspace-guided method for EFCIL, including Subspace-Guided Regularization (SGR), an efficient stable subspace identification algorithm, and Subspace-Guided Prototype Alignment (SGPA).

3.1 Preliminaries

Problem Formulation We consider the Exemplar-Free Class-Incremental Learning (EFCIL) setting, where a model sequentially learns from T distinct tasks $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T\}$. Each task $\mathcal{T}_t$ contains its own set of classes $\mathcal{C}_t$ and training data $\mathcal{D}_t$. The model consists of two components: a feature extractor (backbone) $f_\theta : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^d$ parameterized by θ, and a classifier head W that expands with each new task. Specifically, at task t, $W_t \in \mathbb{R}^{c_t \times d}$ where $c_t = \sum_{k=1}^{t} |\mathcal{C}_k|$ denotes the total number of classes observed up to task t. The key challenge of EFCIL lies in its strict data access constraint: at time step t, the model can only access the current task's data $\mathcal{D}_t$, while all previous data $\{\mathcal{D}_k\}_{k=1}^{t-1}$ remains completely inaccessible [Wang et al., 2022a, Mai et al., 2022]. Despite this constraint, EFCIL methods aim to approximate the performance of an ideal model trained jointly on all data. This intractable objective serves as a performance upper bound and is formulated as:

$$(\theta_t^*, W_t^*) = \arg\min_{\theta_t, W_t} \sum_{k=1}^{t} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_k} \left[\mathcal{L}_{\text{ce}} \left(W_t f_{\theta_t}(\mathbf{x}), y\right)\right], \tag{1}$$

where $\mathcal{L}_{\text{ce}}$ denotes the cross-entropy loss, $\mathbf{x}$ is an input sample, and y is its class label.

Projection Decomposition of Inner Product Spaces For any feature vector $\mathbf{z} \in \mathbb{R}^d$ and subspace $\mathcal{S} \subseteq \mathbb{R}^d$, the orthogonal decomposition yields:

$$\mathbf{z} = \mathbf{z}_{\mathcal{S}} + \mathbf{z}_{\mathcal{S}^\perp}, \tag{2}$$

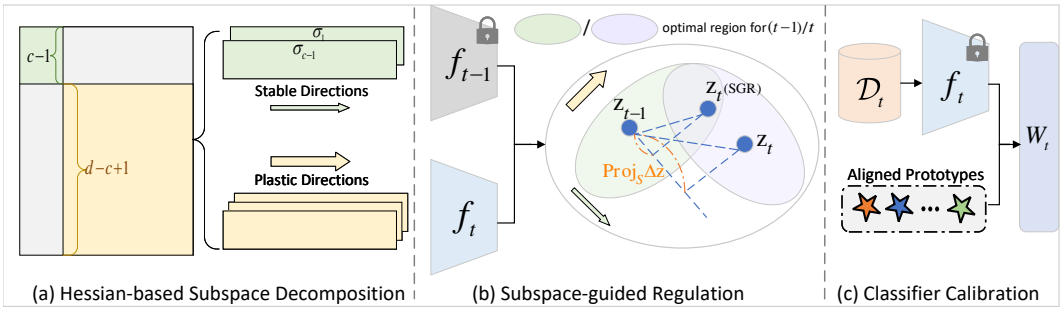


Figure 1: Overview of Subspace-guided Continual Learning (SGCL). (a) Hessian-based subspace decomposition identifies stable directions for preserving past knowledge and plastic directions for learning new tasks. (b) Subspace-guided regulation penalizes feature drift ($\Delta\mathbf{z}$) in stable directions, preventing the new model (f_t) from forgetting knowledge of the old model (f_{t-1}). (c) Classifier calibration updates the classifier (W_t) using aligned prototypes and features from current data ($\mathcal{D}_t$).

where $\mathbf{z}_{\mathcal{S}} = \text{Proj}_{\mathcal{S}}(\mathbf{z})$ and $\mathbf{z}_{\mathcal{S}^\perp} = \mathbf{z} - \mathbf{z}_{\mathcal{S}}$. Given an unit orthonormal basis $\{\mathbf{u}_i\}_{i=1}^k$ for $\mathcal{S}$, the projection is computed as:

$$\text{Proj}_{\mathcal{S}}(\mathbf{z}) = \sum_{i=1}^{k} (\mathbf{z}^\top \mathbf{u}_i)\mathbf{u}_i. \tag{3}$$

This decomposition enables selective regularization on different subspaces.

3.2 Core Principle: Functional Decomposition of Feature Space

Our key assumption is that the feature space can be functionally decomposed into two orthogonal subspaces based on their importance for preserving past knowledge: $\mathbb{R}^d = \mathcal{S} \oplus \mathcal{P}$, where $\mathcal{S}$ is the *stable subspace* containing directions crucial for preserving past knowledge, and $\mathcal{P}$ is the *plastic subspace* providing degrees of freedom for new learning. For convenience, we consider the feature space as the entire d-dimensional space.

Based on this assumption, we identify these subspaces using the feature-space Hessian $\mathbf{H}_f = \nabla^2_{\mathbf{z}} \mathcal{L}_{\text{ce}}$, estimated at the end of task $t-1$ from the then-available data $\mathcal{D}_{t-1}$ and cached for task t. While the Hessian involves second-order derivatives, for cross-entropy loss with feature $\mathbf{z} = f_{\theta_{t-1}}(\mathbf{x})$ and classifier W_{t-1}, it can be analytically computed (see Appendix C for derivation):

$$\mathbf{H}_f = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{t-1}} \left[W_{t-1}^\top (\text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top) W_{t-1} \right], \tag{4}$$

where $\mathbf{p} = \text{softmax}(W_{t-1}\mathbf{z})$. Its rank satisfies:

Proposition 3.1 (Rank of Feature-Space Hessian)**.** *For a c-class classification problem with $W_{t-1} \in \mathbb{R}^{c\times d}$, the rank of $\mathbf{H}_f$ satisfies:*

$$r := \textit{rank}(\mathbf{H}_f) \le \min(\textit{rank}(W_{t-1}), c-1). \tag{5}$$

Moreover, when W_{t-1} has full row rank, we have $r = c-1$.

The stable subspace $\mathcal{S} = \text{Im}(\mathbf{H}_f)$ comprises *all* nonzero eigendirections, while $\mathcal{P} = \text{Ker}(\mathbf{H}_f)$. For the fixed boundary classifier W_{t-1}, $\mathbf{v} \in \mathcal{P}$ iff $W_{t-1}\mathbf{v} \in \text{span}(\mathbf{1})$; thus plastic drift leaves its softmax probabilities and cross-entropy exactly unchanged. Theorem F.1 further establishes optimality under the local quadratic loss surrogate. These statements concern the fixed old head; the expanded and calibrated classifier's accuracy is evaluated empirically. Controlled perturbations and subspace-stability analyses support this geometry (Appendices F.3 and F.4).

3.3 Efficient Stable Subspace Identification

Directly decomposing the $d \times d$ Hessian $\mathbf{H}_f$ is computationally prohibitive. We overcome this by leveraging the low-rank structure: since $\mathbf{H}_f = W_{t-1}^\top \mathbf{A} W_{t-1}$, the stable subspace $\mathcal{S}$ is contained within the row space of W_{t-1}.

Proposition 3.2 (Efficient Subspace Computation)**.** *For $c \leq d$, let the thin QR decomposition be $W_{t-1}^\top = \mathbf{QR}$, with $\mathbf{Q} \in \mathbb{R}^{d\times c}$ having orthonormal columns. The non-zero eigenvalues of $\mathbf{H}_f$ can be found by decomposing the reduced Hessian $\mathbf{H}_{red} \in \mathbb{R}^{c\times c}$:*

$$\mathbf{H}_{red} = \mathbf{RAR}^\top, \tag{6}$$

where $\mathbf{A} = \mathbb{E}_{x\sim\mathcal{D}_{t-1}}[diag(\mathbf{p}) - \mathbf{pp}^\top]$.

1. ***Equivalence:*** *The set of non-zero eigenvalues of $\mathbf{H}_f$ is identical to the set of non-zero eigenvalues of $\mathbf{H}_{red}$. If $(\sigma, \mathbf{u})$ is an eigen-pair of $\mathbf{H}_{red}$, then $(\sigma, \mathbf{Qu})$ is a corresponding eigen-pair of $\mathbf{H}_f$.*

2. ***Efficiency:*** *The computational complexity of finding $\mathcal{S}$ via $\mathbf{H}_{red}$ is $O(dc^2 + c^3)$. This avoids forming and decomposing the full $d \times d$ Hessian $\mathbf{H}_f$, an operation with complexity $O(d^2c + d^3)$, making our method substantially more efficient for $c \ll d$.*

This proposition enables efficient extraction of the stable subspace without expensive full eigen-decomposition. Let $\{(\sigma_i, \mathbf{u}_{r,i})\}_{i=1}^r$ be the non-zero eigen-pairs from $\mathbf{H}_{red}$, where r is defined in Proposition 3.1. The stable subspace basis vectors are $\mathbf{u}_i = \mathbf{Qu}_{r,i}$ with corresponding curvature weights σ_i. Summarizing, the stable subspace $\mathcal{S}$ is explicitly computed as:

$$\mathcal{S} = \text{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_r\}, \quad \mathbf{u}_i = \mathbf{Qu}_{r,i}, \tag{7}$$

Figure 2 illustrates this reduction. When $c > d$, we instead decompose the exact $d \times d$ Hessian. This dimension-adaptive computation uses $m = \min(c, d)$, with complexity $O(cdm + m^3)$, and preserves the same subspace and objective.

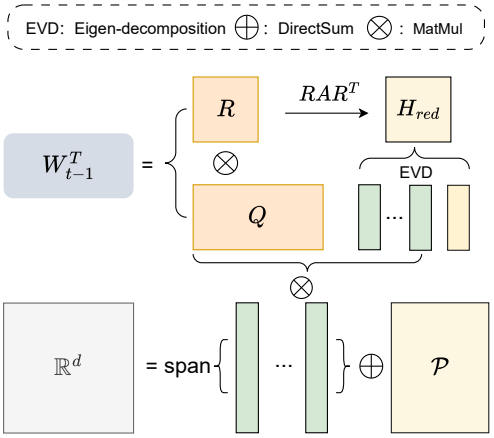


Figure 2: Efficient computation

3.4 Subspace-Guided Regularization(SGR)

Once the stable subspace $\mathcal{S}$ is identified, we can apply differentiated regularization to feature drift $\Delta\mathbf{z} = f_\theta(\mathbf{x}) - f_{\theta_{t-1}}(\mathbf{x})$. Our strategy heavily penalizes drift within $\mathcal{S}$ while allowing flexibility in the plastic subspace $\mathcal{P}$.

Stable Subspace Regularization. We heavily penalize projections of the drift onto stable directions, weighted by their corresponding eigenvalues σ_i:

$$\mathcal{L}_{\text{stable}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}\left[\lambda_s \left\|\text{Proj}_{\mathcal{S}}(\Delta\mathbf{z})\right\|_{\mathbf{H}_f}^2\right] = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}\left[\lambda_s \sum_{i=1}^{r} \sigma_i \|\text{Proj}_{\mathbf{u}_i}(\Delta\mathbf{z})\|_2^2\right], \tag{8}$$

where $\|\cdot\|_{\mathbf{H}_f}^2$ denotes the weighted norm with eigenvalues σ_i.

Plastic Subspace Regularization. For the plastic subspace, we apply minimal uniform regularization:

$$\mathcal{L}_{\text{plastic}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}\left[\lambda_p \left\|\text{Proj}_{\mathcal{P}}(\Delta\mathbf{z})\right\|_2^2\right] = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}\left[\lambda_p \left\|\Delta\mathbf{z} - \sum_{i=1}^{r} \text{Proj}_{\mathbf{u}_i}(\Delta\mathbf{z})\right\|_2^2\right], \tag{9}$$

where $\lambda_p \ll \lambda_s$. The key difference is that stable regularization uses curvature-weighted penalties (σ_i) to preserve critical directions, while plastic regularization applies uniform minimal constraints. Then the total loss for feature adaptation combines the SGR penalties with the standard cross-entropy loss on the current task's data:

$$\mathcal{L}_{\text{adapt}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}\left[\mathcal{L}_{\text{ce}}(W_t f_\theta(\mathbf{x}), y)\right] + \mathcal{L}_{\text{stable}} + \mathcal{L}_{\text{plastic}}. \tag{10}$$

3.5 Subspace-Guided Prototype Alignment (SGPA)

Feature extractor updates cause past class prototypes $\{\mathbf{p}_i^{t-1}\}_{i=1}^{c_{t-1}}$ to become misaligned. SGPA addresses this by aligning the prototypes(subspace-guided) and then calibrating the classifier.

Stability-Modulated Prototype Alignment. We correct prototype drift based on each prototype's stability. First, the average feature drift, $\Delta\mathbf{p}_{\text{drift}}$, is estimated from the current task's data: $\Delta\mathbf{p}_{\text{drift}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}[f_{\theta_t}(\mathbf{x}) - f_{\theta_{t-1}}(\mathbf{x})]$. Then, reusing the stable subspace $\mathcal{S}$ identified in SGR, we compute a stability score S_i for each prototype as its normalized projection magnitude onto $\mathcal{S}$:

$$S_i = \frac{\|\text{Proj}_{\mathcal{S}}(\mathbf{p}_{t-1}^i)\|_2^2}{\|\mathbf{p}_{t-1}^i\|_2^2} = \frac{\sum_{j=1}^{r}(\mathbf{p}_{t-1}^i \cdot \mathbf{u}_j)^2}{\|\mathbf{p}_{t-1}^i\|_2^2}. \tag{11}$$

High S_i indicates the prototype lies in high-curvature directions critical for past tasks. The alignment scales drift by plasticity $(1 - S_i)$:

$$\mathbf{p}_t^i = \mathbf{p}_{t-1}^i + (1 - S_i) \cdot \Delta\mathbf{p}_{\text{drift}}. \tag{12}$$

Classifier Calibration. With aligned prototypes $\{\mathbf{p}_t^i\}$, we retrain the classifier head. To prevent catastrophic forgetting, we generate synthetic features for past classes by sampling from Gaussian distributions centered at these aligned prototypes. Let $\mathcal{P}_{\text{old}} = \{(\mathbf{p}_t^i, y_i)\}_{i=1}^{c_{t-1}}$ be the set of aligned prototypes and their labels for past classes. The calibration loss is:

$$\mathcal{L}_{\text{calib}} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}\left[\mathcal{L}_{\text{ce}}(W_t f_{\theta_t}(\mathbf{x}), y)\right] + \mathbb{E}_{\substack{\mathbf{p}_t^i, y_i\sim\mathcal{P}_{\text{old}} \\ \hat{\mathbf{z}}\sim\mathcal{N}(\mathbf{p}_t^i, \mathbf{\Sigma}_i)}}\left[\mathcal{L}_{\text{ce}}(W_t\hat{\mathbf{z}}, y_i)\right], \tag{13}$$

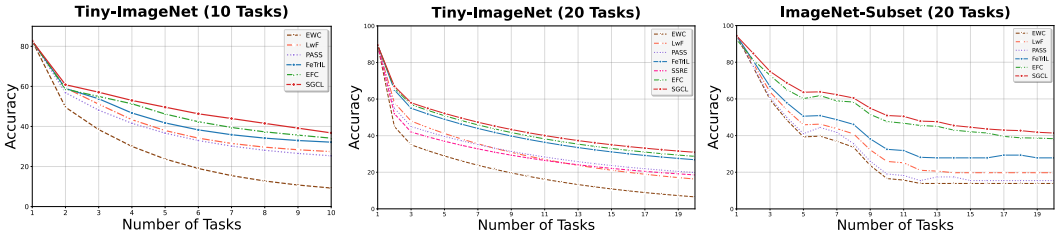
where $\mathbf{\Sigma}_i$ denotes the covariance estimate associated with class i. The detailed algorithmic procedure is provided in Appendix B.

Full-rank regime. When $r = d$, which can occur for $c > d$, $\mathcal{P} = \{0\}$ and $\mathcal{L}_{\text{plastic}} = 0$. For nonzero prototypes, $S_i = 1$, so SGPA leaves them unchanged. SGCL then uses full-space curvature-weighted regularization, current-task learning, and classifier calibration. Appendix E.4 documents this regime on ImageNet-1K.

4 Experimental Results

In this section, we first present the experimental setups, compare SGCL's performance against state-of-the-art EFCIL methods, and finally provide a detailed analysis to validate our approach.

4.1 Experimental Settings



(a) Tiny-ImageNet (N = 10) (b) Tiny-ImageNet (N = 20) (c) ImageNet-Subset (N = 20)

Figure 3: Performance comparison during sequential training on (a) Tiny-ImageNet (N = 10), (b) Tiny-ImageNet (N = 20), and (c) ImageNet-Subset (N = 20).

Datasets and Metrics We evaluate our method on three standard benchmarks: CIFAR-100 [Krizhevsky et al., 2009], Tiny-ImageNet [Wu et al., 2017], and ImageNet-Subset [Deng et al., 2009]. Following the Cold Start EFCIL protocol, classes are split uniformly across tasks. For CIFAR-100 and ImageNet-Subset, we use configurations of 10 tasks with 10 classes each and 20 tasks with 5 classes each. For Tiny-ImageNet, we use 10 tasks with 20 classes and 20 tasks with 10 classes. We report two primary metrics: (1) **Average Accuracy (Acc)**, also known as Last Accuracy,

Table 1: Performance comparison of SGCL against other EFCIL methods across CIFAR-100, Tiny-ImageNet, and ImageNet-Subset under 10-task and 20-task configurations. We report the average accuracy (Acc) and average anytime accuracy (AAA) in percent (%). The best results are **bolded**, and second-best results are underlined.

Method	CIFAR-100				Tiny-ImageNet				ImageNet-Subset			
	10 Tasks		20 Tasks		10 Tasks		20 Tasks		10 Tasks		20 Tasks	
	Acc	AAA	Acc	AAA	Acc	AAA	Acc	AAA	Acc	AAA	Acc	AAA
EWC	32.35	49.14	18.72	31.02	9.25	24.01	6.55	15.70	25.90	39.40	13.89	26.95
LwF	33.95	55.20	18.75	38.39	27.45	45.14	16.30	32.94	38.95	56.41	19.75	40.23
PASS	31.75	47.86	18.65	32.86	25.35	39.25	19.85	32.01	27.65	45.74	15.45	31.65
FeTrIL	35.80	51.20	24.50	38.48	32.15	45.60	26.85	39.54	37.35	52.63	27.85	42.43
SSRE	31.65	47.26	18.75	32.45	24.15	38.82	18.55	30.62	26.65	43.76	17.45	31.15
EFC	43.95	58.58	32.15	47.70	34.45	47.95	28.69	42.07	47.38	60.30	35.75	49.92
ADC	46.48	61.35	35.13	47.56	32.32	43.04	21.33	37.80	46.58	67.07	30.83	49.23
LDC	45.40	59.50	36.85	48.87	34.20	46.80	24.95	40.33	51.40	69.40	31.52	50.60
DPCR	49.06	62.76	**37.98**	49.77	35.01	47.48	26.85	38.65	49.94	67.23	36.06	51.29
SGCL	**49.68**	**62.88**	37.23	**49.80**	**36.78**	**48.72**	**30.92**	**45.51**	**53.52**	**69.54**	**41.44**	**55.60**

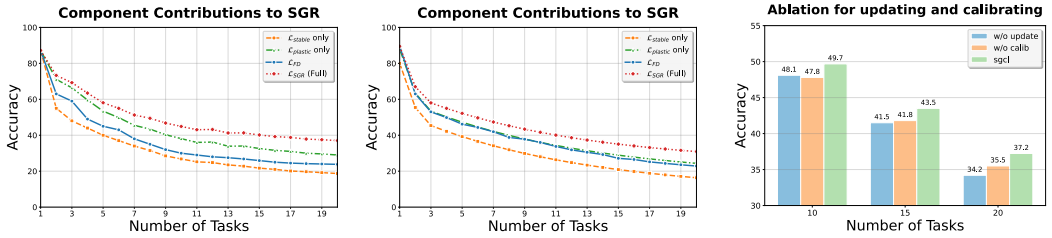
measures the average accuracy on all seen classes after the final task; (2) **Average Anytime Accuracy (AAA)**, equivalent to Average Incremental Accuracy, evaluates the average performance throughout the entire learning process. Specifically, for CIFAR-100 and Tiny-ImageNet, each class contains 500 training samples, with 100 and 50 test samples respectively. For ImageNet-Subset, we follow the protocol established by Douillard et al. [2020], sampling 100 classes from ImageNet-1K, where each class contains approximately 1,300 training samples and 50 test samples. Additionally, we include ImageNet-1K [Deng et al., 2009] and CUB200 [Wah et al., 2011] results in Appendix E.4.

Competing Methods and Implementation Details. We compare SGCL against a comprehensive set of baselines covering classic regularization (EWC, LwF), feature-space management (PASS, SSRE, EFC), and modern prototype-based methods (FeTrIL, LDC, ADC). Replay-based methods are excluded as they do not fit the EFCIL problem definition. For all experiments, we use a ResNet-18 [He et al., 2016] backbone with a batch size of 64. The first task is trained for 100 epochs (160 for ImageNet-Subset), and subsequent tasks for 100 epochs. For CIFAR-100 and Tiny-ImageNet, we use an SGD optimizer with momentum 0.9; the learning rate is 0.1 for the first task and 0.005 for subsequent tasks. For ImageNet-Subset, the first task is trained with SGD (learning rate 0.1, momentum 0.9), while subsequent tasks use an Adam optimizer with learning rates of 1×10^{-5} for the backbone and 1×10^{-4} for the classifier, with a weight decay of 5×10^{-4}. For classifier calibration, we train the classifier for 30 epochs on both prototype and current task features using SGD with a learning rate of 1×10^{-3}, and batch size 256. The regularization weights (λ_s, λ_p) are set to $(5, 0.03)$, $(10, 0.03)$, and $(20, 0.1)$ for CIFAR-100, Tiny-ImageNet, and ImageNet-Subset, respectively. The selection of these hyperparameters is detailed in Appendix E.3. All experiments were conducted on a single NVIDIA RTX 3090Ti GPU.

4.2 Comparison with State-of-the-art

We compare SGCL against a range of strong EFCIL baselines, including classic regularization methods like EWC [Kirkpatrick et al., 2017] and LwF [Li and Hoiem, 2017], as well as recent state-of-the-art approaches such as PASS [Zhu et al., 2021], FeTrIL [Petit et al., 2023], SSRE [Zhu et al., 2022], EFC [Magistri et al., 2024], ADC [Goswami et al., 2024], LDC [Gomez-Villa et al., 2024], and DPCR [He et al., 2025]. Table 1 and Figure 3 show highly competitive performance. SGCL achieves final accuracies of 49.68%, 36.78%, and 53.52% on the three 10-task benchmarks, and 41.44% on ImageNet-Subset/20T versus DPCR's 36.06%. Three-seed comparisons (Appendix E.6) show that SGCL and DPCR perform comparably on CIFAR-100, with gaps within seed-level variation, while SGCL improves mean Acc by 1.77–5.72 percentage points on Tiny-ImageNet and ImageNet-

Subset. Additional comparisons with FCS and Adam-NSCL under the cold-start protocol are given in Appendix E.8.



(a) SGR on CIFAR-100 (b) SGR on Tiny-ImageNet (c) SGPA and Classifier Calibration

Figure 4: Ablation study of SGCL components on CIFAR-100 and Tiny-ImageNet (20 tasks).

4.3 Ablation Study

To validate the effectiveness of our proposed components, we conduct comprehensive ablation studies on different datasets. Figure 4 shows the results of our ablation studies on CIFAR-100 and Tiny-ImageNet with 20 tasks each.

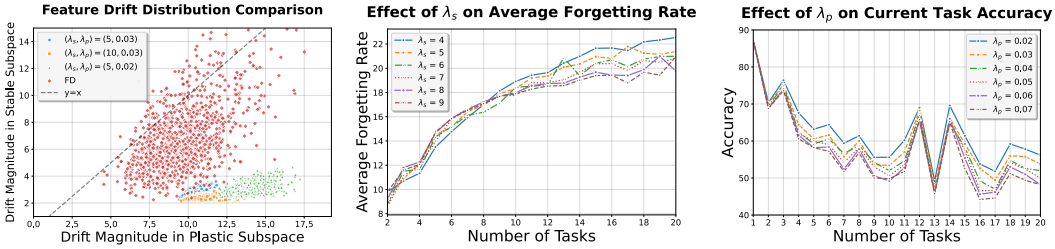
Components of SGR. To isolate the contribution of our Subspace-Guided Regularization, we use Feature Distillation (FD) baseline that uniformly penalizes feature drift across all dimensions, and conducted ablation studies on CIFAR-100 and Tiny-ImageNet (Figures 4a and 4b). We compare this baseline against variants with only stable regularization ($\mathcal{L}_{\text{stable}}$), only plastic regularization ($\mathcal{L}_{\text{plastic}}$), and the full SGR. As shown in the figures, applying only the plastic regularization term yields a greater performance uplift over the baseline compared to applying only the stable term. This is consistent with the fact that the plastic subspace has a much higher dimensionality ($\geq d - c_{t-1} + 1$) than the stable subspace ($\leq c_{t-1} - 1$), giving it a larger influence on the feature space. However, the complete SGR model, which combines both terms, achieves the best performance, consistently outperforming both the baseline and the partial variants across both datasets. This confirms that both components are necessary and that their combination effectively balances stability and plasticity.

Impact of SGPA and Classifier Calibration. We validate the contributions of Subspace-Guided Prototype Alignment (SGPA) and classifier calibration in Figure 4c, comparing our full model against versions without prototype alignment (w/o update, which trains the classifier using outdated prototypes) and without calibration (w/o calib). Removing either component degrades performance in these settings, particularly in longer sequences where prototype drift accumulates. A controlled cross-ablation further shows that replacing EFC regularization with SGR improves Acc by 2.40–5.23 points at fixed calibration; prototype-update controls are reported alongside it in Appendix E.2.

4.4 Analysis

Visualization of Subspace-Guided Drift Control We visualize feature drift for data from the initial task ($\mathcal{T}_0$) after training on all later CIFAR-100 tasks (10 tasks). The drift vector, $\Delta\mathbf{z} = f_{\theta_9}(\mathbf{x}) - f_{\theta_0}(\mathbf{x})$, is projected onto the stable subspace ($\mathcal{S}$) defined after $\mathcal{T}_0$ and its plastic complement. Figure 5a compares the projection magnitudes for SGCL under several hyperparameter settings with Feature Distillation (FD), which applies a uniform penalty across all feature dimensions. Under FD, drift is nearly isotropic and scattered around the $y = x$ axis. In contrast, SGCL makes drift strongly anisotropic: changes concentrate in the plastic subspace while stable components remain small. Increasing λ_s from 5 to 10 further suppresses stable-subspace drift, whereas decreasing λ_p from 0.03 to 0.02 permits larger plastic-subspace changes. This pattern shows that SGR protects old-task knowledge through targeted, controllable drift control.

Plasticity and Stability Analysis Our method uses hyperparameters λ_p and λ_s to independently control plasticity and stability, demonstrating a clear trade-off on CIFAR-100 (20 tasks). Plasticity is governed by λ_p through regularization of the plastic subspace; a smaller value allows for better adaptation and higher current-task accuracy (Figure 5c). Stability is managed by λ_s, which penalizes drift in the stable subspace. We measure knowledge retention using the average forgetting rate—the average accuracy drop on past tasks after training on a new one. A stronger penalty with a higher



(a) Feature Drift Comparison (b) λ_s vs. Average Forgetting (c) λ_p vs. Current Task Accuracy

Figure 5: Analysis of feature drift and hyperparameters: (a) Feature drift comparison between methods, (b) Effect of λ_s on stability, and (c) Effect of λ_p on plasticity.

λ_s mitigates catastrophic forgetting, reflected in a lower average forgetting rate (Figure 5b). These results confirm that our parameters provide a structured mechanism to navigate the stability-plasticity dilemma.

Why $\lambda_p > 0$ is Necessary. Since λ_p governs plasticity, one might expect that setting $\lambda_p = 0$ would maximize adaptation. However, Figure 4a shows that this leads to severe performance degradation. This is not a mistake: because the stable subspace is low-dimensional, constraining only it is insufficient to prevent overall feature drift. The plastic subspace term with $\lambda_p > 0$ serves a dual role—controlling plasticity while also regularizing the high-dimensional complement. As shown in Table 2, replacing it with a global drift penalty ($\|\Delta \mathbf{z}\|_2^2$) still yields reasonable performance. In fact, many regularization methods [Magistri et al., 2024] still rely on such global terms, but SGCL achieves this naturally through λ_p without an additional penalty.

Computational Efficiency We analyze the computational and memory (RAM) overhead of our regularization term against EFC [Magistri et al., 2024], a highly efficient feature-space regularization method. Over the 20-task learning process (100 epochs per task), SGCL shows substantial gains (Table 3), reducing total GFLOPS by 3.7× to 7.5× and requiring significantly less peak memory (e.g., 0.19 MB vs. 44.6 MB on CIFAR-100). This advantage stems from our efficient subspace identification algorithm, which operates on a much smaller reduced matrix, avoiding the large $d \times d$ matrices required by EFC. These memory figures exclude class statistics and model copies. Complete persistent statistics require $4(Cd + Cd^2 + dr + r)$ bytes in FP32: 100.39 MiB for CIFAR-100/ImageNet-Subset and 200.78 MiB for Tiny-ImageNet. Covariances dominate this storage; Appendix E.7 gives the breakdown and model-state accounting.

Table 2: CIFAR-100 drift control.

Strategy	Acc	
	10-task	20-task
No plastic ($\lambda_p = 0$)	28.77	19.86
Global drift ($\|\Delta \mathbf{z}\|_2^2$)	48.73	35.69
Plastic subspace (Ours)	**49.68**	**37.23**

Table 3: 20-task regularization cost.

Dataset	GFLOPS		Memory (MB)	
	EFC	SGCL	EFC	SGCL
CIFAR-100	1,870.32	249.04	44.6	0.19
Tiny-ImageNet	3,693.89	983.70	44.6	0.38
ImageNet-Subset	4,769.32	635.04	44.6	0.19

5 Conclusion

We introduced Subspace-Guided Continual Learning (SGCL), which combines Hessian-guided feature regularization and prototype alignment for exemplar-free class-incremental learning. Its loss-sensitive geometry supports controllable stability and plasticity, with competitive results across convolutional and pretrained transformer settings.

Limitations. SGCL uses a small validation search; reliable automatic coefficient selection remains open. When the Hessian becomes full rank, the plastic null space and SGPA correction become inactive, leaving curvature-weighted regularization and classifier calibration. The invariance result applies to the fixed old classifier, and the preservation optimality concerns its quadratic loss surrogate. These boundaries motivate further work on adaptive regularization and richer prototype-drift models.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 62271060 and 61571050, in part by the Beijing Natural Science Foundation under Grant 4222046. The authors thank the anonymous reviewers for their valuable feedback on earlier versions of this manuscript. The authors declare no competing interests.

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 (ECCV)*, pages 139–154, 2018.

Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 583–592, 2019.

Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018.

Arslan Chaudhry, Naeemullah Khan, Puneet Dokania, and Philip Torr. Continual learning in low-rank orthogonal subspaces. *Advances in Neural Information Processing Systems*, 33:9900–9911, 2020.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

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

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.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pages 86–102. Springer, 2020.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 (4):128–135, 1999.

Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D Bagdanov, Bartlomiej Twardowski, and Joost van de Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *European Conference on Computer Vision*, pages 473–490. Springer, 2024.

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, Bart Twardowski, Joost Van De Weijer, et al. Resurrecting old classes with new data for exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28525–28534, 2024.

Stephen T Grossberg. *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control*, volume 70. Springer Science & Business Media, 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

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. *arXiv preprint arXiv:2503.05423*, 2025.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021b.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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*, pages 831–839, 2019.

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

Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. *Advances in Neural Information Processing Systems*, 34: 29193–29205, 2021.

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

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

Jiyong Li, Dilshod Azizov, Shangsong Liang, et al. Contrastive continual learning with importance sampling and prototype-instance relation distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13554–13562, 2024a.

Qiwei Li, Yuxin Peng, and Jiahuan Zhou. FCS: Feature calibration and separation for non-exemplar class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28495–28504, 2024b.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Simone Magistri, Tomaso Trinci, Albin Soutif-Cormerais, Joost van de Weijer, and Andrew D Bagdanov. Elastic feature consolidation for cold start exemplar-free incremental learning. *arXiv preprint arXiv:2402.03917*, 2024.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

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, 2022.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013.

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*, pages 3911–3920, 2023.

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*, pages 2001–2010, 2017.

Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzciński, Bartosz Zieliński, and Bartłomiej Twardowski. Divide and not forget: Ensemble of selectively trained experts in continual learning. *arXiv preprint arXiv:2401.10191*, 2024a.

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, 2024b.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

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*, pages 9374–9384, 2021.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. CODA-Prompt: COntinual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, 2023.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pages 631–648. Springer, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022b.

Jiayu Wu, Qixiang Zhang, and Guoxi Xu. Tiny imagenet challenge. *Technical report*, 2017.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable Decoupled Low-Rank Adaptation for Class Incremental Learning. In *International Conference on Learning Representations*, 2025.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

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*, pages 6982–6991, 2020.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017.

Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3718–3727, 2023.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey. *arXiv preprint arXiv:2302.03648*, 1(2):6, 2023.

Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. *arXiv preprint arXiv:2401.16386*, 2024.

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*, pages 5871–5880, 2021.

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*, pages 9296–9305, 2022.

A Broader Impacts

SGCL supports continual model updates without retaining raw past examples. However, retained feature statistics do not provide a formal privacy guarantee, and the learned model may inherit biases present in the training data. Deployment therefore requires application-specific evaluation of privacy risks and performance across relevant groups.

B Detailed Algorithmic Procedure

The reference backbone $f_{\theta_{t-1}}$ and boundary eigensystem remain fixed during task t, while the expanded classifier W_t is optimized and calibrated. The fixed old head in the invariance analysis is W_{t-1}, not this updated W_t. At each task boundary, the new eigensystem replaces the previous cache; intermediate Hessian and QR matrices are discarded.

Algorithm 1 Detailed SGCL training procedure

Require: Tasks $\{\mathcal{D}_t\}_{t=1}^{T}$; model f_{θ_0}; classifier W_0.
Train θ_1, W_1 on $\mathcal{D}_1$ with $\mathcal{L}_{\text{ce}}$.
$\mathcal{P}_1 \leftarrow \textsc{ComputeProto\&Cov}(f_{\theta_1}, \mathcal{D}_1)$.
Cache $\mathcal{S}_1 \leftarrow \textsc{BoundaryEigen}(W_1, f_{\theta_1}, \mathcal{D}_1)$ (defined below).
for $t = 2, \ldots, T$ **do**
  Freeze reference $f_{\theta_{t-1}}$; initialize $\theta_t \leftarrow \theta_{t-1}$ and W_t by extending W_{t-1} for $\mathcal{C}_t$.
  Update θ_t, W_t on $\mathcal{D}_t$ using $\mathcal{L}_{\text{adapt}}$ (10)
  $\Delta\mathbf{p}_{\text{drift}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}[f_{\theta_t}(\mathbf{x}) - f_{\theta_{t-1}}(\mathbf{x})]$.
  for each $(\mathbf{p}_{t-1}^i, \mathbf{\Sigma}^i) \in \mathcal{P}_{t-1}$ **do**
    $S_i = \|\text{Proj}_{\mathcal{S}_{t-1}}(\mathbf{p}_{t-1}^i)\|_2^2 / \|\mathbf{p}_{t-1}^i\|_2^2$ (11)
    $\mathbf{p}_t^i = \mathbf{p}_{t-1}^i + (1 - S_i) \cdot \Delta\mathbf{p}_{\text{drift}}$ (12)
  end for
  $\mathcal{P}_{\text{old}} \leftarrow \{(\mathbf{p}_t^i, \mathbf{\Sigma}^i)\}$.
  Retrain W_t on $\mathcal{P}_{\text{old}}$ and $\mathcal{D}_t$ with $\mathcal{L}_{\text{calib}}$ (13)
  $\mathcal{P}_t \leftarrow \mathcal{P}_{\text{old}} \cup \textsc{ComputeProto\&Cov}(f_{\theta_t}, \mathcal{D}_t)$.
  $\mathcal{S}_t \leftarrow \textsc{BoundaryEigen}(W_t, f_{\theta_t}, \mathcal{D}_t)$; replace the previous eigensystem.
end for

BOUNDARYEIGEN computes the nonzero eigenpairs of Equation (4) using the current boundary model and data. For $c_t \leq d$, it uses $W_t^\top = \mathbf{Q}_t\mathbf{R}_t$, decomposes $\mathbf{R}_t\mathbf{A}_t\mathbf{R}_t^\top$, and lifts eigenvectors with $\mathbf{Q}_t$. For $c_t > d$, it decomposes the exact $d \times d$ feature Hessian directly. Both branches return only the basis and eigenvalues needed by SGR and SGPA.

C Derivation of the Feature-Space Hessian

We derive the analytical form of the feature-space Hessian for the cross-entropy loss. Let $\mathcal{L}_{\text{ce}}$ be the cross-entropy loss for a single sample $(\mathbf{x}, y)$, where y is the ground-truth label. The model consists of a feature extractor $\mathbf{z} = f_\theta(\mathbf{x})$ and a linear classifier $W \in \mathbb{R}^{c\times d}$.

The logits are given by $\mathbf{a} = W\mathbf{z}$. The predicted probabilities are computed using the softmax function, $\mathbf{p} = \text{softmax}(\mathbf{a})$, where $p_i = \frac{e^{a_i}}{\sum_{j=1}^{c} e^{a_j}}$. The cross-entropy loss is then:

$$\mathcal{L}_{\text{ce}} = -\log p_y = -a_y + \log\left(\sum_{j=1}^{c} e^{a_j}\right). \tag{14}$$

We are interested in the Hessian of this loss with respect to the feature vector $\mathbf{z}$, i.e., $\mathbf{H}_\mathbf{z} = \nabla_\mathbf{z}^2\mathcal{L}_{\text{ce}} = \frac{\partial^2\mathcal{L}_{\text{ce}}}{\partial\mathbf{z}\partial\mathbf{z}^\top}$.

First, we compute the gradient of the loss with respect to the logits $\mathbf{a}$. For the i-th component of the logits, we have:

$$\frac{\partial \mathcal{L}_{\text{ce}}}{\partial a_i} = \frac{\partial}{\partial a_i}\left(-a_y + \log\left(\sum_{j=1}^{c} e^{a_j}\right)\right) = -\delta_{iy} + \frac{e^{a_i}}{\sum_{j=1}^{c} e^{a_j}} = p_i - \delta_{iy}, \tag{15}$$

where δ_{iy} is the Kronecker delta. In vector form, this is $\nabla_{\mathbf{a}}\mathcal{L}_{\text{ce}} = \mathbf{p} - \mathbf{e}_y$, where $\mathbf{e}_y$ is the one-hot vector for the label y.

Next, we apply the chain rule to find the gradient with respect to the features $\mathbf{z}$:

$$\nabla_{\mathbf{z}}\mathcal{L}_{\text{ce}} = \frac{\partial \mathbf{a}^\top}{\partial \mathbf{z}} \nabla_{\mathbf{a}}\mathcal{L}_{\text{ce}} = W^\top(\mathbf{p} - \mathbf{e}_y). \tag{16}$$

To compute the Hessian, we differentiate the gradient $\nabla_{\mathbf{z}}\mathcal{L}_{\text{ce}}$ with respect to $\mathbf{z}^\top$:

$$\mathbf{H}_{\mathbf{z}} = \frac{\partial}{\partial \mathbf{z}^\top}\left(W^\top(\mathbf{p} - \mathbf{e}_y)\right) = W^\top \frac{\partial \mathbf{p}}{\partial \mathbf{z}^\top}. \tag{17}$$

We need the Jacobian of the probability vector $\mathbf{p}$ with respect to the features $\mathbf{z}$. Using the chain rule again:

$$\frac{\partial \mathbf{p}}{\partial \mathbf{z}^\top} = \frac{\partial \mathbf{p}}{\partial \mathbf{a}^\top}\frac{\partial \mathbf{a}}{\partial \mathbf{z}^\top}. \tag{18}$$

The Jacobian of the logits $\mathbf{a} = W\mathbf{z}$ with respect to $\mathbf{z}$ is simply $\frac{\partial \mathbf{a}}{\partial \mathbf{z}^\top} = W$. The Jacobian of the softmax function is a standard result:

$$\frac{\partial p_i}{\partial a_j} = p_i(\delta_{ij} - p_j). \tag{19}$$

In matrix form, this Jacobian is $\frac{\partial \mathbf{p}}{\partial \mathbf{a}^\top} = \text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top$.

Combining these results, we get:

$$\frac{\partial \mathbf{p}}{\partial \mathbf{z}^\top} = (\text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top)W. \tag{20}$$

Substituting this back into the expression for the Hessian $\mathbf{H}_{\mathbf{z}}$ gives the Hessian for a single sample:

$$\mathbf{H}_{\mathbf{z}} = W^\top(\text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top)W. \tag{21}$$

The full feature-space Hessian $\mathbf{H}_f$ is the expectation of this quantity over the data distribution $\mathcal{D}_{t-1}$, which gives the expression in Equation (4):

$$\mathbf{H}_f = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{t-1}}\left[W_{t-1}^\top(\text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top)W_{t-1}\right]. \tag{22}$$

D Proof of Propositions

D.1 Proof of Proposition 1

Proof. We provide an intuitive, step-by-step argument that makes explicit: (i) the softmax/logit-shift invariance, (ii) the covariance interpretation of the logits Hessian, and (iii) how rank transfers through the linear map induced by W.

Step 1: Single-sample logits Hessian is a covariance. For a fixed input $\mathbf{x}$, let $\mathbf{a} = W\mathbf{z}$ denote the logits and $\mathbf{p} = \text{softmax}(\mathbf{a}) \in (0,1)^c$ with $\sum_i p_i = 1$. The Hessian of the cross-entropy loss w.r.t. logits is

$$\mathbf{H}_a(\mathbf{x}) = \nabla_{\mathbf{a}}^2 \mathcal{L}_{\text{ce}} = \text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top. \tag{23}$$

Consider the random one-hot vector $\mathbf{Y} \in \{\mathbf{e}_1, \ldots, \mathbf{e}_c\}$ drawn from $\mathbb{P}(\mathbf{Y} = \mathbf{e}_i) = p_i$. Then

$$\text{Cov}(\mathbf{Y}) = \mathbb{E}[\mathbf{Y}\mathbf{Y}^\top] - \mathbb{E}[\mathbf{Y}]\,\mathbb{E}[\mathbf{Y}]^\top = \text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top = \mathbf{H}_a(\mathbf{x}). \tag{24}$$

Hence for any $\mathbf{u} \in \mathbb{R}^c$,

$$\mathbf{u}^\top \mathbf{H}_a(\mathbf{x})\,\mathbf{u} = \text{Var}(\mathbf{u}^\top \mathbf{Y}) = \sum_{i=1}^{c} p_i u_i^2 - \Big(\sum_{i=1}^{c} p_i u_i\Big)^2 \geq 0, \tag{25}$$

with equality iff $\mathbf{u}^\top \mathbf{Y}$ is almost surely constant under strictly positive $\mathbf{p}$, which occurs exactly when $u_1 = \cdots = u_c$. Therefore, $\mathbf{H}_a(\mathbf{x})$ is positive semidefinite, its nullspace is $\mathrm{span}(\mathbf{1})$, and it is strictly positive definite on the subspace $\mathbf{1}^\perp = \{\mathbf{u} : \mathbf{1}^\top \mathbf{u} = 0\}$.

Softmax/logit-shift invariance. Adding any constant α to all logits leaves softmax unchanged: $\mathrm{softmax}(\mathbf{a} + \alpha\mathbf{1}) = \mathrm{softmax}(\mathbf{a})$. Directions along $\mathrm{span}(\mathbf{1})$ therefore produce no change in probabilities and incur zero curvature, matching the nullspace characterization above.

Step 2: Expectation preserves structure and fixes rank. Define the expected logits Hessian

$$\bar{\mathbf{H}}_a := \mathbb{E}_{\mathbf{x}}\left[\mathbf{H}_a(\mathbf{x})\right] = \mathbb{E}_{\mathbf{x}}\left[\mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top\right]. \tag{26}$$

As a convex combination of PSD matrices, $\bar{\mathbf{H}}_a$ is PSD. The nullspace and positivity on $\mathbf{1}^\perp$ are preserved: for any $\mathbf{u}$,

$$\mathbf{u}^\top \bar{\mathbf{H}}_a\, \mathbf{u} = \mathbb{E}_{\mathbf{x}}\left[\mathbf{u}^\top \mathbf{H}_a(\mathbf{x})\, \mathbf{u}\right] = \mathbb{E}_{\mathbf{x}}\left[\mathrm{Var}_{i\sim\mathbf{p}(\mathbf{x})}(u_i)\right]. \tag{27}$$

This equals 0 iff $u_1 = \cdots = u_c$, hence $\mathrm{Null}(\bar{\mathbf{H}}_a) = \mathrm{span}(\mathbf{1})$ and $\bar{\mathbf{H}}_a \succ 0$ on $\mathbf{1}^\perp$. Consequently, $\mathrm{rank}(\bar{\mathbf{H}}_a) = c - 1$.

Step 3: Transfer to feature space via W. The expected feature-space Hessian is

$$\mathbf{H}_f = \mathbb{E}_{\mathbf{x}}\left[\nabla^2_{\mathbf{z}} \mathcal{L}_{\text{ce}}\right]_{\text{to features}} = W^\top \bar{\mathbf{H}}_a\, W. \tag{28}$$

For any $\mathbf{v} \in \mathbb{R}^d$,

$$\mathbf{v}^\top \mathbf{H}_f\, \mathbf{v} = (W\mathbf{v})^\top \bar{\mathbf{H}}_a (W\mathbf{v}) \geq 0. \tag{29}$$

Moreover,

$$\mathbf{v}^\top \mathbf{H}_f\, \mathbf{v} = 0 \quad \Longleftrightarrow \quad W\mathbf{v} \in \mathrm{Null}(\bar{\mathbf{H}}_a) = \mathrm{span}(\mathbf{1}). \tag{30}$$

Intuitively, $\mathbf{v}$ is a feature direction that only adds an equal shift to all logits; cross-entropy (via softmax) is blind to such shifts, so curvature is zero along these directions.

Upper bound on rank. Using the property $\mathrm{rank}(\mathbf{A}^\top \mathbf{A}) \leq \mathrm{rank}(\mathbf{A})$ for any matrix $\mathbf{A}$, we have:

$$\mathrm{rank}(\mathbf{H}_f) = \mathrm{rank}(W^\top \bar{\mathbf{H}}_a W) \leq \min(\mathrm{rank}(W^\top), \mathrm{rank}(\bar{\mathbf{H}}_a)) = \min(\mathrm{rank}(W), c-1). \tag{31}$$

Tightness when W has full row rank. When $W \in \mathbb{R}^{c\times d}$ has full row rank (which implies $d \geq c$ and $\mathrm{rank}(W) = c$), its image is $\mathrm{Im}(W) = \mathbb{R}^c$, and $\mathrm{Im}(W^\top)$ is a c-dimensional subspace of $\mathbb{R}^d$. The kernel of $\bar{\mathbf{H}}_a^{1/2}$ is $\mathrm{span}(\mathbf{1})$, which is 1-dimensional. Since $\mathrm{Im}(W) = \mathbb{R}^c$ contains $\mathrm{span}(\mathbf{1})$, by the rank-nullity theorem for matrix products:

$$\mathrm{rank}(\bar{\mathbf{H}}_a^{1/2} W) = \mathrm{rank}(W) - \dim(\mathrm{Im}(W) \cap \mathrm{Ker}(\bar{\mathbf{H}}_a^{1/2})) = c - 1. \tag{32}$$

Therefore, when W has full row rank, $\mathrm{rank}(\mathbf{H}_f) = c - 1$.

Conclusion. For general W, the rank of $\mathbf{H}_f$ is upper bounded by $\min(\mathrm{rank}(W), c-1)$, and this bound is tight when W has full row rank. The intrinsically flat direction corresponds to uniform logit shifts (softmax invariance), eliminating one degree of curvature from the logit subspace. □

D.2 Proof of Proposition 2

Proof. We detail why the non-zero spectrum of $\mathbf{H}_f = W^\top \mathbf{A} W$ is identical to that of the reduced matrix $\mathbf{H}_{\text{red}} = \mathbf{R}\mathbf{A}\mathbf{R}^\top$, and how this gives both correctness and efficiency.

Step 1: Subspace structure via QR. For $c \leq d$, take the thin QR of $W^\top$:

$$W^\top = \mathbf{Q}\mathbf{R}, \quad \mathbf{Q} \in \mathbb{R}^{d\times c},\ \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_c,\ \mathbf{R} \in \mathbb{R}^{c\times c} \text{ upper triangular.} \tag{33}$$

The columns of $\mathbf{Q}$ are orthonormal and their span contains $\mathrm{Im}(W^\top)$ (with equality when W has full row rank). Using this factorization,

$$\mathbf{H}_f = W^\top \mathbf{A} W = (\mathbf{Q}\mathbf{R})\, \mathbf{A}\, (\mathbf{Q}\mathbf{R})^\top = \mathbf{Q}\, \underbrace{\left(\mathbf{R}\mathbf{A}\mathbf{R}^\top\right)}_{\mathbf{H}_{\text{red}}}\, \mathbf{Q}^\top. \tag{34}$$

Hence $\mathbf{H}_f$ acts trivially (as zero) on the orthogonal complement $\mathrm{Im}(\mathbf{Q})^\perp$, and maps $\mathrm{Im}(\mathbf{Q})$ into itself via the c × c operator $\mathbf{H}_{\text{red}}$ (expressed in the $\mathbf{Q}$-basis).

Step 2: Eigenpair correspondence (both directions). Any $\mathbf{v} \in \mathbb{R}^d$ decomposes uniquely as $\mathbf{v} = \mathbf{Q}\mathbf{u} + \mathbf{v}_\perp$ with $\mathbf{v}_\perp \perp \mathrm{Im}(\mathbf{Q})$. Since $\mathbf{Q}^\top \mathbf{v}_\perp = \mathbf{0}$, we have $\mathbf{H}_f \mathbf{v} = \mathbf{Q}\,\mathbf{H}_{\text{red}}\,\mathbf{u}$. Therefore, any eigenvector with non-zero eigenvalue must lie in $\mathrm{Im}(\mathbf{Q})$: if $\mathbf{H}_f \mathbf{v} = \sigma \mathbf{v}$ and $\sigma \neq 0$, then $\mathbf{v}_\perp = \mathbf{0}$ and $\mathbf{v} = \mathbf{Q}\mathbf{u}$.

Substituting $\mathbf{v} = \mathbf{Q}\mathbf{u}$ into $\mathbf{H}_f \mathbf{v} = \sigma \mathbf{v}$ gives

$$\mathbf{Q}\,\mathbf{H}_{\text{red}}\,\mathbf{u} = \sigma\,\mathbf{Q}\mathbf{u} \quad \Longleftrightarrow \quad \mathbf{H}_{\text{red}}\,\mathbf{u} = \sigma\,\mathbf{u}, \tag{35}$$

because $\mathbf{Q}$ has full column rank and $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$. Thus, non-zero eigenpairs $(\sigma, \mathbf{v})$ of $\mathbf{H}_f$ correspond bijectively to eigenpairs $(\sigma, \mathbf{u})$ of $\mathbf{H}_{\text{red}}$ via $\mathbf{v} = \mathbf{Q}\mathbf{u}$; conversely, any eigenpair of $\mathbf{H}_{\text{red}}$ lifts to one of $\mathbf{H}_f$.

Step 3: Rayleigh quotient equality (spectral identity). For $\mathbf{v} = \mathbf{Q}\mathbf{u}$,

$$\frac{\mathbf{v}^\top \mathbf{H}_f \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} = \frac{\mathbf{u}^\top \mathbf{H}_{\text{red}} \mathbf{u}}{\mathbf{u}^\top \mathbf{u}}, \tag{36}$$

since $\mathbf{v}^\top \mathbf{H}_f \mathbf{v} = \mathbf{u}^\top \mathbf{H}_{\text{red}} \mathbf{u}$ and $\mathbf{v}^\top \mathbf{v} = \mathbf{u}^\top \mathbf{u}$ by $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$. This shows $\mathbf{H}_f$ and $\mathbf{H}_{\text{red}}$ have identical non-zero eigenvalues (same extremal Rayleigh quotients) and identical inertia on $\mathrm{Im}(\mathbf{Q})$.

Step 4: Complexity implication. Computing $\mathbf{Q}, \mathbf{R}$ costs $O(dc^2)$. Forming $\mathbf{H}_{\text{red}} = \mathbf{R}\mathbf{A}\mathbf{R}^\top$ costs $O(c^3)$ (two c $\times$ c multiplications). Eigendecomposition of $\mathbf{H}_{\text{red}}$ also costs $O(c^3)$. Thus overall $O(dc^2 + c^3)$, versus forming $\mathbf{H}_f$ explicitly ($O(d^2 c)$) and decomposing it ($O(d^3)$) when $c \ll d$.

Therefore, the non-zero eigenvalues (and corresponding eigenvectors) are obtained equivalently and far more efficiently via the c $\times$ c reduced problem. □

E Further Analysis

E.1 Alternative Subspace Identification

We compare the Hessian-guided stable–plastic decomposition with an alternative based on Principal Component Analysis (PCA) of within-class covariance matrices.

PCA-based Subspace Identification. Given the feature space $\mathbb{R}^d$ and c_{t-1} learned classes, let $\Sigma_i \in \mathbb{R}^{d \times d}$ denote the covariance matrix of class i with n_i samples. The weighted average covariance matrix is:

$$\bar{\Sigma} = \sum_{i=1}^{c_{t-1}} \frac{n_i}{N} \Sigma_i, \quad N = \sum_{i=1}^{c_{t-1}} n_i. \tag{37}$$

Performing eigendecomposition $\bar{\Sigma} = U \Lambda U^T$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$, the stable subspace is defined by the top $K = c_{t-1} - 1$ principal components:

$$\mathcal{S}_{\text{PCA}} = \text{span}\{u_1, u_2, \ldots, u_K\}, \tag{38}$$

with the remaining $d - K$ dimensions forming the plastic subspace $\mathcal{P}_{\text{PCA}}$. This approach uses second-order feature statistics (covariance) with computational complexity $O(d^2 c + d^3)$.

Comparative Evaluation. Table 4 compares the two identification strategies on CIFAR-100 and Tiny-ImageNet. For a fair comparison, both methods use only SGR for feature regularization, excluding classifier calibration. Results show that the Hessian-based method consistently outperforms the PCA-based approach across all settings, with the advantage being particularly pronounced in the 20-task configurations. This validates that loss curvature analysis more precisely captures feature directions critical for preserving past knowledge.

E.2 SGR and SGPA Coupling

In our work, we address feature drift and prototype drift within a unified framework. SGR constrains feature drift by penalizing changes in the stable subspace, while SGPA leverages the same subspace information to guide prototype alignment. This coupling is not only theoretically coherent but also computationally efficient, as SGPA reuses the stable subspace computed for SGR without introducing extra overhead.

Table 4: Performance comparison of different subspace identification methods (SGR only, without classifier calibration).

Method	CIFAR-100				Tiny-ImageNet			
	10 Tasks		20 Tasks		10 Tasks		20 Tasks	
	Acc	AAA	Acc	AAA	Acc	AAA	Acc	AAA
PCA-based	36.50	48.68	20.50	32.21	31.59	41.84	23.22	36.12
Hessian-based	**47.75**	**54.32**	**35.47**	**46.55**	**34.21**	**45.18**	**27.35**	**39.86**

To demonstrate the effectiveness of this integrated design, we compare our full model against hybrid versions where SGR is paired with other general-purpose prototype drift compensation modules: SDC (Semantic Drift Compensation)[Yu et al., 2020], LDC (Learnable Drift Compensation)[Gomez-Villa et al., 2024], and ADC (Adversarial Drift Compensation)[Goswami et al., 2024]. The results on CIFAR-100 are presented in Table 5.

Table 5: Performance comparison on CIFAR-100 (10/20 tasks) with SGR paired with different drift-compensation modules. Our coupled SGPA approach yields the best overall performance.

Method	10 Tasks		20 Tasks	
	Acc	AAA	Acc	AAA
SGR + ADC	43.00	59.05	32.76	45.08
SGR + SDC	48.98	62.12	36.55	48.56
SGR + LDC	49.10	62.21	35.04	48.28
SGR + SGPA (Ours)	**49.68**	**62.88**	**37.23**	**49.80**

As shown in the table, while SGR provides a strong foundation that improves performance with all drift compensation modules, the tightly coupled SGR+SGPA configuration achieves the best results. This highlights the benefit of our unified, co-designed approach to managing both feature and prototype drift.

Controlled comparison with EFC. Table 6 crosses the EFC and SGR regularizers with the two calibration strategies on CIFAR-100/10T, using the same seed and protocol. At fixed EFC calibration, SGR improves Acc/AAA by 5.23/3.95 points; at fixed SGPA, it improves them by 2.40/2.46 points. SGPA also improves Acc under either regularizer. This isolates a benefit from the regularization mechanism beyond the choice of calibration, complementing the Hessian-versus-PCA comparison and the perturbation analysis in Appendix F.3.

Table 6: Regularizer/calibration cross-ablation on CIFAR-100 (10 tasks).

Regularizer	Calibration	Acc	AAA
EFC	EFC	43.95	58.58
EFC	SGPA	47.28	60.42
SGR	EFC	49.18	62.53
SGR	SGPA	**49.68**	**62.88**

Source of prototype-alignment gains. Keeping SGR and all other settings fixed, Table 7 compares six prototype-update rules. The random subspace is orthonormal and has the same rank as the Hessian support; global drift sets $S_i = 0$. The class-wise oracle uses old data only to measure true class-wise drift and is a diagnostic reference. SGPA gains 1.58 points over no update and 1.18 over global drift. Curvature weighting changes Acc by only 0.04 points, supporting the unweighted score in Equation (11).

Stronger semantic shift. On a CIFAR-100 split with two semantically related superclasses per task, initial prototype drift increases from 0.63 under the random split to 1.09. Enabling SGPA improves Acc/AAA from 45.80/59.28 to 46.83/60.44 and reduces forgetting from 21.46 to 20.40.

Table 7: Prototype-update controls with fixed SGR on CIFAR-100 (10 tasks).

Prototype update	Acc
No update	48.10
Global drift ($S_i = 0$)	48.50
Random subspace (matched rank)	47.80
Hessian projection, unweighted (SGPA)	49.68
Hessian projection, curvature-weighted	49.72
Class-wise oracle drift	50.90

SGPA uses a shared drift estimate with prototype-specific modulation; class-conditioned or nonlinear drift estimation remains a direction for broader distribution shifts.

E.3 Choice of λ_s and λ_p

Geometric Interpretation. The coefficients λ_s and λ_p govern stability and plasticity. Their effective scales also depend on the Hessian spectrum and the subspace dimensions, so raw coefficient values need not be comparable across datasets.

Role of the hyperparameters. To prioritize knowledge retention while enabling adaptation, we consistently enforce $\lambda_s \gg \lambda_p$, where λ_s controls the stable subspace and λ_p controls the plastic subspace.

Parameter sensitivity. In the evaluated low-rank settings, we use a wider candidate range for λ_s and a finer range for λ_p, whose penalty acts on the larger plastic subspace. Table 8 provides a post-hoc sensitivity analysis, separate from the validation search below.

Table 8: Sensitivity to λ_s on Tiny-ImageNet ($\lambda_p = 0.03$) and ImageNet-Subset ($\lambda_p = 0.1$).

λ_s	Tiny-ImageNet		ImageNet-Subset	
	10-Task	20-Task	10-Task	20-Task
10	36.78	30.92	46.58	33.21
11	36.10	30.03	47.14	34.65
12	35.41	29.24	48.08	35.11

Dataset-dependent ranges. Dataset difficulty serves as a qualitative prior for choosing candidate ranges; validation performance determines the selected pair. Table 9 illustrates the effects of changing coefficients relative to the CIFAR-100 setting. These post-hoc comparisons are distinct from the search grid and do not define an automatic difficulty-based rule.

Table 9: Post-hoc coefficient comparisons relative to the CIFAR-100 setting ($\lambda_s = 5, \lambda_p = 0.03$).

Dataset	(λ_s, λ_p)	**Acc**	**Notes**
Tiny-ImageNet (10 tasks)	$(5, 0.03)$	34.91	CIFAR-100 baseline
	$(10, 0.03)$	**36.78**	Increase λ_s
	$(5, 0.04)$	36.15	Increase λ_p a little
	$(5, 0.05)$	34.80	Increase more λ_p
ImageNet-Subset (10 tasks)	$(5, 0.03)$	47.22	CIFAR-100 baseline
	$(10, 0.05)$	51.63	Moderate increase
	$(20, 0.1)$	**53.52**	Optimal setting
	$(25, 0.15)$	49.87	Over-regularization

Hyperparameter Search Strategy. Following standard continual learning practices [Douillard et al., 2020], we randomly split each class's training data into 90% for training and 10% for validation. All hyperparameters were selected based on validation set performance, and final test results were obtained by retraining on the full training data.

We evaluated six pairs on CIFAR-100, using $\lambda_s \in \{1, 5, 10\}$ and $\lambda_p \in \{0.02, 0.03\}$; four on Tiny-ImageNet, using $\lambda_s \in \{8, 10\}$ and $\lambda_p \in \{0.03, 0.05\}$; and four on ImageNet-Subset, using $\lambda_s \in \{10, 20\}$ and $\lambda_p \in \{0.05, 0.1\}$. The selected pairs are $(5, 0.03)$, $(10, 0.03)$, and $(20, 0.1)$, respectively. Selection used only the 10-task splits, with unchanged transfer to 20 tasks. The ImageNet-Subset pair also transfers to CUB-200 and ImageNet-1K. Baselines follow their official hyperparameter and optimizer recipes rather than being retuned on this grid.

Automatic Selection. Exploratory rules based on Hessian trace/mean-eigenvalue normalization, stable/plastic dimension normalization, and learnable coefficients did not transfer reliably across datasets and underperformed validation-selected values. We therefore retain the small validation search, with reliable automatic selection remaining an open direction.

Optimizer Robustness. On ImageNet-Subset/20T, replacing Adam in subsequent tasks with SGD (backbone learning rate 10^{-4}, head learning rate 2×10^{-4}, momentum 0.9) changes final Acc from 41.44 to 40.53. The gain over DPCR remains under this alternative optimizer.

E.4 Performance on other benchmarks

Furthermore, we evaluate SGCL on CUB-200[Wah et al., 2011] and ImageNet-1K[Deng et al., 2009] (Table 10). ImageNet-1K uses the ImageNet-Subset settings and ten tasks of 100 classes. With ResNet-18 ($d = 512$), we switch to the exact $d \times d$ Hessian decomposition when $c > d$. For CUB-200, we reduce the backbone learning rate (7×10^{-4} initially and 6×10^{-5} subsequently), while retaining $(\lambda_s, \lambda_p) = (20, 0.1)$.

Full-rank behavior on ImageNet-1K. At boundaries after tasks 6–10, the Hessian is numerically full rank: $r_t = 512$ and $d - r_t = 0$. Thus, tasks 7–10 use $\mathcal{S} = \mathbb{R}^{512}$ and $\mathcal{P} = \{0\}$; the plastic penalty and SGPA drift correction are inactive. The stable penalty remains curvature-weighted over the full feature space, permitting more adaptation along lower-curvature directions through its finite soft penalty. Current-task cross-entropy and classifier calibration with unchanged old prototypes and covariances continue to operate. The later-task results therefore characterize this full-rank limiting case of SGCL.

Table 10: Performance comparison on CUB-200 and ImageNet-1K benchmarks. We report Average Accuracy (Acc) and Average Anytime Accuracy (AAA).

Method	**CUB-200**				**ImageNet-1K**	
	10 Tasks		20 Tasks		10 Tasks	
	Acc	AAA	Acc	AAA	Acc	AAA
EFC	51.03	63.28	46.13	59.37	42.35	56.14
ADC	44.65	59.62	19.47	39.72	31.34	50.95
LDC	40.16	54.73	24.49	42.67	35.15	53.88
SGCL (Ours)	**52.10**	**64.02**	**46.78**	**60.10**	**44.82**	**58.49**

E.5 Performance on ViT

To verify the generality of our approach beyond convolutional architectures, we conduct experiments on a pre-trained (ImageNet-21K) ViT-B/16 [Dosovitskiy et al., 2021] backbone, evaluated on ImageNet-A [Hendrycks et al., 2021b] and ImageNet-R [Hendrycks et al., 2021a] with 10 tasks. We compare *LoRA* [Hu et al., 2022] and *Fine-Tuning* (FT), each with and without SGR. As shown in Table 11, SGR consistently improves both adaptation strategies across both benchmarks.

These results show that SGR benefits both parameter-efficient adaptation and full fine-tuning. To isolate its contribution within SD-LoRA, we additionally compare SD-LoRA with and without the stable SGR term under the same pretrained protocol over three seeds (Table 12). SGR improves mean Acc by 0.94/0.91 points on ImageNet-A/R. The table also includes representative prompt methods under the pretrained ViT-B/16, 10-task protocol, separately from the cold-start ResNet comparisons.

Table 11: Average Accuracy on ImageNet-A (A) and ImageNet-R (R) with ViT-B/16 backbone (10 tasks). SGR consistently improves both LoRA and Fine-Tuning.

	A-LoRA	A-FT	R-LoRA	R-FT
w/o SGR	52.93	17.03	74.24	60.57
w/ SGR	**56.14**	**25.34**	**78.40**	**71.12**
SD-LoRA [Wu et al., 2025]	55.96	—	77.34	—

Table 12: Pretrained ViT-B/16 comparison (10 tasks), final Acc (%). SD-LoRA rows report mean $\pm$ standard deviation over three seeds; prompt rows are point estimates.

Method	ImageNet-A	ImageNet-R
L2P [Wang et al., 2022b]	42.94	71.26
DualPrompt [Wang et al., 2022a]	45.49	68.22
CODA-Prompt [Smith et al., 2023]	45.36	74.05
SD-LoRA [Wu et al., 2025]	55.96 ± 0.98	77.30 ± 0.35
SD-LoRA+SGR	$\mathbf{56.90} \pm 0.90$	$\mathbf{78.21} \pm 0.30$

E.6 Multi-seed Reliability and Forgetting

Table 13 reports Acc/AAA for SGCL and DPCR over three seeds. On CIFAR-100, their differences lie within seed-level variation. On Tiny-ImageNet and ImageNet-Subset, SGCL improves mean Acc by 1.77–5.72 points and AAA by 1.90–6.39 points. These statistics complement the point estimates in Table 1.

Table 13: Three-seed results: mean $\pm$ standard deviation (%).

Setting	DPCR Acc	DPCR AAA	SGCL Acc	SGCL AAA
CIFAR-100/10T	49.26 ± 0.40	62.90 ± 0.55	49.53 ± 0.30	62.45 ± 0.90
CIFAR-100/20T	37.70 ± 1.50	49.50 ± 1.55	37.53 ± 1.66	49.93 ± 1.71
Tiny-ImageNet/10T	34.85 ± 0.55	47.35 ± 0.50	36.62 ± 0.43	49.25 ± 0.44
Tiny-ImageNet/20T	26.60 ± 1.15	38.40 ± 0.90	30.47 ± 1.20	44.79 ± 0.51
ImageNet-Subset/10T	49.40 ± 1.35	66.70 ± 1.40	53.30 ± 1.32	69.32 ± 1.44
ImageNet-Subset/20T	35.50 ± 1.60	50.75 ± 1.70	41.22 ± 1.57	55.35 ± 1.80

We also report final forgetting and backward transfer (BWT) in Table 14. For task-wise accuracy $a_{i,j}$ on task j after learning task i, forgetting averages $\max_{j \le i < T} a_{i,j} - a_{T,j}$ over $j < T$, while BWT averages $a_{T,j} - a_{j,j}$ over the same tasks. Lower forgetting and higher BWT indicate better retention.

E.7 Persistent State and Memory Accounting

After task t, SGCL retains one prototype and covariance per seen class and the latest eigensystem (U_t, σ_t). Old prototypes are aligned, old covariances retained, and new-class statistics appended. Each eigensystem replaces the previous one; Hessian and QR intermediates are discarded. With C seen classes, feature dimension d, and rank $r \le \min(d, C-1)$, the persistent statistics occupy

$$M_{\text{stats}} = 4\big(Cd + Cd^2 + dr + r\big) \text{ bytes}$$

in FP32. Table 15 uses the same full-covariance accounting for SGCL, EFC, and DPCR. Covariances dominate; SGCL's basis and weights contribute only 0.19/0.39 MiB for 100/200 classes.

The table uses $r = C - 1$ for SGCL in these settings; DPCR's drift predictor is discarded. Alongside the current trainable model, SGCL keeps one frozen previous backbone, replaced after each task. Table 3 measures regularization cost; Table 15 measures stored statistics, not peak training memory including activations and optimizer state. Storage grows with the number of classes through Cd^2, rather than through a sequence of task-specific subspaces. For a fixed class count, final storage is unchanged between 10T and 20T. Exemplar-free learning excludes stored raw examples while retaining these aggregate statistics.

Table 14: Final forgetting and BWT (percentage points).

Setting	DPCR forgetting	DPCR BWT	SGCL forgetting	SGCL BWT
CIFAR-100/10T	13.89	-13.52	13.40	-13.40
CIFAR-100/20T	26.63	-19.17	19.85	-18.63
Tiny-ImageNet/10T	32.98	-32.98	32.12	-32.12
Tiny-ImageNet/20T	34.89	-34.89	27.81	-27.03
ImageNet-Subset/10T	11.84	-10.73	9.42	-8.12
ImageNet-Subset/20T	22.95	-21.81	20.13	-19.65

Table 15: Persistent statistics with FP32 and $d = 512$ (MiB, $1\,\mathrm{MiB} = 2^{20}$ bytes). CIFAR-100 and ImageNet-Subset have 100 classes; Tiny-ImageNet has 200. Model parameters are accounted for separately below.

Method	Prototypes	Covariances	Other state	100 classes	200 classes
SGCL	Cd	Cd^2	$dr + r$	100.39	200.78
EFC	Cd	Cd^2	EFM: d^2	101.20	201.39
DPCR	Cd	Cd^2	None	100.20	200.39

E.8 Additional Cold-start Comparisons

We evaluate FCS [Li et al., 2024b] and Adam-NSCL [Wang et al., 2021] under the same cold-start, single-head protocol as SGCL (Table 16). FCS's original configuration uses a larger initial task, while Adam-NSCL's original task/multi-head and class/single-head protocols differ from this setting. The aligned results therefore characterize their performance under the protocol studied here.

Table 16: Additional cold-start, single-head comparisons: final Acc (%). Entries marked < 10 denote accuracy below 10%.

Method	CIFAR-100/10T	CIFAR-100/20T	Tiny-ImageNet/10T	Tiny-ImageNet/20T
FCS	37.70	< 10	22.09	< 10
Adam-NSCL	33.42	26.68	19.75	< 10

E.9 Asset Licenses and Terms of Use

All datasets, pretrained backbones, and baseline implementations used in this study are publicly available for academic research and were used in accordance with their respective licenses and terms of access:

- **Datasets:** CIFAR-100 [Krizhevsky et al., 2009] is publicly distributed by the authors for academic research and educational use (no formal SPDX license specified). ImageNet-1K and ImageNet-Subset [Deng et al., 2009, Douillard et al., 2020], as well as Tiny-ImageNet [Wu et al., 2017], are subject to the ImageNet custom non-commercial research and educational terms of access (`https://image-net.org/download`). CUB-200-2011 [Wah et al., 2011] is distributed by Caltech/UCSD for non-commercial research and educational purposes. ImageNet-A [Hendrycks et al., 2021b] and ImageNet-R [Hendrycks et al., 2021a] are released under the MIT License.
- **Models and Code:** ResNet-18 [He et al., 2016] (via PyTorch/`torchvision`, BSD-3-Clause License), ImageNet-21K pretrained ViT-B/16 [Dosovitskiy et al., 2021] (Apache-2.0 License), LoRA [Hu et al., 2022] (MIT License), and open-source continual learning baseline implementations (e.g., EFC [Magistri et al., 2024] and SD-LoRA [Wu et al., 2025], MIT License) are used under their respective open-source licenses.

F Theoretical Analysis of the Hessian-Guided Decomposition

In this appendix, we provide a regularization-free theoretical justification for the Hessian-guided stable–plastic decomposition used in SGCL, with the task-boundary old classifier held fixed.

Under the local quadratic model of old-task loss, the Courant–Fischer min-max theorem shows that among *all* m-dimensional subspaces—with no eigen-alignment restriction—the highest-curvature directions are the optimal choices for the stable subspace (Theorem F.1). Moreover, the SGCL choice $\mathcal{S}_t = \mathrm{Im}(\mathbf{H}_f^{(t-1)})$ is the unique decomposition that maximizes the plasticity budget while guaranteeing zero quadratic forgetting.

F.1 Local quadratic model of forgetting

Recall that the model consists of a feature extractor $f_\theta : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^d$ and a linear classifier $W_t \in \mathbb{R}^{c_t \times d}$, where $c_t = \sum_{k=1}^{t} |\mathcal{C}_k|$ is the total number of classes observed up to task t. For an input $\mathbf{x}$ we denote its feature by $\mathbf{z} = f_\theta(\mathbf{x}) \in \mathbb{R}^d$.

At the end of task $t-1$, let $\mathcal{L}_{\le t-1}$ denote the old-task loss, written as an average over the past task distributions $\{\mathcal{D}_k\}_{k=1}^{t-1}$ as in Equation (1) of the main paper. Treating W_{t-1} as fixed and viewing $\mathcal{L}_{\le t-1}$ as a function of the feature representation $\mathbf{z}$, we denote the corresponding feature-space Hessian at task $t-1$ by

$$\mathbf{H}_f^{(t-1)} \;=\; \nabla_{\mathbf{z}}^2 \mathcal{L}_{\le t-1}(\mathbf{z})\Big|_{\mathbf{z}=\mathbf{z}_{t-1}}, \tag{39}$$

where $\mathbf{z}_{t-1} = f_{\theta_{t-1}}(\mathbf{x})$ is the feature before learning task t. As derived in the main paper (Equation (4)), for the cross-entropy loss with a softmax classifier $\mathbf{H}_f^{(t-1)}$ is symmetric positive semidefinite.

For a small feature drift $\Delta\mathbf{z}_t \in \mathbb{R}^d$ induced by training task t, the second-order Taylor expansion of the old-task loss gives

$$\begin{aligned} \Delta\mathcal{L}_{\le t-1}^{(t)}(\Delta\mathbf{z}_t) &:= \mathcal{L}_{\le t-1}(\mathbf{z}_{t-1} + \Delta\mathbf{z}_t) - \mathcal{L}_{\le t-1}(\mathbf{z}_{t-1}) \\ &= \mathbf{g}_{\le t-1}^\top \Delta\mathbf{z}_t + \frac{1}{2}\,\Delta\mathbf{z}_t^\top \mathbf{H}_f^{(t-1)} \Delta\mathbf{z}_t + o(\|\Delta\mathbf{z}_t\|_2^2), \end{aligned} \tag{40}$$

where $\mathbf{g}_{\le t-1} := \nabla_{\mathbf{z}} \mathcal{L}_{\le t-1}(\mathbf{z})\big|_{\mathbf{z}=\mathbf{z}_{t-1}}$. We focus on the curvature-induced component and quantify forgetting using the quadratic surrogate

$$\widetilde{\Delta\mathcal{L}}_{\le t-1}^{(t)}(\Delta\mathbf{z}_t) \;:=\; \frac{1}{2}\,\Delta\mathbf{z}_t^\top \mathbf{H}_f^{(t-1)} \Delta\mathbf{z}_t. \tag{41}$$

When $\mathbf{g}_{\le t-1} = 0$, this is the leading Taylor term; higher-order terms remain. The following theorem characterizes this surrogate. Exact softmax invariance in the Hessian kernel is established separately in Appendix F.4.

F.2 Regularization-free optimality via Courant–Fischer

Before analyzing any particular regularization scheme, we establish that the optimality of high-curvature stable directions is a fundamental property of the loss landscape itself. Consider any orthogonal decomposition $\mathbb{R}^d = \mathcal{S}_t \oplus \mathcal{P}_t$ with $\dim(\mathcal{P}_t) = m \ge 1$, where $\mathcal{P}_t$ is the "plastic" subspace in which feature drift is allowed. The worst-case quadratic forgetting over all bounded drifts confined to $\mathcal{P}_t$ is

$$\Phi_{\text{wc}}(\mathcal{P}_t) \;:=\; \max_{\Delta\mathbf{z}_t \in \mathcal{P}_t,\ \|\Delta\mathbf{z}_t\| \le \epsilon} \widetilde{\Delta\mathcal{L}}_{\le t-1}^{(t)}(\Delta\mathbf{z}_t) \;=\; \frac{\epsilon^2}{2} \max_{\mathbf{v} \in \mathcal{P}_t,\ \|\mathbf{v}\|=1} \mathbf{v}^\top \mathbf{H}_f^{(t-1)} \mathbf{v}. \tag{42}$$

Theorem F.1 (Optimality of high-curvature stable directions)**.** *Let $\mathbf{H}_f^{(t-1)} \succeq 0$ have eigenvalues $\sigma_1^{(t-1)} \ge \cdots \ge \sigma_d^{(t-1)} \ge 0$ and rank $r = \mathrm{rank}(\mathbf{H}_f^{(t-1)})$, and let $\epsilon > 0$. For any plasticity budget $m \in \{1, \ldots, d\}$, the minimum worst-case quadratic forgetting over* all *m-dimensional subspaces $\mathcal{P}_t \subseteq \mathbb{R}^d$ (with no eigen-alignment restriction) satisfies*

$$\min_{\dim(\mathcal{P}_t)=m} \Phi_{\text{wc}}(\mathcal{P}_t) \;=\; \frac{\epsilon^2}{2}\,\sigma_{d-m+1}^{(t-1)}. \tag{43}$$

The minimum is attained by $\mathcal{P}_t^\star = \mathrm{span}\{\mathbf{u}_{d-m+1}^{(t-1)}, \ldots, \mathbf{u}_d^{(t-1)}\}$ *(the* m *smallest-eigenvalue directions), and the corresponding optimal stable subspace is* $\mathcal{S}_t^\star = \mathrm{span}\{\mathbf{u}_1^{(t-1)}, \ldots, \mathbf{u}_{d-m}^{(t-1)}\}$ *(the highest-curvature directions).*

In particular, when $r < d$*, for* $m = d - r$ *we obtain* $\Phi_{\mathrm{wc}}(\mathcal{P}_t^\star) = 0$ *with* $\mathcal{P}_t^\star = \mathrm{Ker}(\mathbf{H}_f^{(t-1)})$ *and* $\mathcal{S}_t^\star = \mathrm{Im}(\mathbf{H}_f^{(t-1)})$*, recovering the SGCL decomposition. This is the largest plastic subspace that achieves zero worst-case quadratic forgetting. When* $r = d$*, only the zero-dimensional subspace* $\mathcal{P}_t = \{0\}$ *achieves zero forgetting, with* $\Phi_{\mathrm{wc}}(\{0\}) = 0$ *by the bounded-drift definition.*

Proof. By the Courant–Fischer min-max theorem, the j-th eigenvalue of $\mathbf{H}_f^{(t-1)}$ satisfies

$$\sigma_j^{(t-1)} \;=\; \min_{\dim(V)=d-j+1} \; \max_{\mathbf{v}\in V,\ \|\mathbf{v}\|=1} \mathbf{v}^\top \mathbf{H}_f^{(t-1)} \mathbf{v}.$$

Setting $j = d - m + 1$ gives $d - j + 1 = m = \dim(\mathcal{P}_t)$, so

$$\min_{\dim(\mathcal{P}_t)=m} \; \max_{\mathbf{v}\in\mathcal{P}_t,\ \|\mathbf{v}\|=1} \mathbf{v}^\top \mathbf{H}_f^{(t-1)} \mathbf{v} \;=\; \sigma_{d-m+1}^{(t-1)},$$

and the minimum is attained by $\mathcal{P}_t^\star = \mathrm{span}\{\mathbf{u}_{d-m+1}^{(t-1)}, \ldots, \mathbf{u}_d^{(t-1)}\}$. Multiplying both sides by $\epsilon^2/2$ yields (43).

For the SGCL case with $r < d$ and $m = d - r$, we have $\sigma_{d-m+1}^{(t-1)} = \sigma_{r+1}^{(t-1)} = 0$ since $\mathrm{rank}(\mathbf{H}_f^{(t-1)}) = r$. The optimal plastic subspace is $\mathrm{span}\{\mathbf{u}_{r+1}^{(t-1)}, \ldots, \mathbf{u}_d^{(t-1)}\} = \mathrm{Ker}(\mathbf{H}_f^{(t-1)})$, and its stable complement is $\mathrm{Im}(\mathbf{H}_f^{(t-1)})$. To see that $m = d - r$ is the largest plastic budget with zero forgetting, note that for $m > d - r$ one has $d - m + 1 \le r$, hence $\sigma_{d-m+1}^{(t-1)} > 0$ and zero forgetting is unattainable. If $r = d$, positive definiteness gives a positive quadratic loss for every nonzero drift, so only $\mathcal{P}_t = \{0\}$ has zero worst-case forgetting. □

Interpretation. Theorem F.1 makes no reference to any regularization scheme. It states a property of the Hessian $\mathbf{H}_f^{(t-1)}$ alone: for any fixed plasticity budget m, the optimal directions to preserve (stable subspace) are always the highest-curvature ones, and the optimal directions for adaptation (plastic subspace) are the lowest-curvature ones. In particular, the optimization in (43) is over the full Grassmannian $\mathrm{Gr}(m, d)$, not merely over subsets of eigenvectors; even subspaces oblique to the Hessian eigenbasis are included, and the eigenvector-aligned solution is still provably optimal.

Summary. Under the quadratic surrogate (41) of the old-task loss increase, the Hessian-guided decomposition used by SGCL,

$$\mathcal{S}_t \;=\; \mathrm{Im}(\mathbf{H}_f^{(t-1)}) \;=\; \mathrm{span}\{\mathbf{u}_i^{(t-1)} : \sigma_i^{(t-1)} > 0\}, \qquad \mathcal{P}_t \;=\; \mathrm{Ker}(\mathbf{H}_f^{(t-1)}), \tag{44}$$

has the largest plastic subspace with zero worst-case quadratic forgetting. This justifies the subspace choice independently of a particular regularizer. SGR implements soft penalties, while Algorithm 1 also updates and recalibrates the classifier; the theorem is therefore a geometric characterization of the fixed-head surrogate, rather than an end-to-end accuracy guarantee.

F.3 Empirical Verification of Subspace Functional Roles

The theoretical analysis links the stable subspace (nonzero-curvature directions) to old-loss sensitivity and the plastic subspace (null space) to invariance under the fixed old head. We examine these functional roles through two complementary experiments.

Sensitivity to stable-subspace regularization strength. In the main paper, Figure 5b shows that reducing the stable penalty weight λ_s increases average forgetting, linking protection of these directions to empirical retention.

Feature-perturbation experiment. To further validate the functional distinction, we conduct a controlled perturbation study on CIFAR-100 and Tiny-ImageNet Task 1 (averaged over 10 runs). Using the stable basis U_s computed from the training-set Hessian and the plastic basis U_p, we perturb test features $\mathbf{z}$ via

$$\mathbf{z}_{\text{stable}} = \mathbf{z} + \alpha\,\boldsymbol{\epsilon}_s, \qquad \mathbf{z}_{\text{plastic}} = \mathbf{z} + \alpha\,\boldsymbol{\epsilon}_p, \tag{45}$$

where $\boldsymbol{\epsilon}_s \in \operatorname{span}(U_s)$ and $\boldsymbol{\epsilon}_p \in \operatorname{span}(U_p)$ are unit-norm random perturbations scaled by α.

Table 17: Feature-perturbation experiment with the fixed old classifier. Plastic perturbations preserve accuracy, while stable perturbations affect it. Drop is the accuracy decrease in percentage points, reported as mean $\pm$ standard deviation over 10 perturbation runs.

Dataset	α	**Acc. (stable pert.)**	**Drop**	**Acc. (plastic pert.)**	**Drop**
CIFAR-100 (baseline 90.30)	0.1	90.29	0.01 ± 0.07	90.30	0.00 ± 0.00
	0.5	90.17	0.13 ± 0.16	90.30	0.00 ± 0.00
	1.0	89.96	0.34 ± 0.21	90.30	0.00 ± 0.00
	2.0	89.17	1.13 ± 0.65	90.30	0.00 ± 0.00
Tiny-ImageNet (baseline 80.10)	0.1	80.16	-0.06 ± 0.10	80.10	0.00 ± 0.00
	0.5	80.38	-0.28 ± 0.14	80.10	0.00 ± 0.00
	1.0	80.15	-0.05 ± 0.23	80.10	0.00 ± 0.00
	2.0	79.40	0.70 ± 0.59	80.10	0.00 ± 0.00

In Table 17, plastic perturbations produce zero accuracy change at every tested magnitude. Stable perturbations increasingly degrade CIFAR-100 accuracy; on Tiny-ImageNet, small perturbations can slightly improve accuracy, while $\alpha = 2$ causes a 0.70-point drop. These results support the distinction between decision-invariant and decision-sensitive directions.

F.4 Subspace Stability and Accumulated Drift

Exact support and invariance. For a fixed old softmax classifier W and strictly positive probabilities, the proof of Proposition 3.1 gives $\operatorname{Ker}(\mathbf{A}) = \operatorname{span}(\mathbf{1})$ and

$$\operatorname{Ker}(\mathbf{H}_f) = \{\mathbf{v} : W\mathbf{v} \in \operatorname{span}(\mathbf{1})\}, \qquad \operatorname{Im}(\mathbf{H}_f) = W^\top \mathbf{1}^\perp.$$

Indeed, $\mathbf{v}^\top \mathbf{H}_f \mathbf{v} = (W\mathbf{v})^\top \mathbf{A}(W\mathbf{v})$ vanishes exactly for a common logit shift. Taking orthogonal complements gives the image identity, equivalently $\operatorname{span}\{w_i - w_j\}$ for classifier rows w_i. Consequently, for every $\mathbf{v} \in \operatorname{Ker}(\mathbf{H}_f)$ and any magnitude,

$$\operatorname{softmax}\big(W(\mathbf{z}+\mathbf{v})\big) = \operatorname{softmax}(W\mathbf{z}), \qquad \mathcal{L}_{\text{ce}}\big(W(\mathbf{z}+\mathbf{v}), y\big) = \mathcal{L}_{\text{ce}}(W\mathbf{z}, y).$$

SGCL uses this full nonzero support. Changing probabilities can alter eigenvalues and axes within it, but not its span for fixed W; when W has full row rank, its dimension is $c-1$.

Temporal and distribution-shift diagnostics. On CIFAR-100/10T, we hold the old boundary classifier fixed and recompute an oracle old-data Hessian with the current feature extractor every 10 epochs. Old data are used only for measurement, never for training. Across tasks 2/5/10, the mean principal angle to the boundary support is at most 2.5×10^{-5} degrees and overlap is 1.000. Table 18 gives maximum angles and a separate comparison of old-data versus shifted-current-data estimates. In the latter comparison, maximum angles are at most 3×10^{-6} degrees and overlap is also 1.000, while spectrum cosine similarities are 0.996–0.999.

Table 18: CIFAR-100/10T subspace diagnostics. Temporal angles compare oracle supports during training to the boundary support; spectrum cosines compare old-data and current-data estimates.

Task	Rank	Max temporal angle (degrees)	Temporal overlap	Spectrum cosine
2	9	5.5×10^{-5}	1.000	0.996
5	39	1.2×10^{-4}	1.000	0.999
10	89	2.3×10^{-4}	1.000	0.998

A separate replay-free training comparison recomputes the eigensystem every 10 epochs using only current-task data. Acc/AAA changes from 49.51/62.293 with boundary computation to 49.68/61.779

with dynamic recomputation. The mixed change supports boundary computation in this setting, while distinguishing stable support from evolving curvature weights.

Long-horizon drift. For each old-task boundary, we freeze its classifier and Hessian and track later checkpoints. For the initial-to-final transition (task 0 to 9 in Figure 5a), applying only accumulated plastic drift changes old-task loss by at most 2.94×10^{-5}, consistent with exact invariance up to numerical precision. The anchor's full second-order Taylor prediction, including the linear term, retains correlation $r = 0.93$ with the true loss increase. This is a diagnostic of accumulated drift; training itself constrains adjacent-task feature changes and refreshes the Hessian at each boundary.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the contributions of SGR, efficient subspace identification, and SGPA. Propositions 3.1–3.2 and Theorem F.1 support the geometric claims under the stated fixed-classifier and quadratic-surrogate scope. Section 4 and Appendix E.6 report the empirical results and their variability.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 5 discusses validation-based coefficient selection, full-rank degeneration of the stable–plastic decomposition and SGPA correction, and the scope of the theoretical claims. Appendices E.3, E.4, and F provide details.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The Hessian derivation and proposition proofs are in Appendices C and D. Appendix F states the fixed-old-classifier scope, defines the local quadratic surrogate, proves Theorem F.1, and treats the full-rank case. Appendix F.4 establishes exact softmax invariance separately.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 4.1 describes datasets, splits, architecture, optimizers, learning rates, batch sizes, epochs, and coefficients. Algorithm 1 gives the training procedure. Appendix E.3 specifies the validation search, transfer to 20-task splits, baseline recipes, and optimizer comparison.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Code and reproduction instructions are publicly available at `https://github.com/bnufw/sgcl` (also linked in the abstract). All datasets used (CIFAR-100, Tiny-ImageNet, ImageNet-Subset, ImageNet-1K, CUB-200, ImageNet-A, and ImageNet-R) are publicly available.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 4.1 specifies the main training and evaluation settings. Appendix E.3 gives the candidate coefficient pairs, validation split, selection procedure, and baseline optimizer protocols; Appendices E.4 and E.5 describe the additional benchmark settings.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: Table 13 reports mean and standard deviation over three seeds for SGCL and DPCR on all six main settings; Table 12 does so for SD-LoRA with and without SGR. Table 17 reports standard deviations over 10 perturbation runs. Point estimates are identified separately, and the CIFAR-100 differences are described as within seed-level variation rather than significant wins.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 4.1 identifies the NVIDIA RTX 3090Ti GPU. Table 3 reports regularization-specific GFLOPS and memory; Appendix E.7 separately accounts for persistent statistics and the frozen previous backbone.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This research focuses on continual learning methodology using publicly available benchmark datasets. We have reviewed the NeurIPS Code of Ethics and confirm that our work conforms to it in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix A discusses the potential benefits of exemplar-free model updates, the privacy limitations of retained feature statistics, and potential risks arising from training-data biases.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper proposes a continual learning regularization method and does not release pre-trained models, generative models, or scraped datasets that pose risks for misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All datasets (CIFAR-100, Tiny-ImageNet, ImageNet-Subset, ImageNet-1K, CUB-200, ImageNet-A, and ImageNet-R), model backbones/adaptation modules (ResNet-18, ViT-B/16, LoRA), and baseline methods are cited with their original publications. Their licenses and terms of use are explicitly stated in Appendix E.9.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [N/A]

Justification: This paper does not release new datasets or pre-trained models. The contribution is a training methodology described in detail in the paper.

Guidelines:

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

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

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

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard component. The proposed SGCL method is a continual learning technique based on Hessian analysis and subspace decomposition.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.