STABILITY WHERE IT MATTERS: DECISION-SENSITIVE REPRESENTATION IN CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual learning with pre-trained models commonly mitigates forgetting by constraining parameters, gradients, or entire feature representations. While effective, these strategies do not explicitly distinguish representation changes that alter historical decisions from changes that are functionally irrelevant to them. We introduce Decision-Sensitive Drift Regularization (DSDR), a training-time principle that uses the decision geometry of previously learned classifiers to separate temporal representation drift into decision-sensitive and decision-null components. DSDR selectively suppresses only the former, directly linking representation stability to preservation of historical predictions while leaving complementary directions available for plasticity. The method requires no additional inference-time module and can be integrated into parameter-efficient continual adaptation without altering the inference procedure. Experiments under strict class-incremental learning on ImageNet-R across 5, 10, and 20-task settings, together with CIFAR-100 and DomainNet, demonstrate robust performance across task horizons with reduced forgetting and competitive final accuracy. These results support decision-sensitive drift as a simple and general principle for balancing stability and plasticity in continual learning. *Code will be made available upon final acceptance.*

1 INTRODUCTION

Continual learning (CL) aims to enable a model to acquire new knowledge sequentially while retaining what it has learned from previous data Parisi et al. (2019). Among its standard formulations, *class-incremental learning* (CIL) is particularly challenging: disjoint sets of classes are introduced over time, yet at inference the model must distinguish among all classes observed so far without access to task identity Rebuffi et al. (2017); Soutif-Cormerais et al. (2021). This requirement makes catastrophic forgetting especially severe and places the stability-plasticity dilemma at the center of CIL: a successful learner must remain sufficiently *stable* to preserve previously acquired knowledge while retaining enough *plasticity* to learn new classes.

Early continual-learning methods addressed forgetting through regularization, replay, and parameter isolation. Regularization-based approaches such as EWC Kirkpatrick et al. (2017), SI Zenke et al. (2017), and MAS Aljundi et al. (2018) restrict changes to parameters considered important for previously learned tasks. Replay-based methods Rebuffi et al. (2017); Lopez-Paz & Ranzato (2017); Chaudhry et al. (2018) revisit stored or generated information from earlier classes, whereas parameter-isolation methods Rusu et al. (2016); Mallya & Lazebnik (2018) reserve separate model capacity to reduce cross-task interference. While these strategies established effective mechanisms for mitigating forgetting, the growing use of large pre-trained models has changed the practical setting of continual learning. Repeatedly updating a large backbone, maintaining rehearsal memory, or increasing task-specific capacity can diminish the computational and storage advantages offered by pre-training, motivating more parameter-efficient forms of continual adaptation.

Parameter-efficient fine-tuning (PEFT) provides a natural solution by adapting pre-trained models through a small set of learnable parameters while leaving most backbone weights frozen. Prompt-based approaches such as L2P Wang et al. (2022b), DualPrompt Wang et al. (2022a), and CODA-Prompt Smith et al. (2023b) demonstrate that pre-trained Vision Transformers can be continually adapted without modifying the entire backbone. Low-Rank Adaptation (LoRA) Hu et al. (2022) offers another effective alternative by expressing task adaptation through compact low-rank updates. Building on this formulation, recent parameter-efficient continual-learning methods increasingly focus on controlling interference within the adaptation process itself. InfLoRA Liang & Li (2024) constructs low-rank update subspaces intended to avoid interference with previously learned knowledge, while SplitLoRA Qiu et al. (2026) explicitly partitions historical gradient space to balance stability and plasticity. EWC-LoRA Zheng et al. (2026) revisits parameter-importance regularization for shared low-rank updates, and DGS Li et al. (2026) combines gradient-space regulation with classifier and representation alignment.

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

These advances provide increasingly effective ways to control *how* a model is updated, yet they leave a different question less explored: *which representation changes produced by those updates actually need to be prevented?* Parameter-space regularization identifies weights that should change cautiously, while gradient-space methods constrain update directions before they are translated into feature changes. Representation-alignment methods instead encourage broader consistency of features across tasks. However, not every change in representation is equally relevant to a previously learned decision. Two old samples undergo similar amounts of feature drift while affecting their historical predictions very differently; conversely, a representation move substantially without changing any old-class decision if that movement occurs along directions to which the historical classifier is insensitive. Treating representation drift only by its magnitude therefore ignores the geometry through which that drift is converted into changes in prediction.

Recent studies provide evidence that this geometry is important, but from different perspectives. DPCR He et al. (2025) studies semantic shift in exemplar-free CIL and estimates historical class movement through dual projection before reconstructing the classifier. More recently, GOLD Lai et al. (2026a) shows in continual test-time adaptation that the row space of a linear classifier characterizes feature directions capable of altering its outputs. However, neither considers how past decision structure should constrain temporal feature drift during supervised CIL. These findings suggest that representation change should not be viewed as a homogeneous quantity. They also expose a question that is particularly relevant to strict class-incremental learning:

Rather than preserving representations indiscriminately, can we identify and stabilize only the component of temporal drift that can alter historical decisions?

We answer this question through *decision-sensitive representation drift*. The classifier learned from previous classes partitions the feature space into directions that can alter historical logits and complementary decision-null directions that are invisible to those linear decisions. This suggests that continual stability need not suppress representation evolution uniformly, but should instead protect the directions where drift can disturb previously learned decisions. Based on this view, we introduce **Decision-Sensitive Drift Regularization (DSDR)**, which uses the historical classifier to identify the decision-sensitive subspace and selectively penalize temporal feature drift within it. By leaving decision-null directions comparatively unconstrained, DSDR preserves flexibility for learning new classes while protecting prior decisions. The method is training-only, introduces no additional trainable parameters, can be integrated with existing PEFT-based continual learners, and leaves the inference architecture unchanged.

Our main contributions are summarized as follows:

- We introduce **Decision-Sensitive Drift Regularization (DSDR)**, which reframes continual representation stability by asking *where* representations drift, rather than penalizing temporal feature change uniformly.
- We derive a classifier-induced decomposition of temporal representation drift into decision-sensitive and decision-null components, and use it to selectively regularize only the directions that affect historical decisions.
- We demonstrate the effectiveness of DSDR across multiple class-incremental benchmarks, where it achieves strong results on ImageNet-R, CIFAR-100, and DomainNet. Extensive ablation and mechanistic analyses further validate the role of decision-sensitive drift in improving the stability-plasticity trade-off.

We validate DSDR under strict CIL on ImageNet-R across 5-, 10-, and 20-task sequences, together with CIFAR-100 and DomainNet. Controlled comparisons with full, decision-null, and rank-matched random drift regularization assess the proposed decision geometry, while integration with SplitLoRA shows that DSDR also complements gradient-space constraints. Collectively, these results support the central premise that, for continual stability, *where representations drift can matter as much as how much they drift*.

2 Related Work

2.1 Subspace-Based Continual Adaptation

Recent parameter-efficient continual-learning methods increasingly control interference through the geometry of the update space. InfLoRA Liang & Li (2024) constructs low-rank subspaces to reduce interference with previously learned knowledge, while SplitLoRA Qiu et al. (2026) partitions historical gradient space to balance stability and plasticity. Related approaches impose null-space or importance-based constraints on prompt or LoRA updates Zhang et al. (2025); Zheng et al. (2026), and DGS Li et al. (2026) combines gradient regulation with representation alignment.

These methods primarily constrain parameters or update directions before they are expressed as feature changes. DSDR instead operates on the resulting temporal representation drift and regularizes it according to its relevance to historical decisions.

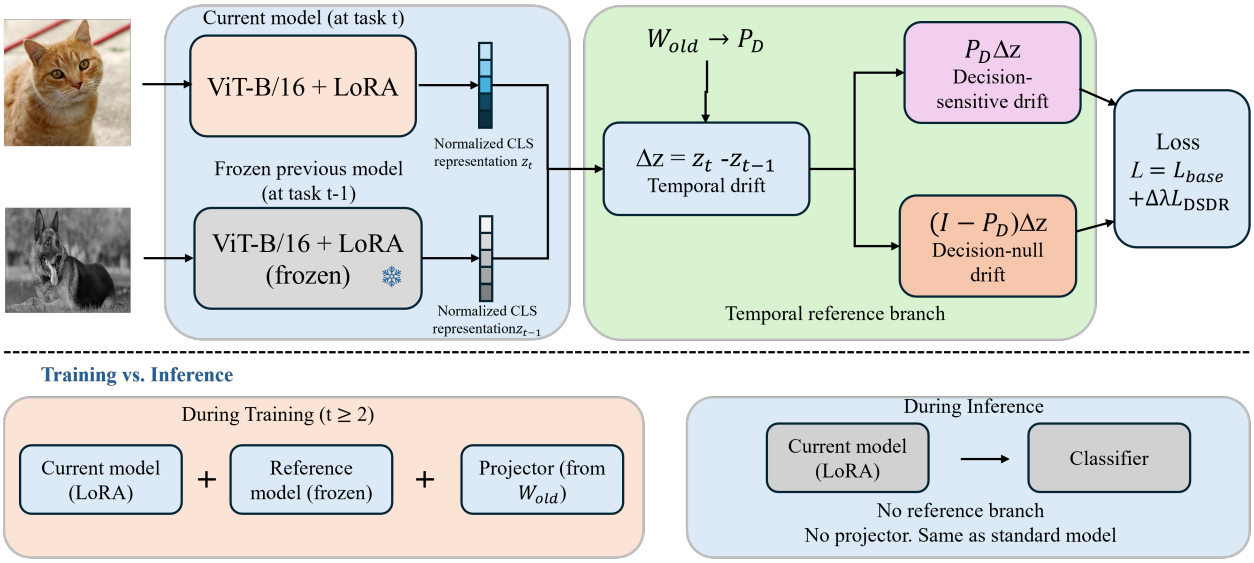


Figure 1: **Overview of Decision-Sensitive Drift Regularization (DSDR).** The current and frozen previous-task models produce representations whose difference defines temporal drift Δz. Using the historical classifier, DSDR decomposes this drift into decision-sensitive $P_D\Delta z$ and decision-null $(I - P_D)\Delta z$ components and regularizes only the former, preserving directions that affect historical decisions while retaining plasticity in the complementary subspace. The reference branch and projector are training-only and introduce no inference-time modification.

2.2 Representation Drift and Decision Geometry

Representation drift has also been addressed through explicit estimation or compensation of historical feature movement. DPCR He et al. (2025), for example, estimates semantic shift through projection-based modeling and uses it to recover historical classifier behavior. DSDR differs in that it does not reconstruct or compensate for drift after it occurs; it selectively constrains decision-relevant drift during learning.

The closest geometric connection is GOLD Lai et al. (2026a), which uses classifier row-space structure to identify feature directions capable of changing linear predictions in continual test-time adaptation. DSDR uses the same underlying geometric property for a different purpose: the historical classifier defines which temporal representation changes should be stabilized during supervised class-incremental learning. The contribution therefore lies in using classifier-induced geometry as a selective stability criterion rather than as a mechanism for enabling adaptation.

3 Preliminaries

3.1 Strict Class-Incremental Learning

We consider strict class-incremental learning (CIL) over a sequence of T tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$. At task t, the learner receives a dataset $\mathcal{D}_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{N_t}$, with a class set $\mathcal{Y}_t$ disjoint from the classes introduced in previous tasks. During training on $\mathcal{T}_t$, only the current-task data $\mathcal{D}_t$ are available. At inference, the model must discriminate among all classes observed so far,

$$\mathcal{Y}_{1:t} = \bigcup_{k=1}^{t} \mathcal{Y}_k, \tag{1}$$

without access to task identity, following the standard class-incremental protocol Rebuffi et al. (2017); Soutif-Cormerais et al. (2021). Let $f_{\boldsymbol{\theta}_t}$ denote the feature extractor after learning task t, and let $\mathbf{z}_t \in \mathbb{R}^d$ denote the image representation provided to the classifier. The classifier for the classes observed up to task t is denoted by $\mathbf{W}_t \in \mathbb{R}^{C_t \times d}$, where $C_t = |\mathcal{Y}_{1:t}|$, giving logits $\boldsymbol{\ell}_t = \mathbf{W}_t\mathbf{z}_t + \mathbf{b}_t$. The strict-CIL prediction is obtained through a global argmax over all classes seen so far.

3.2 Parameter-Efficient Representation Learning

We adopt a frozen pre-trained Vision Transformer (ViT) Dosovitskiy et al. (2021) and perform continual adaptation using Low-Rank Adaptation (LoRA) Hu et al. (2022). For a pre-trained linear transformation $\mathbf{W}_0$, LoRA parameterizes its task-dependent update as

$$\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}, \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{r \times d_{\text{in}}}$ and $\mathbf{B} \in \mathbb{R}^{d_{\text{out}} \times r}$ with $r \ll \min(d_{\text{in}}, d_{\text{out}})$. Such low-rank adaptation has become a common mechanism for parameter-efficient continual learning, with methods such as InfLoRA Liang & Li (2024) and SplitLoRA Qiu et al. (2026) further constraining the update subspace to reduce interference with previously learned tasks.

Representation used by DSDR. reasons about how feature changes affect the classifier, the representation must be defined at the interface between the feature extractor and the classifier. In a ViT, an input image is divided into patches, embedded, and processed jointly with a learnable classification token through a sequence of Transformer blocks. The state of the classification token after the final Transformer encoder and its final normalization serves as the image-level representation used by the classification head. Accordingly, throughout this work we define $\mathbf{z}_t = f_{\boldsymbol{\theta}_t}(\mathbf{x}) \in \mathbb{R}^d$ as the *final normalized class representation immediately before the linear classifier*. For the ViT-B/16 backbone used in our experiments, $d = 768$.

This choice is important because the historical classifier $\mathbf{W}_{\text{old}}$ acts directly on this representation. Thus, the geometry induced by $\mathbf{W}_{\text{old}}$ has an exact interpretation at this location: directions in the representation space can be analyzed according to whether they are visible or invisible to historical logits. We therefore apply DSDR at this pre-classifier representation by default.

4 Decision-Sensitive Drift Regularization

4.1 Architectural Overview

Existing PEFT-based continual-learning methods primarily control how model parameters or gradients are updated. InfLoRA constructs low-rank adaptation subspaces designed to reduce interference with previous tasks Liang & Li (2024), while SplitLoRA Qiu et al. (2026) explicitly partitions historical gradient space to balance stability and plasticity. However, an update-space constraint does not directly specify how the resulting optimization changes the representation used by the historical classifier. DSDR addresses this downstream effect. Its objective is to identify, among the representation changes produced while learning a new task, those that are capable of modifying previously learned decisions. The complete framework is illustrated in Fig. 1 and consists of two components: a *Temporal Reference Branch* and a *Decision-Sensitive Drift Regularizer*.

At the beginning of each task $t > 1$, we retain a frozen copy of the model learned after task $t-1$. Using a previous network as a fixed reference is a standard strategy in rehearsal-free continual learning, where historical behavior must be preserved without directly revisiting old-task training data . In DSDR, however, this frozen model is not used to distill complete predictions or representations. Instead, it provides a temporal anchor against which the representation change induced by current-task learning is measured. For an input $\mathbf{x} \sim \mathcal{D}_t$, the same image is therefore processed by two branches. Both representations are extracted at the same pre-classifier location defined in Sec. 3.2: the final normalized class representation. The difference between the two representations describes how current-task optimization moves the image in feature space. Rather than constraining this change uniformly, DSDR uses the classifier learned for previous classes to isolate the component that can affect old logits. Only this component is penalized. The frozen branch and decision-space operator are required only during training and are removed after the task is learned; inference therefore retains the architecture and parameter count of the underlying continual learner.

4.2 Temporal Reference branch

The first challenge is to determine how current-task optimization changes the representation learned before the task. To directly compare current representations with historical examples we employed the previous model as a temporal reference. Before training task t, we freeze the feature extractor obtained after task $t-1$, denoted by $f_{\bar{\boldsymbol{\theta}}_{t-1}}$. For each current-task input $\mathbf{x}$, the frozen and current models are evaluated at the same pre-classifier representation:

$$\bar{\mathbf{z}} = f_{\bar{\boldsymbol{\theta}}_{t-1}}(\mathbf{x}), \qquad \mathbf{z} = f_{\boldsymbol{\theta}_t}(\mathbf{x}). \tag{3}$$

The temporal representation drift is then defined as $\Delta\mathbf{z} = \mathbf{z} - \text{sg}(\bar{\mathbf{z}})$, where $\text{sg}(\cdot)$ denotes stop-gradient. The reference branch is kept frozen and in evaluation mode, so gradients propagate only through the current model. Importantly, $\|\Delta\mathbf{z}\|_2$ measures how much the representation has changed, but not whether that change affects previously learned decisions. For a linear classifier, only particular directions of the representation space can alter its outputs. We therefore collect the classifier weights corresponding to all classes learned before task t into $\mathbf{W}_{\text{old}} \in \mathbb{R}^{C_{\text{old}} \times d}$. For a shared classifier, $\mathbf{W}_{\text{old}}$ contains the preserved rows corresponding to historical classes. When separate task heads are used, their frozen weight matrices can be concatenated. In either case, $\mathbf{W}_{\text{old}}$ acts directly on the same d-dimensional pre-classifier representation used in equation 3.

To see how representation drift affects historical decisions, consider the historical logits associated with the reference and current representations:

$$\bar{\boldsymbol{\ell}}_{\text{old}} = \mathbf{W}_{\text{old}}\bar{\mathbf{z}}, \qquad \boldsymbol{\ell}_{\text{old}} = \mathbf{W}_{\text{old}}\mathbf{z}. \tag{4}$$

Their difference is

$$\Delta\boldsymbol{\ell}_{\text{old}} = \boldsymbol{\ell}_{\text{old}} - \bar{\boldsymbol{\ell}}_{\text{old}} = \mathbf{W}_{\text{old}}\Delta\mathbf{z}. \tag{5}$$

Table 1: FAA (%) and CAA (%) on ImageNet-R under 5-, 10-, and 20-task class-incremental settings. All backbone networks are pre-trained on ImageNet-21K. Results are reported as mean ± standard deviation over three random seeds. Best results are in **best**.

Method	Pub.	5-task		10-task		20-task	
		FAA (↑)	CAA (↑)	FAA (↑)	CAA (↑)	FAA (↑)	CAA (↑)
Upper-bound	-	84.09 ± 0.21	-	84.09 ± 0.21	-	84.09 ± 0.21	-
FT	-	18.74 ± 0.44	48.39 ± 0.58	10.12 ± 0.51	35.23 ± 0.92	4.75 ± 0.40	22.80 ± 0.37
FT++	-	60.42 ± 0.87	71.59 ± 0.50	48.93 ± 1.15	66.79 ± 0.92	35.98 ± 1.38	59.68 ± 0.95
L2P++ Wang et al. (2022b)	CVPR22	70.83 ± 0.58	78.34 ± 0.47	69.29 ± 0.73	78.30 ± 0.69	65.89 ± 1.30	77.15 ± 0.65
Deep L2P++ Wang et al. (2022b)	CVPR22	73.93 ± 0.37	80.14 ± 0.54	71.66 ± 0.64	79.63 ± 0.90	68.42 ± 1.20	78.68 ± 1.03
DualPrompt Wang et al. (2022a)	ECCV22	73.05 ± 0.50	79.47 ± 0.40	71.32 ± 0.62	78.94 ± 0.72	67.87 ± 1.39	77.42 ± 0.80
CODA-P Smith et al. (2023b)	CVPR23	76.51 ± 0.38	82.04 ± 0.54	75.45 ± 0.56	81.59 ± 0.82	72.37 ± 1.19	79.88 ± 1.06
HiDe-Prompt Wang et al. (2023)	NeurIPS23	76.29 ± 0.10	78.77 ± 0.11	76.74 ± 0.18	78.76 ± 0.11	76.46 ± 0.06	78.76 ± 0.11
EvoPrompt Kurniawan et al. (2024)	AAAI24	77.16 ± 0.18	82.22 ± 0.54	76.83 ± 0.08	82.09 ± 0.68	74.41 ± 0.23	80.96 ± 1.42
InfLoRA Liang & Li (2024)	CVPR24	79.82 ± 0.27	84.07 ± 0.48	78.10 ± 0.43	83.47 ± 1.23	73.81 ± 0.47	81.02 ± 0.56
VQ-Prompt Jiao et al. (2024)	NeurIPS24	79.23 ± 0.29	82.96 ± 0.50	78.71 ± 0.22	83.24 ± 0.68	78.10 ± 0.22	82.70 ± 1.16
VPT-NSP^2 Lu et al. (2024)	NeurIPS24	79.71 ± 0.22	84.54 ± 0.68	79.35 ± 0.19	84.92 ± 0.41	76.72 ± 0.44	82.91 ± 0.60
SD-LoRA Wu et al. (2025)	ICLR25	79.15 ± 0.20	83.01 ± 0.42	77.34 ± 0.35	82.04 ± 0.24	75.26 ± 0.37	80.22 ± 0.72
SplitLoRA Qiu et al. (2026)	ICLR26	81.92 ± 0.29	85.83 ± 0.55	81.00 ± 0.17	85.84 ± 0.62	78.82 ± 0.28	84.57 ± 0.44
DSDR (Ours)	-	$\mathbf{83.19 \pm 0.33}$	$\mathbf{87.21 \pm 0.44}$	$\mathbf{81.62 \pm 0.33}$	$\mathbf{86.65 \pm 0.29}$	$\mathbf{78.88 \pm 0.57}$	$\mathbf{84.69 \pm 0.37}$

Thus, a representation direction can affect historical logits only if it is visible through $\mathbf{W}_{\text{old}}$. In particular, any direction $\mathbf{v}$ satisfying $\mathbf{W}_{\text{old}}\mathbf{v} = \mathbf{0}$ lies in the null space of the historical classifier and produces no historical logit change. Conversely, the row space of $\mathbf{W}_{\text{old}}$ contains the feature directions observable by the historical classifier. We therefore refer to it as the *historical decision-sensitive subspace*. This interpretation is consistent with analyses of classifier-induced feature geometry Lai et al. (2026a) and related projection-based treatments of semantic shift in exemplar-free CIL He et al. (2025).

The exact orthogonal projector onto this subspace is

$$\mathbf{P}_D = \mathbf{W}_{\text{old}}^{\dagger}\mathbf{W}_{\text{old}}, \tag{6}$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudoinverse. The temporal drift can consequently be decomposed as

$$\Delta\mathbf{z} = \underbrace{\mathbf{P}_D\Delta\mathbf{z}}_{\Delta\mathbf{z}_D} + \underbrace{(\mathbf{I} - \mathbf{P}_D)\Delta\mathbf{z}}_{\Delta\mathbf{z}_N}, \tag{7}$$

where $\Delta\mathbf{z}_D$ denotes decision-sensitive drift and $\Delta\mathbf{z}_N$ denotes decision-null drift. The distinction between these two components follows from the Moore-Penrose identity $\mathbf{W}_{\text{old}}\mathbf{W}_{\text{old}}^{\dagger}\mathbf{W}_{\text{old}} = \mathbf{W}_{\text{old}}$. Using equation 6,

$$\begin{aligned}\mathbf{W}_{\text{old}}(\mathbf{I} - \mathbf{P}_D) &= \mathbf{W}_{\text{old}} - \mathbf{W}_{\text{old}}\mathbf{W}_{\text{old}}^{\dagger}\mathbf{W}_{\text{old}} \\ &= \mathbf{0}.\end{aligned} \tag{8}$$

Therefore, $\mathbf{W}_{\text{old}}\Delta\mathbf{z}_N = \mathbf{0}$. Substituting equation 7 into equation 5 yields

$$\begin{aligned}\Delta\boldsymbol{\ell}_{\text{old}} &= \mathbf{W}_{\text{old}}\left(\Delta\mathbf{z}_D + \Delta\mathbf{z}_N\right) \\ &= \mathbf{W}_{\text{old}}\Delta\mathbf{z}_D \\ &= \mathbf{W}_{\text{old}}\mathbf{P}_D\Delta\mathbf{z}.\end{aligned} \tag{9}$$

equation 9 establishes the key distinction used by DSDR: under the frozen historical linear classifier, decision-sensitive drift can alter historical logits, whereas decision-null drift is invisible to them. The resulting decomposition identifies *where* temporal representation change is relevant to previously learned decisions.

4.3 Decision-Sensitive Drift Regularization

The decomposition above identifies the component of representation drift that is visible to historical classifiers. The remaining question is how to preserve this component without unnecessarily constraining representation plasticity. A conventional feature-consistency objective, $\|\Delta\mathbf{z}\|_2^2$, penalizes the entire temporal displacement and therefore suppresses both decision-sensitive and decision-null movement. DSDR instead regularizes only the decision-sensitive component identified in Sec. 4.2.

For numerical stability, we implement the decision operator using the ridge-stabilized form $\widetilde{\mathbf{P}}_D = \mathbf{W}_{\text{old}}^{\top}\left(\mathbf{W}_{\text{old}}\mathbf{W}_{\text{old}}^{\top} + \epsilon_p\mathbf{I}\right)^{-1}\mathbf{W}_{\text{old}}$, where $\epsilon_p > 0$ is a small stabilization term. Unlike the exact projector $\mathbf{P}_D$, the

Table 2: FAA (%) and CAA (%) on CIFAR-100 with 10 tasks and DomainNet with 5 tasks. Results are reported over three random seeds. *Note:* CIFAR-100 uses low-resolution 32×32 images, which can yield more compressed representations and make decision geometry less informative; this contributes to the stronger performance of gradient-space SplitLoRA in this setting.

Method	**Pub.**	**CIFAR-100**		**DomainNet**	
		FAA (↑)	CAA (↑)	FAA (↑)	CAA (↑)
Upper-bound	-	91.92 ± 0.05	-	90.12 ± 0.13	-
DualPrompt Wang et al. (2022a)	ECCV22	84.42 ± 0.30	90.06 ± 0.07	72.14 ± 0.05	77.71 ± 0.06
CODA-Prompt Smith et al. (2023b)	CVPR23	86.62 ± 0.11	91.08 ± 0.28	73.23 ± 0.13	78.72 ± 0.07
LAE Gao et al. (2023)	ICCV23	84.15 ± 0.16	89.84 ± 0.03	66.85 ± 0.40	75.01 ± 0.17
C-LoRA Smith et al. (2023a)	TMLR24	82.97 ± 0.47	88.81 ± 0.34	69.34 ± 0.16	75.25 ± 0.11
InfLoRA Liang & Li (2024)	CVPR24	87.06 ± 0.25	91.59 ± 1.43	78.26 ± 0.50	78.82 ± 0.34
VPT-NSP^2 Lu et al. (2024)	NeurIPS24	88.04 ± 0.11	92.25 ± 0.80	83.83 ± 0.19	88.63 ± 0.10
CoSO Cheng et al. (2026)	CVPR25	88.76 ± 0.16	92.99 ± 0.23	74.27 ± 0.07	80.05 ± 0.04
SplitLoRA Qiu et al. (2026)	ICLR26	$\mathbf{90.33 \pm 0.73}$	$\mathbf{93.70 \pm 0.32}$	84.31 ± 0.23	88.99 ± 0.57
DSDR (Ours)	-	88.84 ± 0.58	93.23 ± 0.51	$\mathbf{85.46 \pm 0.61}$	$\mathbf{90.34 \pm 0.62}$

ridge-stabilized operator is not exactly idempotent; however, its output remains in $\mathrm{Row}(\mathbf{W}_{\mathrm{old}})$ and it annihilates directions in the exact null space of the historical classifier. The decision-sensitive drift used during training is therefore $\widetilde{\Delta \mathbf{z}}_D = \widetilde{\mathbf{P}}_D \Delta \mathbf{z}$. Rather than matching historical logits directly, DSDR acts in representation space by discouraging temporal movement along directions observable by the historical classifier.

To reduce sensitivity to the absolute scale of the representation, we normalize the projected drift by the energy of the frozen reference features. For a mini-batch $\mathcal{B}$ of size B, the DSDR loss is

$$\mathcal{L}_{\mathrm{DSDR}} = \frac{\frac{1}{B}\sum_{i=1}^{B} \left\| \widetilde{\mathbf{P}}_D \left(\mathbf{z}_i - \bar{\mathbf{z}}_i \right) \right\|_2^2}{\frac{1}{B}\sum_{i=1}^{B} \|\bar{\mathbf{z}}_i\|_2^2 + \delta}, \tag{10}$$

where δ is a small constant for numerical stability. The normalization expresses temporal drift relative to the reference feature scale rather than in absolute coordinates. The selective nature of the objective can also be seen from its gradient. For a single sample and omitting the constant normalization term for clarity,

$$\nabla_{\mathbf{z}} \left\| \widetilde{\mathbf{P}}_D \Delta \mathbf{z} \right\|_2^2 = 2 \widetilde{\mathbf{P}}_D^{\top} \widetilde{\mathbf{P}}_D \Delta \mathbf{z}. \tag{11}$$

Because the range of $\widetilde{\mathbf{P}}_D^{\top}$ is contained in $\mathrm{Row}(\mathbf{W}_{\mathrm{old}})$, the DSDR restoring gradient acts only within the historical decision-sensitive space. For the exact orthogonal projector, equation 11 reduces to

$$\nabla_{\mathbf{z}} \|\mathbf{P}_D \Delta \mathbf{z}\|_2^2 = 2 \mathbf{P}_D \Delta \mathbf{z}, \tag{12}$$

since $\mathbf{P}_D^{\top} = \mathbf{P}_D$ and $\mathbf{P}_D^2 = \mathbf{P}_D$. DSDR therefore differs from global feature preservation in what it seeks to stabilize. Rather than forcing the complete historical representation to remain unchanged, it selectively discourages temporal feature changes that are visible to previously learned classifiers, while leaving decision-null directions unconstrained by the DSDR term and available to the base learning objective.

4.4 Overall Learning Objective

DSDR is designed as a complementary regularizer and does not replace the optimization objective of the underlying continual learner. Let $\mathcal{L}_{\mathrm{base}}$ denote its current-task learning objective. The complete objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{base}} + \lambda \mathcal{L}_{\mathrm{DSDR}}, \tag{13}$$

where λ controls the strength of historical decision preservation. During the first task, no historical classifier exists and DSDR is inactive, i.e., $\mathcal{L} = \mathcal{L}_{\mathrm{base}}$. For every subsequent task, the frozen temporal reference and historical decision operator are constructed once before optimization begins. During each training iteration, the reference feature is computed without gradient tracking, while gradients from both $\mathcal{L}_{\mathrm{base}}$ and $\mathcal{L}_{\mathrm{DSDR}}$ update only the current trainable parameters.

After completing the task, the current model becomes the reference model for the next incremental stage and the newly learned classifier rows are incorporated into $\mathbf{W}_{\mathrm{old}}$. Neither the frozen reference branch nor the decision-space operator is required at test time. Consequently, DSDR adds no trainable parameters and leaves both the inference architecture and inference complexity of the underlying continual learner unchanged.

Table 3: Ablation of the drift constraint on ImageNet-R under the 5-task setting. All regularized variants use $\lambda = 2$ and seed 2024.

Variant	FAA (↑)	CAA (↑)	Forget. (↓)
No DSDR ($\lambda = 0$)	78.64	82.91	6.18
Full drift	77.26	81.74	**2.36**
Random subspace	82.47	86.58	3.54
Decision-null	80.31	84.62	5.27
DSDR	**83.22**	**87.33**	3.10

Table 4: Sensitivity to the DSDR regularization coefficient λ on ImageNet-R under the 5-task setting. All runs use the same seed, task order, and training protocol, with only λ varied. * denotes the default value used in the main experiments.

λ	FAA (↑)	CAA (↑)	Forget. (↓)
0	78.64	82.91	6.18
0.25	80.47	84.36	5.14
0.5	81.36	**87.51**	4.37
1	82.41	86.78	3.62
2*	**83.22**	87.33	**3.10**
4	82.84	86.91	3.39
8	82.17	86.24	3.91

5 Experiments

We evaluate DSDR on three standard class-incremental learning benchmarks and compare it with state-of the art continual learning methods under the same pre-trained ViT setting.

5.1 Experimental Settings

Datasets and Evaluation Metrics We conduct experiments on three widely used continual learning benchmarks: ImageNet-R Russakovsky et al. (2015), CIFAR-100 Krizhevsky & Hinton (2009), and DomainNet Peng et al. (2019). Following prior PEFT-based continual learning protocols Gao et al. (2023); Wang et al. (2022b); Liang & Li (2024); Qiu et al. (2026), ImageNet-R is evaluated under 5, 10, and 20-task settings, corresponding to 40, 20, and 10 new classes per task, respectively. CIFAR-100 is divided into 10 tasks of 10 classes each, while DomainNet is uniformly partitioned into 5 tasks following the protocol used by prior work. In all cases, class sets are disjoint across tasks and evaluation follows the strict class-incremental setting, where task identity is unavailable at inference and predictions are made over all classes observed so far.

We compare DSDR with representative state-of-the-art continual learning methods in Tables 1 and 2. Following the protocol of prior PEFT-based work Qiu et al. (2026), all of our models are evaluated using the same ImageNet-21K-pretrained ViT-B/16 setting, while results for previously published methods are taken from the SplitLoRA benchmark under the corresponding protocol. The *Upper-bound* denotes joint training on all classes simultaneously. We report Final Average Accuracy (FAA) and Cumulative Average Accuracy (CAA), where FAA measures average accuracy over all tasks after the complete sequence and CAA summarizes performance throughout the learning trajectory. Our results are reported as mean $\pm$ standard deviation over three random seeds.

Implementation Details. Following the experimental protocol of recent PEFT-based continual learning methods Liang & Li (2024); Qiu et al. (2026), we use ViT-B/16 Dosovitskiy et al. (2021) pre-trained on ImageNet-21K as the backbone. The backbone is frozen during continual learning and has an embedding dimension of $D = 768$. LoRA modules are inserted into the key and value projections of all 12 Transformer blocks, with rank $r = 10$ and scaling factor $\alpha = 20$.

We optimize the trainable parameters using AdamW with an initial learning rate of 10^{-3} for LoRA parameters and 10^{-2} for the classification head, and use weight decay 5×10^{-5}. Each task is trained for 10 epochs with a batch size of 256 and 224×224 input images. We use the same data augmentation and optimization schedule as the official SplitLoRA protocol. For DSDR, we set the regularization coefficient to $\lambda = 2$ and the ridge scaling coefficient to $\eta = 10^{-3}$. The previous-task model is discarded at inference. model architecture or inference computation. All experiments are conducted on a single NVIDIA RTX PRO 6000 GPU. The main results reported in Tables 1 and 2 are averaged over three random seeds. All ablation studies and experimental analyses are conducted using a fixed seed of 2024; accordingly.

5.2 Experimental Results

Results across Benchmarks. Tables 1 and 2 evaluate DSDR across ImageNet-R, CIFAR-100, and DomainNet. On ImageNet-R, DSDR achieves the highest FAA and CAA across the 5-, 10-, and 20-task settings, with the largest gains over SplitLoRA in the 5-task configuration and smaller but consistent improvements as the sequence becomes longer. This shows that the benefit of decision-sensitive drift regularization persists across different task granularities.

Table 5: Sample-level analysis of historical prediction changes on ImageNet-R. Results aggregate 12,061 old-sample observations across the four task transitions.

Metric	$\lambda = 0$	**DSDR**
Forgotten samples (%)	0.78	**0.52**
Prediction flips (%)	2.60	**1.81**
$\rho(D_{\text{dec}}, \text{flip})$	0.132	0.093
$\rho(D_{\text{null}}, \text{flip})$	0.090	0.057

Table 6: Practical overhead of DSDR. The temporal reference and decision regularizer are used only during training and do not modify the inference model.

Metric	**SplitLoRA**	**+DSDR**
Inference params.	87.80M	87.80M
Compute	20.44 GMACs	20.44 GMACs
Latency (BS1)	3.93 ms	3.91 ms
Training time	1.20 h	1.57 h

Table 7: Comparison of SplitLoRA and DSDR on ImageNet-R under the 5-task class-incremental setting. Higher FAA and CAA are better, while lower average forgetting is better.

Method	**FAA** (↑)	**CAA** (↑)	**Avg. Forget.** (↓)
SplitLoRA	80.87	86.89	7.54
SplitLoRA + DSDR	**83.03**	**88.26**	**5.44**

DSDR also performs best on DomainNet, improving both FAA and CAA over the compared methods, indicating that the approach remains effective under substantial visual and domain variation. *On CIFAR-100, the native* 32×32 *images are substantially lower resolution and must be upsampled to the common* 224×224 *ViT input, which can yield comparatively compressed final representations and make the historical classifier row space a less expressive proxy for interference-relevant directions. In contrast, ImageNet-R and DomainNet originate from higher-resolution imagery and retain richer visual structure when processed at* 224×224*, providing a more informative representation geometry for DSDR. This difference help explain why SplitLoRA's direct gradient-space constraint is particularly effective on CIFAR-100, whereas DSDR shows stronger gains on ImageNet-R and DomainNet.* Overall, the results show that DSDR provides a consistent source of stability across most evaluated settings while retaining the same pre-trained backbone and low-rank adaptation framework.

5.3 Ablation Studies and Mechanistic Analysis

We analyze DSDR along four dimensions: whether its effectiveness depends on the choice of representation subspace, how sensitive it is to the regularization strength, whether the proposed decomposition is reflected in historical prediction changes, and whether it provides complementary benefits to gradient-space constraints.

Effect of the Drift Subspace. We first isolate the role of the subspace in which representation drift is constrained. All variants in Table 3 use the same backbone, LoRA configuration, classifier, task sequence, optimization settings, and training protocol; only the form of the drift regularizer is changed. The unregularized variant removes DSDR entirely, while the full-drift variant penalizes the complete temporal representation change. The random-subspace variant applies the same regularization to a rank-matched random projection, controlling for both regularization strength and subspace dimensionality. The decision-null variant instead constrains the component orthogonal to the historical decision space.

The results show that suppressing representation movement indiscriminately is not sufficient. Full-drift regularization produces the lowest forgetting but substantially degrades both final and continual accuracy, indicating that excessive preservation restricts the plasticity required for learning new classes. Restricting an arbitrary low-dimensional subspace performs considerably better, but remains below DSDR, showing that the gain cannot be explained solely by reducing the dimensionality of the constrained feature space. Regularizing the decision-null component is also less effective, suggesting that preserving directions with limited influence on historical classifier outputs does not provide the same stability benefit. In contrast, DSDR achieves the strongest overall balance between retention and adaptation. This ablation supports the central design choice of constraining representation drift according to its relevance to historical decisions rather than its magnitude alone.

Sensitivity to the Regularization Strength. Table 4 evaluates the regularization coefficient λ while keeping the seed, task order, and all other training settings fixed. Performance improves substantially once decision-sensitive drift is regularized, and remains competitive over a broad range of non-zero coefficients. Moderate regularization provides the best trade-off, whereas larger values gradually reduce accuracy, consistent with an increasingly restrictive constraint on representation adaptation. The default setting $\lambda = 2$ is selected because it provides the strongest final accuracy and forgetting behavior while maintaining high continual accuracy. Overall, the method does not rely on a

narrowly tuned coefficient, although excessively strong regularization begins to shift the stability-plasticity balance toward preservation.

Relationship Between Drift and Historical Prediction Changes. We next examine whether the classifier-induced decomposition corresponds to actual changes in historical predictions. At every task boundary, all available samples from previously learned classes are evaluated using the model immediately before and after learning the new task. For each sample, we record whether a previously correct prediction becomes incorrect, whether the predicted class changes, and the magnitudes of its decision-sensitive and decision-null drift. Aggregating observations across the four transitions of the 5-task sequence yields the sample-boundary population reported in Table 5; thus, n counts observations across transitions rather than unique images.

DSDR reduces both forgetting events and prediction flips, showing that constraining decision-sensitive drift translates into greater stability at the output level. More importantly, decision-sensitive drift exhibits a stronger association with prediction changes than decision-null drift under both the unregularized and regularized models. This ordering is preserved even though the correlations become smaller after applying DSDR, which is expected as the method reduces both decision-sensitive movement and the number of resulting prediction changes. The correlations are modest and therefore should not be interpreted as establishing a direct causal relationship. Rather, they provide sample-level evidence that the two components of the decomposition differ in their functional relevance to historical predictions.

Practical Overhead. The efficiency analysis in Table 6 separates training-time regularization from deployment-time cost. DSDR uses a frozen temporal reference and the historical decision operator only while learning a new task; neither component is retained in the inference graph. Consequently, the number of deployed parameters and inference operations is unchanged, and the small measured latency difference falls within normal profiling variation. The computational cost of DSDR is therefore confined to training, where the additional reference forward pass and drift computation increase wall-clock time. This property allows DSDR to improve continual-learning stability without modifying the deployed architecture or increasing inference complexity.

Complementarity with Gradient-Space Constraints. Finally, Table 7 evaluates whether DSDR remains useful when combined with SplitLoRA. The two methods constrain different stages of the learning process: SplitLoRA limits interference in the parameter-update space, whereas DSDR constrains the resulting evolution of representations according to their effect on historical decisions. Adding DSDR to SplitLoRA consistently improves continual performance and reduces forgetting, showing that gradient-space separation alone does not eliminate decision-sensitive representation drift. These results indicate that representation-level decision preservation and gradient-space interference reduction capture complementary aspects of continual learning. Importantly, this complementarity is obtained without introducing additional inference-time parameters or computation.

6 Conclusion

We introduced **Decision-Sensitive Drift Regularization (DSDR)**, a representation-space approach to continual learning that distinguishes historically consequential feature changes from drift that is invisible to previous classifiers. Using the historical classifier geometry, DSDR decomposes temporal representation drift into decision-sensitive and decision-null components and selectively regularizes only the former. This provides a direct way to preserve previous decisions without uniformly constraining representation evolution. Across strict class-incremental settings, DSDR improves the stability-plasticity trade-off and remains effective across different task horizons. Controlled ablations show that constraining the decision-sensitive subspace is more effective than full-drift, decision-null, or rank-matched random alternatives, while sample-level analysis links decision-sensitive drift more strongly to historical prediction changes. DSDR also complements gradient-space methods such as SplitLoRA, indicating that stability can be controlled at both the update and representation levels. Since the regularizer is required only during training, DSDR introduces no additional inference-time parameters or operations. Overall, our results support the central premise that in continual learning, *where representations drift can matter as much as how much they drift*.

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)*, pp. 139–154, 2018.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Quan Cheng, Yuanyu Wan, Lingyu Wu, Chenping Hou, and Lijun Zhang. Continuous subspace optimization for continual learning. *Advances in Neural Information Processing Systems*, 38:15376–15398, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11483–11493, 2023.

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.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Li Jiao, Qiuxia Lai, Yu Li, and Qiang Xu. Vector quantization prompting for continual learning. *Advances in Neural Information Processing Systems*, 37:34056–34076, 2024.

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 and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Muhammad Rifki Kurniawan, Xiang Song, Zhiheng Ma, Yuhang He, Yihong Gong, Yang Qi, and Xing Wei. Evolving parameterized prompt memory for continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13301–13309, 2024.

Guannan Lai, Da-Wei Zhou, Zhenguo Li, and Han-Jia Ye. The golden subspace: Where efficiency meets generalization in continual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3866–3875, June 2026a.

Guannan Lai, Da-Wei Zhou, Zhenguo Li, and Han-Jia Ye. The golden subspace: Where efficiency meets generalization in continual test-time adaptation. *arXiv preprint arXiv:2603.21928*, 2026b.

Kai Li, Jiafeng Li, Lianghua He, and Ying Wen. Dgs: Dual gradient and semantic-shift guided low-rank adaptation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32267–32277, 2026.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Yue Lu, Shizhou Zhang, De Cheng, Yinghui Xing, Nannan Wang, Peng Wang, and Yanning Zhang. Visual prompt tuning in null space for continual learning. *Advances in neural information processing systems*, 37:7878–7901, 2024.

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*, pp. 7765–7773, 2018.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

Haomiao Qiu, Miao Zhang, Ziyue Qiao, Weili Guan, Min Zhang, and Liqiang Nie. Splitlora: Balancing stability and plasticity in continual learning through gradient space splitting. In *International Conference on Learning Representations*, volume 2026, pp. 48458–48480, 2026.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *arXiv preprint arXiv:2304.06027*, 2023a.

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 (CVPR)*, pp. 11909–11919, June 2023b.

Albin Soutif-Cormerais, Marc Masana, Joost Van de Weijer, and Bartłomiej Twardowski. On the importance of cross-task features for class-incremental learning. *arXiv preprint arXiv:2106.11930*, 1, 2021.

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *Advances in Neural Information Processing Systems*, 36:69054–69076, 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*, pp. 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*, pp. 139–149, 2022b.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. S-lora: Scalable low-rank adaptation for class incremental learning. *arXiv e-prints*, pp. arXiv–2501, 2025.

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

Shizhou Zhang, Yue Lu, De Cheng, Yinghui Xing, Nannan Wang, Peng Wang, and Yanning Zhang. Vpt-nsp 2++: Importance-aware visual prompt tuning in null space for continual learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Yaoyue Zheng, Yin Zhang, Joost van de Weijer, Gido van de Ven, Shaoyi Du, Xuetao Zhang, and Zhiqiang Tian. Revisiting weight regularization for low-rank continual learning. In *International Conference on Learning Representations*, volume 2026, pp. 42895–42919, 2026.

A Appendix

A.1 Relation to Existing Subspace-Based Methods

Table A. 8 highlights that the main distinction of DSDR lies not in introducing subspace geometry itself, but in how that geometry is used. SplitLoRA operates in gradient space and constrains update directions, DPCR uses representation projections to estimate and correct semantic shift, and GOLD uses classifier row-space geometry as the space for test-time adaptation. In contrast, DSDR uses the historical classifier geometry as a *temporal stability criterion*: it identifies which representation changes can affect previous decisions and regularizes only those changes. This makes DSDR complementary to update-space methods, while remaining distinct from approaches that use classifier geometry primarily for adaptation or semantic correction.

Table 8: Comparison with closely related geometric approaches. The methods differ in the continual-learning problem they address, the space in which geometry is defined, and the quantity being constrained.

Method	**Setting**	**Problem addressed**	**Geometry acts on**	**Geometric constraint**
SplitLoRA Qiu et al. (2026)	CIL	Stability-plasticity of parameter updates	Gradients / LoRA updates	Separates stable and plastic update directions in gradient space.
DPCR He et al. (2025)	EFCIL	Historical semantic shift	Class representations	Uses projection-based correction to compensate representation shift.
GOLD Lai et al. (2026b)	CTTA	Efficient test-time adaptation	Feature adaptation directions	Uses classifier-sensitive geometry to guide adaptation.
DSDR (Ours)	CIL	**Temporal decision stability**	**Representation drift**	**Orthogonally decomposes temporal drift into decision-sensitive and decision-null components and regularizes only the former.**