

Exemplar-Free Continual Learning for State Space Models

Isaac Ning Lee^{1,2}

¹Monash University

Leila Mahmoodi¹

²National University of Singapore

Trung Le¹

Mehrtash Harandi¹

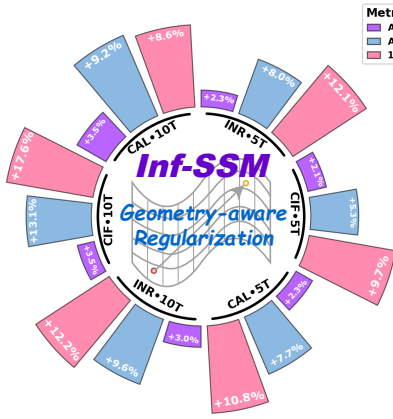
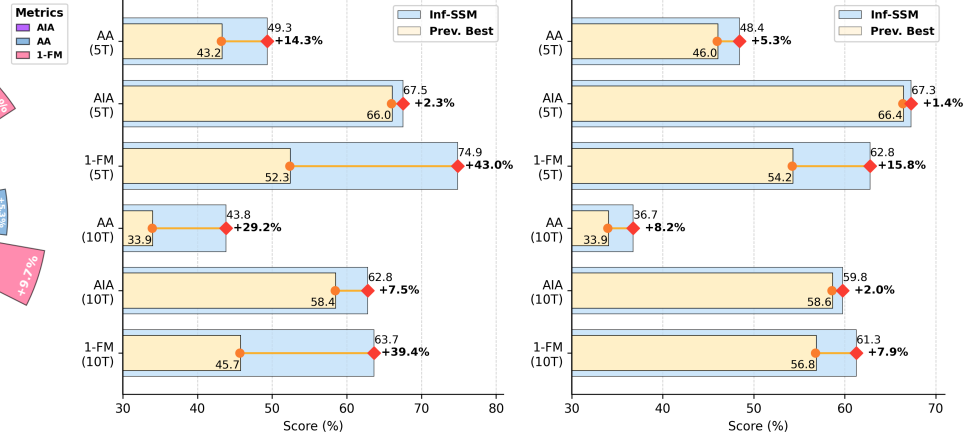


Figure 1. **Left:** Average performance gain from integrating Inf-SSM into representative CIL baselines (ER [51], LUCIR [24], X-DER [5], L2P-R [59], CLFD [37]) on 5-task (5T) and 10-task (10T) benchmarks built from ImageNet-R (INR) [23], CIFAR-100 (CIF) [31], and Caltech-256 (CAL) [17]. **Center:** Comparison of Inf-SSM with exemplar-free CIL methods (EWC [29], SI [61], MAS [3], LwF [33]) when regularizing (A, C) on INR. **Right:** Same comparison on INR, CIF, and CAL when baselines are extended to regularize (A, B, C), while Inf-SSM regularizes only (A, C). Metrics: AA (↑) = final accuracy after the last task; AIA (↑) = average incremental accuracy over all tasks; FM (↓) = forgetting measure.

Abstract

State-Space Models (SSMs) excel at capturing long-range dependencies with structured recurrence, making them well-suited for sequence modeling. However, their evolving internal states pose unique challenges in Continual Learning (CL). Without access to the full distribution of previous tasks, updates to the state-space dynamics become unconstrained, leading to catastrophic forgetting. To address this, we propose **Inf-SSM**, a geometry-aware regularization framework for CL in SSMs. It constrains state evolution via the infinite-dimensional Grassmannian of SSM observability subspaces, without requiring any exemplars from past tasks. Unlike classical CL methods that restrict weight updates, Inf-SSM directly regularizes the infinite-horizon state evolution encoded by the extended observability subspace of the SSM. We show that enforcing this regularization requires solving a matrix equation known as the Sylvester equation, which typically incurs $\mathcal{O}(n^3)$ complexity. Thus, we develop an $\mathcal{O}(n^2)$ solution by exploiting the structure and properties of SSMs. This leads to an efficient regularization mechanism that can be seamlessly integrated into existing CL methods. Comprehensive experiments on

challenging benchmarks of ImageNet-R, CIFAR-100, and Caltech-256 demonstrate a significant reduction in forgetting while improving accuracy across sequential tasks.

1. Introduction

In this paper, we propose **Inf-SSM**, a novel regularization method to equip State-Space Models (SSMs) [20] with Continual Learning (CL) capabilities, enabling them to integrate new information while preserving previously learned knowledge without any past exemplars.

State-space models (SSMs) [20] have emerged as structured sequence models that offer a scalable and efficient alternative to transformers. Recent architectures such as S4, S6, and Mamba-2 [8, 18, 20] effectively capture long-range dependencies with linear computational complexity, achieving strong performance across NLP, Vision, Generative Models, Acoustics, and Robotics [18, 34, 36, 48, 62, 65]. In NLP, Mamba [18] has positioned SSMs as competitive alternatives to transformers. In vision, Vision Mamba (Vim) [65] and its variants [22, 38] have demonstrated strong performance in spatial-temporal modeling while of-

fering faster inference than attention-based architectures.

Beyond their performance, SSMs provide intrinsic advantages such as structured recurrence and linear memory complexity, making them well-suited for sequence modeling. However, **a major gap remains**: adapting SSMs to continual learning while mitigating Catastrophic Forgetting (CF) [44, 45] remains largely **unexplored** [58]. CF occurs when a model loses previously learned knowledge as it is trained on new tasks. Despite their promising capabilities, the structured recurrence of SSMs makes continual and lifelong adaptation non-trivial, demanding specialized techniques to avoid CF [18, 65].

While well-established CL algorithms designed for MLPs, CNNs, and Transformers can be directly adapted to SSMs and yield competitive baselines, such adaptations typically treat SSM parameters as generic weights and overlook the underlying state-space geometry and temporal dynamics. As a result, they fail to fully exploit the structural advantages of SSMs in continual learning. This motivates the development of CL methods that are derived from the true state-space representation and regularize the model in a way that is faithful to its geometry.

Our Contribution. A key property of SSMs, as we will show shortly, is their invariance in system representations [1, 49]. This allows us to describe an SSM model using its extended observability subspace [9, 49]. The extended observability subspace provides a structured representation of an SSM, capturing its complete behavior through an infinite unrolling of the system’s response.

To work with extended observability subspaces, we adopt the geometry of the Grassmann manifold [26], which is the natural choice for analyzing subspaces. Such modeling captures the system representations’ invariance by identifying all orthonormal bases spanning the same subspace as a single point, and endows the space with a smooth manifold structure [1, 9, 54]. However, since the extended observability subspace belongs to an infinite-dimensional Grassmannian, direct distance computation is far from trivial. We address this challenge by demonstrating that, for SSMs, computing distances on the infinite Grassmannian is tractable, significantly reducing the computational complexity of our algorithm. Solving the computation challenges allows us to leverage the extended observability subspace to retain the model’s past knowledge effectively **without the need for storing past examples**. This enables Inf-SSM to be complementary to existing CL techniques, regardless of the availability of past task data or stored feature embeddings.

As a preview, Inf-SSM substantially boosts the performance of existing CL methods when integrated with them and consistently outperforms memory-less baselines in both accuracy and efficiency, as illustrated in Fig. 1.

In summary:

- We proposed Inf-SSM, a simple state regularization loss for CL in SSMs based on the extended observability subspace. Inf-SSM is an exemplar-free method and does not rely on stored information from prior tasks to function.
- We demonstrate that Inf-SSM is computationally efficient by exploiting structural properties in SSMs’ states to reduce the computational complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ and reduce the FLOPS count up to $100\times$. The simple and exemplar-free design of Inf-SSM makes it plug-and-play to improve existing CL algorithms, regardless of whether they are replay or replay-free methods.
- We empirically validate Inf-SSM across diverse CL paradigms, where it reduces FM by **9.36%** and increases AA by **8.31%** on average across the 5 and 10 task settings on three datasets when integrated with existing CL baselines.

2. Preliminary

In this section, we will first introduce the notation used throughout the paper. Then, we will formalize the Class-incremental Learning (CIL) and Exemplar-free Class-incremental Learning (EFCIL) problem in §2.1. Finally, we will review the theory of Grassmannian and SSM in §2.2 and §2.3, which underpin our proposed method.

Notations. In this paper, matrices are denoted as bold capital letters $\mathbf{X}$ and column vectors as bold lower-case letters $\mathbf{x}$. The $n \times n$ identity matrix is shown by $\mathbf{I}_n$. The diagonal elements of $\mathbf{X} \in \mathbb{R}^{n \times n}$ is shown by $\mathbf{X}_{diag}$. The *Frobenius norm* of $\mathbf{A} \in \mathbb{R}^{m \times n}$ is given by $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n \mathbf{A}[i, j]^2}$ and the *Hadamard product* of two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ is denoted $\mathbf{A} \odot \mathbf{B}$ and defined entrywise by $(\mathbf{A} \odot \mathbf{B})_{ij} = \mathbf{A}_{ij} \mathbf{B}_{ij}$. We use $\mathbf{x}[\cdot]$ to denote discrete-time vectors and $\mathbf{x}(\cdot)$ for continuous-time vectors. We write $\text{SN}(x) = 2/(1 + \exp(-x)) - 1$ as the Soft-normalization function [27]. The general linear group $\text{GL}(n)$ is the set of all $n \times n$ invertible matrices and is formally defined as: $\text{GL}(n) = \{\mathbf{P} \in \mathbb{R}^{n \times n} \mid \det(\mathbf{P}) \neq 0\}$. See §A for additional notation.

2.1. Problem Statement

Let us define a non-stationary task sequence $\{\mathcal{T}_T\}_{T=1}^N$, where T denotes the task identifier, and N is the total number of tasks. The data distribution of task T is denoted by $\mathcal{D}_T = \{\mathcal{X}_T, \mathcal{Y}_T\}$; where $\mathcal{X}_T$ and $\mathcal{Y}_T$ are input space and class labels, specific to task T . The number of classes in task T is shown by $\mathcal{C}_T$, and the label sets of every two tasks are disjoint, meaning $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset$ for $i \neq j$.

In the CIL protocol, the task identifier T is not available at inference time. The EFCIL protocol further tightens this setting: during training on task T , the model only has access

to the current data distribution $\mathcal{D}_T$, and no samples or feature embeddings from previous distributions $\{\mathcal{D}_i\}_{i=1}^{T-1}$ are allowed. This exemplar-free constraint makes EFCIL more challenging but also more flexible, as it removes the need to store data from past tasks.

2.2. Grassmann Manifold

The *Grassmann manifold* $\text{Gr}(n, d)$ is the set of all n -dimensional linear subspaces of $\mathbb{R}^d$. Formally, it is defined as

$$\text{Gr}(n, d) = \{\mathbf{X} \in \mathbb{R}^{d \times n} \mid \mathbf{X}^\top \mathbf{X} = \mathbf{I}_n\} / O(n), \quad (1)$$

where $O(n)$ denotes the orthogonal group of $n \times n$ matrices, accounting for the fact that different orthonormal bases can represent the same subspace. The **infinite Grassmannian** is defined as

$$\text{Gr}(n, \infty) = \lim_{d \rightarrow \infty} \text{Gr}(n, d), \quad (2)$$

which encapsulates all possible n -dimensional linear subspaces in an infinite-dimensional Hilbert space [60].

2.3. State-Space Models

Recent structured SSM architectures [8, 18–20, 65] are built upon the discretized state-space model¹:

$$\begin{aligned} \mathbf{h}[t] &= \mathbf{A}\mathbf{h}[t-1] + \mathbf{B}x[t], \\ y[t] &= \mathbf{C}\mathbf{h}[t], \end{aligned} \quad (3)$$

where $\mathbf{h}[t] \in \mathbb{R}^n$ is the hidden state, and $x[t], y[t] \in \mathbb{R}$ denote the input and output, respectively. The model is parameterized by the state transition matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, input matrix $\mathbf{B} \in \mathbb{R}^{n \times 1}$, and output matrix $\mathbf{C} \in \mathbb{R}^{1 \times n}$. Classical formulations assume $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ to be linear time-invariant (LTI), while more recent architectures such as Mamba [18], Mamba-2 [8], and Vim [65] introduce input- or position-dependent dynamics, effectively yielding time-varying state transitions. Such a dynamical nature allows a more adaptive model, but at the same time, poses a significant problem in regularizing the model's state.

Moreover, earlier SSMs, such as S4 [20] use a full $\mathbf{A} \in \mathbb{R}^{n \times n}$, whereas many recent models [18, 19, 21, 38, 65] exploit diagonal parameterizations of $\mathbf{A}$ to improve efficiency while preserving expressivity. Further details on SSMs, Mamba, Mamba-2, and Vim are provided in §D.

3. Proposed Method

Our aim is to endow SSMs and their variants with the ability to preserve their prior knowledge while gaining new. Consider a network trained on tasks $1:T-1$ with parameters ω_{T-1} . Continual learning on the incoming dataset $\mathcal{D}_T$

¹Details on discretization are provided in §B.1.

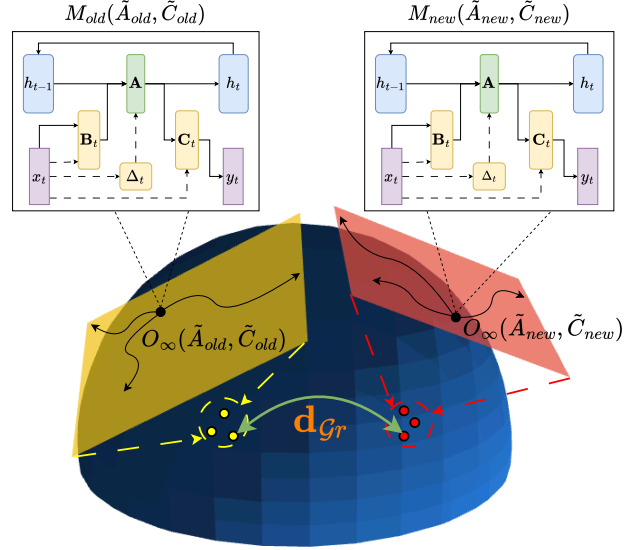


Figure 2. SSM at each sequence position τ is characterized by the infinite-horizon observability subspace O_∞ defined by the tuple $(\tilde{\mathbf{A}}, \tilde{\mathbf{C}})$, and visualized as a trajectory to an infinite horizon. The colored plane represents the complete set of O_∞ . Each trajectory is mapped to a point on the Grassmannian, and the pairwise distance d_{Gr} is illustrated as the geodesic on the sphere representing the Grassmann manifold.

typically follows two strategies: **(1)** Regularize the parameters for the incoming new data $\mathcal{D}_T$, making sure that the model does not significantly diverge from its initial parameters ω_{T-1} ; **(2)** Regularize the output behavior of the model to ensure the output space of the new model encapsulates the previous tasks' knowledge.

Translating such an idea to an SSM implies regularizing $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ or directly constraining the output y_t of the model. We discuss both possibilities in more detail below.

Regularizing the parameters. One can regularize the parameters of the SSM, *i.e.* $(\mathbf{A}, \mathbf{B}, \mathbf{C})$, leading to

$$\begin{aligned} L(\mathbf{A}_T, \mathbf{B}_T, \mathbf{C}_T) &= \|\mathbf{A}_T - \mathbf{A}_{T-1}\|_F^2 + \|\mathbf{B}_T - \mathbf{B}_{T-1}\|_F^2 \\ &\quad + \|\mathbf{C}_T - \mathbf{C}_{T-1}\|_F^2. \end{aligned} \quad (4)$$

The problem here is far more significant. The normal regularization of parameters, while implicitly addressing the finite horizon problem, is oblivious to the geometry of SSMs and totally ignores the P-equivalence.

Definition 3.1 (P-equivalence). Let an SSM be represented by the tuple of system parameters $(\mathbf{A}, \mathbf{B}, \mathbf{C})$. For an invertible matrix $\mathbf{P} \in \text{GL}(n)$, two SSMs with system parameters $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}, \mathbf{P}\mathbf{B}, \mathbf{C}\mathbf{P}^{-1})$ are said to be **P-equivalent** since they represent the same system behavior. That is, the transformation: $(\mathbf{A}' = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}, \mathbf{B}' =$

$\mathbf{PB}, \mathbf{C}' = \mathbf{CP}^{-1}$) preserves the input-output mapping (i.e., the two systems are indistinguishable in terms of their external behavior).

This has a significant implication as one can move along the orbit defined by the P-equivalence without affecting the system behavior. In other words, there are infinitely many parameter realizations for an SSM that lead to exactly the same sample path $y[t]$. As such and to regularize an SSM, the loss needs to be invariant to $(\mathbf{A}, \mathbf{B}, \mathbf{C}) \rightarrow (\mathbf{PAP}^{-1}, \mathbf{PB}, \mathbf{CP}^{-1})$ for any $\mathbf{P} \in \text{GL}(n)$, as illustrated in Fig. 3. We will show that our algorithm leads to a loss that is faithful to P-equivalence.

Regularizing the output. Alternatively, one can also regularize the output behavior of the model. This principle is the basis of the replay-based methods, which maintain a memory $\mathcal{M}$ containing representative samples from past experience. The objective typically enforces consistency between the current model and its previous version over these stored samples:

$$\mathcal{L}(\mathbf{A}_T, \mathbf{B}_T, \mathbf{C}_T) = \mathbb{E}_{x \sim \mathcal{M}} \sum_{t=1}^{\tau} \|y_T[t] - y_{T-1}[t]\|^2, \quad (5)$$

with τ denoting the time horizon. Ideally, one would take $\tau \rightarrow \infty$ so that the entire temporal behavior of the model is preserved as, in principle, SSMs can induce an output with an infinite sequence even with a finite input.

As in the EFCIL setting, storing $\mathcal{M}$ is not feasible, thus, we view the model as a dynamical system and consider its response when excited by random Gaussian input. Such excitation statistically probes the system and characterizes its behavior without requiring any stored samples. We will show soon that this idea allows our algorithm to operate effectively in the EFCIL regime, achieving the benefits of replay without memory while capturing the model's infinite-horizon behavior.

Our method. Although vanilla regularization (e.g., Equation (4)) improves CL performance which we will show in §5.1, SSMs still suffer from significant forgetting. We argue that a key reason is that standard penalties **ignore the geometry of SSMs**. In particular, moving along the orbit

$$\pi = \{(\mathbf{PAP}^{-1}, \mathbf{PB}, \mathbf{CP}^{-1}), \forall \mathbf{P} \in \text{GL}(n)\}, \quad (6)$$

does not change the input-output behavior of the SSM, yet conventional regularizers are not invariant to this transformation. As a result, they may over-penalize parameter updates that are functionally equivalent and under-penalize updates that alter the underlying dynamics.

What we propose to do instead is to take into account the trajectory defined by an SSM and use that to preserve the

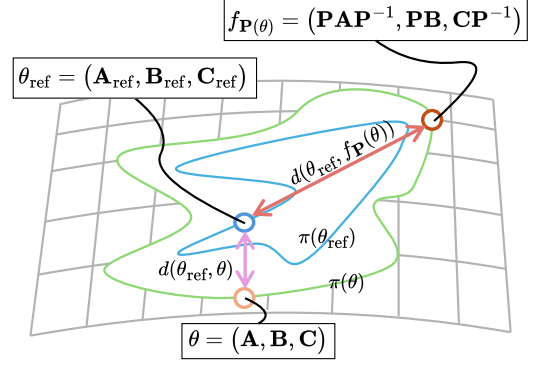


Figure 3. **Green:** Orbit $\pi(\theta)$ of the target SSM $\theta = (\mathbf{A}, \mathbf{B}, \mathbf{C})$; **Blue:** Orbit $\pi(\theta_{\text{ref}})$ of the reference. A state reparameterization $x \mapsto \mathbf{P}x$ moves θ to $f_{\mathbf{P}}(\theta) = (\mathbf{PAP}^{-1}, \mathbf{PB}, \mathbf{CP}^{-1})$, preserving SSM's behavior but changing the Frobenius distance to the reference. Thus, the Frobenius norm is not invariant to P-equivalence.

knowledge. This perspective is naturally compatible with arbitrarily long horizons as a byproduct of our construction. We will discuss this below, but first, we need to take a detour and introduce the Extended observability matrix of an SSM.

3.1. Extended Observability

Consider studying the behavior for an SSM excited by $x[t] \sim \mathcal{N}(0, 1)$. The expected response of the system can be derived as

$$\begin{aligned} \mathbb{E}[y[t]] &= \mathbb{E}[\mathbf{C}\mathbf{h}[t]] = \mathbb{E}[\mathbf{C}(\mathbf{A}\mathbf{h}[t-1] + \mathbf{B}x[t])] \\ &= \mathbf{CA}\mathbb{E}[\mathbf{h}[t-1]]. \end{aligned} \quad (7)$$

This implies,

$$\begin{bmatrix} \mathbb{E}[y[1]] \\ \mathbb{E}[y[2]] \\ \mathbb{E}[y[3]] \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \end{bmatrix} \mathbf{h}[0] \quad (8)$$

In other words, the dynamics of an SSM can be well-captured by the pair $(\mathbf{A}, \mathbf{C})$. The extended observability matrix is defined accordingly.

Definition 3.2 (Extended Observability). *The extended observability of an SSM with parameters $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is defined as:*

$$\mathbf{O}_{\infty}(\mathbf{A}, \mathbf{C}) \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \end{bmatrix} \in \mathbb{R}^{\infty \times n}. \quad (9)$$

One important emerging property of the observability matrix for an SSM is that the subspace spanned by the

columns of $\mathbf{O}_\infty(\mathbf{A}, \mathbf{C})$ is invariant to the orbit defined by Eq. (6). More specifically, let $\mathcal{S}_\infty(\mathbf{A}, \mathbf{C})$ be the subspace spanned by the columns of $\mathbf{O}_\infty(\mathbf{A}, \mathbf{C})$. Then we have the following essential result.

Theorem 3.3 (Invariance of the $\mathcal{S}_\infty$ under P-equivalence). *Let $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $(\mathbf{A}' = \mathbf{PAP}^{-1}, \mathbf{B}' = \mathbf{PB}, \mathbf{C}' = \mathbf{CP}^{-1})$ be two equivalent representations for an SSM for $\mathbf{P} \in \text{GL}(n)$. The subspace spanned by the extended observability matrices remains unchanged. That is*

$$\mathcal{S}_\infty(\mathbf{A}', \mathbf{C}') = \mathcal{S}_\infty(\mathbf{A}, \mathbf{C}). \quad (10)$$

This is a well-known result in the context of system identification in linear dynamical systems (see [27]). We also provide a proof in Theorem C.2 in §C.2 to have a self-contained work.

We note that $\mathcal{S}_\infty(\mathbf{A}, \mathbf{C}) \in \text{Gr}(n, \infty)$. Although with $\mathcal{S}_\infty(\mathbf{A}, \mathbf{C})$, we can describe the expected output of an SSM uniquely, to use it for regularizing an SSM and deploying it for CL, we need one more step. That is, given two subspaces on $\text{Gr}(n, \infty)$, how can their distance/similarity be measured?

3.2. Distance on infinite Grassmannian

For $\mathcal{S}, \mathcal{S}' \in \text{Gr}(n, d)$, the chordal distance [60] is defined as

$$d_{\text{chord}}^2(\mathcal{S}, \mathcal{S}') = \|\mathcal{S}\mathcal{S}^\top - \mathcal{S}'\mathcal{S}'^\top\|_F^2 = 2n - 2\|\mathcal{S}^\top \mathcal{S}'\|_F^2 \quad (11)$$

As $d \rightarrow \infty$ for the subspace spanned by the extended observability matrix, it is challenging to compute d_{chord}^2 as $\|\mathcal{S}^\top \mathcal{S}'\|_F^2$ cannot be computed explicitly. Let $\mathbb{R}^{m \times n} \ni \mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_n]$ be a full rank matrix. The n -dimensional subspace spanned by the columns of $\mathbf{O}$ can be written as

$$\mathcal{S} = \mathbf{O}(\mathbf{O}^\top \mathbf{O})^{-1/2}, \quad (12)$$

as shown in §C.3. Hence, $d_{\text{chord}}^2(\mathcal{S}, \mathcal{S}')$ can be written as:

$$d_{\text{chord}}^2(\mathcal{S}, \mathcal{S}') = 2n - 2\left\|(\mathbf{O}^\top \mathbf{O})^{-1/2} \mathbf{O}^\top \mathbf{O}' (\mathbf{O}'^\top \mathbf{O}')^{-1/2}\right\|_F^2$$

As such, one needs to be able to compute terms such as $\mathbf{G}_1 = (\mathbf{O}^\top \mathbf{O})$, $\mathbf{G}_2 = (\mathbf{O}'^\top \mathbf{O}')$, and $\mathbf{G}_3 = (\mathbf{O}^\top \mathbf{O}')$ for observability matrices. The lemma below shows how this is done.

Lemma 3.4. *Let $\mathbf{A}, \mathbf{A}' \in \mathbb{R}^{n \times n}$ and $\mathbf{C}, \mathbf{C}' \in \mathbb{R}^{1 \times n}$ be the parameters of two SSMs with the extended observability matrices*

$$\mathbf{O}_\infty(\mathbf{A}, \mathbf{C}) = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA}^2 \\ \vdots \end{bmatrix}, \quad \mathbf{O}_\infty(\mathbf{A}', \mathbf{C}') = \begin{bmatrix} \mathbf{C}' \\ \mathbf{C}'\mathbf{A}' \\ \mathbf{C}'(\mathbf{A}')^2 \\ \vdots \end{bmatrix}.$$

Define the Gram matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$ as:

$$\mathbf{G} = \mathbf{O}_\infty(\mathbf{A}, \mathbf{C})^\top \mathbf{O}_\infty(\mathbf{A}', \mathbf{C}') = \sum_{t=0}^{\infty} (\mathbf{A}^\top)^t \mathbf{C}^\top \mathbf{C}' (\mathbf{A}')^t.$$

Then $\mathbf{G}$ can be obtained by solving the following equation

$$\mathbf{A}^\top \mathbf{G} \mathbf{A}' - \mathbf{G} = -\mathbf{C}^\top \mathbf{C}'. \quad (13)$$

The form $\mathbf{A}^\top \mathbf{G} \mathbf{A}' = \mathbf{G} - \mathbf{C}^\top \mathbf{C}'$ is an instance of the Sylvester problem and can be solved, for example, with the Bartels-Stewart algorithm [4]. However, the computational complexity of solving the Sylvester algorithm is $\mathcal{O}(n^3)$, which is computationally expensive as n grows larger. As pointed out in §2.3, for structured SSMs (e.g., when $\mathbf{A}$ is diagonal), we take advantage of the structure to reduce the computational complexity of $\mathbf{G}$.

Lemma 3.5. *Define $\mathbf{A}_{\text{diag}}, \mathbf{A}'_{\text{diag}} \in \mathbb{R}^{n \times 1}$, as diagonal elements of $\mathbf{A}, \mathbf{A}'$. The Sylvester equation in Equation (13) can be written as:*

$$\mathbf{G} \odot (\mathbf{1}_n - \mathbf{A}_{\text{diag}} \mathbf{A}'_{\text{diag}}^\top) = \mathbf{C}^\top \mathbf{C}'$$

Hence, $\mathbf{G}$ can be computed as:

$$\mathbf{G} = \mathbf{C}^\top \mathbf{C}' \odot \frac{1}{\mathbf{1}_n - \mathbf{A}_{\text{diag}} \mathbf{A}'_{\text{diag}}^\top} = \mathbf{O}^\top \mathbf{O}' \quad (14)$$

This **reduces the computational complexity** of solving the Sylvester problem from $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2)$ and reduces the FLOPS count from $25n^3$ of the Bartels-Stewart algorithm [14] to $4n^2$. In the case of $n = 16$ in Vim [65], we **reduce the FLOPS count by 100×**.

Putting everything together, to compute $d_{\text{chord}}^2(\mathcal{S}, \mathcal{S}')$, we solve three Sylvester equations in the form Eq. (14) to obtain $\mathbf{G}_1, \mathbf{G}_2$, and $\mathbf{G}_3$, followed by performing matrix algebra on the resulting $\mathbf{G}$ matrices based on the form of distance. This enables us to define a loss based on the distances of the extended observability matrices for the model of task $T - 1$ and the model under training at task T .

3.3. Extension to S6

In S4D [19], and Mamba [18], the dimensionality of each states are: $\mathbf{A} \in \mathbb{R}^{\tau \times o \times n}$, and $\mathbf{C} \in \mathbb{R}^{\tau \times n}$, where τ denotes sequence length and o denotes outer dimension (or channels) of SSMs. Each sequence τ consists of o LDS. Thus, the state transitions, input, and output mappings are time-variant.

To benefit from the geometry of the extended observability, we propose to generate a set of extended observability matrices, $\mathbb{O} = \{(\mathbf{O}_{\infty, t}(\tilde{\mathbf{A}}_t, \tilde{\mathbf{C}}_t))\}_{t=1}^\tau$, where:

$$\tilde{\mathbf{A}} = \text{SN}\left(\frac{1}{o} \sum_{i=1}^o \bar{\mathbf{A}}_{i,j}\right) \in \mathbb{R}^{\tau \times n}$$

$$\tilde{\mathbf{C}} = \text{SN}(\mathbf{C}) \in \mathbb{R}^{\tau \times n}$$

and Soft-Normalization $\text{SN}(x) = 2/(1 + \exp(-x)) - 1$ is applied to enforce Schur Stability [27]. The states are averaged along the outer dimension, as computing distance for $\tau \times o$ pairs of SSMs is computationally prohibitive. For example, in Vim-small, this will involve $197 \times 384 \approx 7.6 \times 10^4$ pairs of SSMs, exceeding the VRAM of a single H100 GPU. Additionally, averaging along o also preserves maximal variance (more information in §D.3). Subsequently, we use the set $\mathbb{O}$ to regularize the model in CL.

3.4. Inf-SSM

To utilize Eq. (11) in continual learning, we propose a combination of state distillation and regularization as illustrated in Fig. 2. For the set of extended observability matrices for tasks $T-1$ and T as $\mathbb{O}_{T-1}$ and $\mathbb{O}_T$ respectively, we define the loss of Inf-SSM as

$$\mathcal{L}_{\text{ISM}} = \mathbb{E}_{\mathcal{D}_T} \left\{ d_{\text{chord}}^2(\mathbb{O}_{T-1}, \mathbb{O}_T) \right\}. \quad (15)$$

Hence, by combining with the classification loss, the total CL loss is:

$$\mathcal{L}_{\text{tot}} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{ISM}} \quad (16)$$

where λ is the regularization strength. $\mathcal{L}_{\text{cls}}$ is the classification loss for the base model. The full Inf-SSM algorithm is included in Algorithm 1 under §E. The minimalist design of Inf-SSM (only one additional hyperparameter λ), in contrast to some recent SOTA CL methods that may require more than 5 additional hyperparameters, makes it substantially easier to tune and deploy.

4. Related Work

Continual Learning. Continual learning aims to balance *stability*² and *plasticity*³ [6, 47]. We focus on the CIL setting [25, 55], where task identities are typically unknown at test time as described in §2.1.

A central axis in CIL is whether examples from previous tasks can be stored. (i) In EFCIL, retaining past samples is prohibited due to privacy, security, or storage constraints [46, 52]. In this regime, many methods rely on regularization: EWC [29], SI [61], and MAS [3] estimate parameter importance and penalize changes to critical weights. Another line of work distills knowledge from previous models, using logits [33], embeddings [11], or features [28, 53]. (ii) When storing or reconstructing examples is allowed, *replay-based* methods interleave past and new data to mitigate forgetting, with ER [51] being a prominent example. Methods that combine EFCIL-style techniques with replay are often referred to as hybrid approaches, including LUCIR [24] and X-DER [5]. More recent hybrids, such as L2P [59], leverage prompt learning, while

CLFD [37] benefits from frequency domain features to further enhance CL performance. These methods are particularly effective when memory and privacy constraints permit replay. We refer to §F.1 for a more extensive discussion on recent CL methods.

Vision SSM. SSMs have recently attracted considerable attention due to their theoretical ability to model infinitely long sequences, in contrast to Transformers, which are limited by fixed token lengths [18, 34, 36, 48, 65]. In computer vision, Vim [65] extends SSM by applying a patch embedding to images and treating each patch token as a sequential element processed through bidirectional Mamba blocks. Subsequent works like VMamba [38] incorporate a 2D-selective scan to capture spatial dependencies better, while MambaVision [22] integrates SSM with Transformers to form a hybrid architecture.

Takeaways. While existing CL algorithms are generally compatible with SSMs, they treat SSMs as generic neural networks and ignore the rich underlying geometry of SSMs. Similarly, recent Mamba-based CL methods [7, 32, 63], although effective in their specific scenarios, generally rely on storing feature embeddings from past tasks and do not explicitly exploit the geometric properties of SSMs (see §F.2). Motivated by this gap, we study CL for SSM and, to the best of our knowledge, introduce the *first* CL method that explicitly exploits the geometry of SSMs. Our approach is complementary to existing CL algorithms and can be used in both exemplar-free and replay-based scenarios.

5. Experiments

In this section, we first integrate Inf-SSM into CIL methods to prove its adaptability. Next, we evaluate Inf-SSM under conditions where $(\mathbf{A}, \mathbf{C})$ and $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ are regularized, comparing its performance against existing foundational EFCIL methods across three different datasets.

Baselines. To demonstrate the versatility of Inf-SSM, we integrate it with a diverse set of CIL methods spanning different paradigms. Specifically, we consider replay-based method: ER [51], hybrid methods: LUCIR [24] and X-DER [5], prompt-based method: L2P-R [59], and a frequency-based method: CLFD [37] as shown in Tab. 1. To isolate and highlight the benefits of Inf-SSM’s geometry-aware regularization, in Tab. 2 and Tab. 6, we further compare Inf-SSM independently against flagship EFCIL methods of EWC [29], SI [61], and MAS [3] from the regularization category and LwF [33] from the distillation category. We *specifically choose* these methods to comprehensively evaluate Inf-SSM while avoiding baselines that are misaligned with our goal (more details in §J.1).

Datasets. We follow previous CL studies [35, 39, 41, 42] and use ImageNet-R [23], CIFAR-100 [31], and Caltech-256 [17]. For all datasets, we split the classes equally into 5

²**Stability:** the ability to retain past knowledge.

³**Plasticity:** the ability to acquire new knowledge.

Table 1. AA(% $\uparrow$), AIA(% $\uparrow$), and FM(% $\downarrow$) of Vim-small are reported for ImageNet-R, CIFAR-100, and Caltech-256 datasets over 5 Tasks and 10 tasks benchmarks for existing CL methods and integration with Inf-SSM. **Note:** Due to differences in batch size and epochs, the performance of different baselines cannot be directly compared.

Method	ImageNet-R			CIFAR-100			Caltech-256		
	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std
5-Tasks Scenario									
ER [51]	30.91 $\pm$ 1.53	56.84 $\pm$ 1.21	64.04 $\pm$ 1.01	33.00 $\pm$ 1.12	59.66 $\pm$ 0.79	77.05 $\pm$ 1.37	49.94 $\pm$ 1.54	70.95 $\pm$ 0.98	55.61 $\pm$ 1.60
+Inf-SSM	31.26 $\pm$ 2.65	59.69 $\pm$ 2.02	59.45 $\pm$ 3.12	33.78 $\pm$ 0.69	60.32 $\pm$ 1.12	76.43 $\pm$ 0.84	51.39 $\pm$ 0.67	71.91 $\pm$ 0.26	53.23 $\pm$ 0.75
LUCIR [24]	31.18 $\pm$ 1.49	60.97 $\pm$ 0.94	60.24 $\pm$ 1.93	32.77 $\pm$ 0.26	62.14 $\pm$ 0.54	76.03 $\pm$ 0.42	64.79 $\pm$ 0.22	80.72 $\pm$ 0.50	26.79 $\pm$ 0.64
+Inf-SSM	35.35 $\pm$ 0.89	59.53 $\pm$ 0.85	52.05 $\pm$ 0.51	33.10 $\pm$ 1.06	62.58 $\pm$ 1.80	74.67 $\pm$ 1.65	68.63 $\pm$ 0.37	80.91 $\pm$ 0.66	19.13 $\pm$ 0.53
X-DER [5]	47.42 $\pm$ 1.45	66.01 $\pm$ 0.99	42.99 $\pm$ 1.62	42.33 $\pm$ 0.54	67.41 $\pm$ 0.84	65.07 $\pm$ 0.49	58.51 $\pm$ 0.06	75.71 $\pm$ 0.49	44.84 $\pm$ 0.35
+Inf-SSM	52.61 $\pm$ 3.01	69.40 $\pm$ 1.58	31.00 $\pm$ 1.90	48.33 $\pm$ 0.70	70.14 $\pm$ 0.39	56.66 $\pm$ 0.85	68.04 $\pm$ 1.48	80.99 $\pm$ 0.71	32.11 $\pm$ 1.67
L2P-R [59]	31.57 $\pm$ 0.34	56.65 $\pm$ 0.54	60.45 $\pm$ 0.75	35.86 $\pm$ 0.96	62.96 $\pm$ 0.18	73.03 $\pm$ 1.52	31.14 $\pm$ 1.49	58.24 $\pm$ 0.51	78.66 $\pm$ 1.76
+Inf-SSM	33.60 $\pm$ 0.54	57.15 $\pm$ 0.81	57.63 $\pm$ 0.43	36.90 $\pm$ 0.31	64.41 $\pm$ 0.50	71.71 $\pm$ 0.48	34.09 $\pm$ 1.17	58.87 $\pm$ 0.41	74.77 $\pm$ 1.57
CLFD [37]	35.36 $\pm$ 1.07	51.80 $\pm$ 1.13	30.13 $\pm$ 0.41	33.01 $\pm$ 1.35	57.10 $\pm$ 1.96	69.71 $\pm$ 0.82	48.86 $\pm$ 1.15	64.78 $\pm$ 1.34	33.63 $\pm$ 0.30
+Inf-SSM	37.68 $\pm$ 2.71	53.28 $\pm$ 2.15	28.45 $\pm$ 0.48	34.20 $\pm$ 2.37	58.31 $\pm$ 2.26	67.96 $\pm$ 1.77	50.46 $\pm$ 2.92	65.91 $\pm$ 1.57	32.19 $\pm$ 1.78
10-Tasks Scenario									
ER [51]	21.21 $\pm$ 1.01	50.40 $\pm$ 0.78	70.24 $\pm$ 0.97	28.03 $\pm$ 0.66	57.05 $\pm$ 1.25	76.44 $\pm$ 0.84	42.78 $\pm$ 1.30	66.88 $\pm$ 0.06	57.95 $\pm$ 1.37
+Inf-SSM	22.10 $\pm$ 1.94	52.47 $\pm$ 1.12	64.20 $\pm$ 2.47	28.75 $\pm$ 0.93	57.22 $\pm$ 1.57	75.78 $\pm$ 1.17	43.71 $\pm$ 1.08	67.74 $\pm$ 0.22	56.71 $\pm$ 1.17
LUCIR [24]	23.63 $\pm$ 1.67	56.84 $\pm$ 1.28	50.36 $\pm$ 1.02	24.04 $\pm$ 0.96	53.27 $\pm$ 0.73	80.98 $\pm$ 1.20	52.24 $\pm$ 1.78	72.88 $\pm$ 0.75	31.45 $\pm$ 1.09
+Inf-SSM	28.60 $\pm$ 0.44	57.37 $\pm$ 0.63	42.96 $\pm$ 1.12	25.72 $\pm$ 0.46	56.33 $\pm$ 0.53	78.75 $\pm$ 0.69	55.44 $\pm$ 1.75	74.96 $\pm$ 0.51	28.49 $\pm$ 1.98
X-DER [5]	39.98 $\pm$ 0.32	61.41 $\pm$ 0.82	47.56 $\pm$ 0.36	34.01 $\pm$ 0.38	62.62 $\pm$ 0.54	69.53 $\pm$ 0.29	48.30 $\pm$ 0.77	70.77 $\pm$ 0.29	51.57 $\pm$ 0.82
+Inf-SSM	44.09 $\pm$ 0.64	63.57 $\pm$ 0.74	38.72 $\pm$ 0.74	47.67 $\pm$ 0.80	68.08 $\pm$ 0.22	53.01 $\pm$ 0.83	60.76 $\pm$ 1.48	77.32 $\pm$ 0.09	37.42 $\pm$ 1.39
L2P-R [59]	23.08 $\pm$ 1.04	50.69 $\pm$ 0.36	60.45 $\pm$ 1.50	22.29 $\pm$ 0.41	52.49 $\pm$ 0.30	82.70 $\pm$ 0.52	24.13 $\pm$ 1.09	51.99 $\pm$ 1.37	78.89 $\pm$ 1.19
+Inf-SSM	24.91 $\pm$ 1.51	51.75 $\pm$ 0.54	54.25 $\pm$ 1.63	23.64 $\pm$ 1.70	52.68 $\pm$ 0.39	81.22 $\pm$ 1.94	25.65 $\pm$ 1.68	52.88 $\pm$ 0.34	77.11 $\pm$ 1.81
CLFD [37]	28.90 $\pm$ 0.95	46.72 $\pm$ 0.90	31.62 $\pm$ 1.83	29.18 $\pm$ 1.36	54.67 $\pm$ 1.75	62.63 $\pm$ 1.74	42.12 $\pm$ 0.99	61.57 $\pm$ 0.50	31.09 $\pm$ 0.25
+Inf-SSM	30.25 $\pm$ 1.29	48.78 $\pm$ 1.43	30.90 $\pm$ 0.97	29.72 $\pm$ 1.27	55.50 $\pm$ 2.38	61.08 $\pm$ 0.56	43.26 $\pm$ 0.78	62.52 $\pm$ 0.86	29.73 $\pm$ 0.97

Table 2. AA(% $\uparrow$), AIA(% $\uparrow$), and FM(% $\downarrow$) of EFCIL methods on ImageNet-R, CIFAR-100, and Caltech-256 over 5-Tasks and 10-Tasks scenario on Vim-small. **Note:** Regularization focus is on parameter sets (A, B, C) among all methods **except** Inf-SSM. Second-best results are underlined.

Method	ImageNet-R			CIFAR-100			Caltech-256		
	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std
5-Tasks Scenario									
Seq	38.36 $\pm$ 4.47	61.29 $\pm$ 1.76	56.43 $\pm$ 5.14	36.68 $\pm$ 1.66	61.25 $\pm$ 1.41	55.00 $\pm$ 2.33	37.58 $\pm$ 1.08	60.17 $\pm$ 0.95	71.48 $\pm$ 1.52
EWC [29]	45.58 $\pm$ 2.72	65.62 $\pm$ 1.16	47.31 $\pm$ 3.18	38.25 $\pm$ 1.30	63.12 $\pm$ 1.10	50.71 $\pm$ 1.20	42.93 $\pm$ 1.64	64.27 $\pm$ 1.31	64.30 $\pm$ 2.24
SI [61]	<u>45.72</u> $\pm$ 3.04	65.18 $\pm$ 1.48	47.21 $\pm$ 3.34	37.38 $\pm$ 1.40	61.78 $\pm$ 1.05	53.04 $\pm$ 1.35	<u>47.57</u> $\pm$ 0.21	65.29 $\pm$ 1.04	<u>57.88</u> $\pm$ 0.43
MAS [3]	44.70 $\pm$ 2.77	65.59 $\pm$ 0.79	48.23 $\pm$ 2.92	37.59 $\pm$ 1.44	61.95 $\pm$ 1.18	53.13 $\pm$ 1.81	44.87 $\pm$ 1.19	66.44 $\pm$ 1.07	61.00 $\pm$ 1.53
LwF-ABC [33]	45.09 $\pm$ 6.58	<u>65.69</u> $\pm$ 3.17	<u>40.77</u> $\pm$ 8.26	<u>44.62</u> $\pm$ 2.67	<u>66.81</u> $\pm$ 1.41	<u>38.68</u> $\pm$ 3.77	46.52 $\pm$ 2.66	<u>66.58</u> $\pm$ 1.08	59.03 $\pm$ 3.43
Inf-SSM	49.34 $\pm$ 3.36	67.51 $\pm$ 1.47	25.14 $\pm$ 3.86	45.18 $\pm$ 2.28	67.34 $\pm$ 1.86	36.59 $\pm$ 3.33	50.75 $\pm$ 3.16	67.04 $\pm$ 1.43	49.93 $\pm$ 3.61
10-Tasks Scenario									
Method	ImageNet-R			CIFAR-100			Caltech-256		
	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std	AA $\pm$ std	AIA $\pm$ std	FM $\pm$ std
Seq	32.95 $\pm$ 1.71	55.36 $\pm$ 0.70	58.30 $\pm$ 2.19	20.58 $\pm$ 1.01	51.37 $\pm$ 0.35	71.49 $\pm$ 1.20	24.27 $\pm$ 1.29	51.06 $\pm$ 1.21	79.01 $\pm$ 1.36
EWC [29]	<u>41.99</u> $\pm$ 2.28	61.78 $\pm$ 2.24	49.02 $\pm$ 2.10	22.20 $\pm$ 1.11	53.46 $\pm$ 0.35	63.92 $\pm$ 1.28	28.35 $\pm$ 0.34	56.64 $\pm$ 0.74	72.73 $\pm$ 0.47
SI [61]	41.59 $\pm$ 1.17	61.13 $\pm$ 1.43	46.81 $\pm$ 1.00	20.29 $\pm$ 0.62	49.28 $\pm$ 1.68	38.33 $\pm$ 1.41	27.66 $\pm$ 1.00	54.34 $\pm$ 1.27	74.69 $\pm$ 0.85
MAS [3]	40.10 $\pm$ 1.48	61.30 $\pm$ 0.64	48.18 $\pm$ 1.84	20.44 $\pm$ 1.60	49.69 $\pm$ 1.55	37.99 $\pm$ 1.01	28.15 $\pm$ 0.79	55.25 $\pm$ 0.70	73.50 $\pm$ 0.75
LwF-ABC [33]	41.85 $\pm$ 0.82	<u>62.63</u> $\pm$ 0.92	<u>40.10</u> $\pm$ 0.61	<u>24.39</u> $\pm$ 3.25	<u>53.48</u> $\pm$ 1.49	<u>25.29</u> $\pm$ 2.81	<u>35.45</u> $\pm$ 1.16	<u>59.63</u> $\pm$ 0.74	<u>64.32</u> $\pm$ 1.41
Inf-SSM	43.82 $\pm$ 1.55	62.82 $\pm$ 1.29	36.34 $\pm$ 1.54	26.53 $\pm$ 3.10	54.24 $\pm$ 1.87	24.00 $\pm$ 1.80	39.88 $\pm$ 2.43	62.28 $\pm$ 2.33	55.85 $\pm$ 4.05

and 10 sequential tasks. More information in §G.1.

Evaluation Metrics. Three main evaluation metrics in CL are (1) Average Accuracy AA, (2) Average Incremental Accuracy AIA and (3) Forgetting Measure FM [58]. AA and AIA are mainly used for evaluation of overall performance while FM measures the stability of the model. Ideally, a high AA and AIA with a low FM are desirable. More details are included in §G.2.

Implementation details. The backbone architecture is Vim-Small [65] in **all experiments**. Detailed hyperparameters are included in §J.2.

5.1. Empirical Results and Analyses

Integration with existing CL methods. Tab. 1 shows that on average, integrating Inf-SSM improves the corresponding baselines AA by **8.31%** and reduces FM by **9.36%**. This underlines the generality of Inf-SSM, as it improves all metric instances, except for the marginal decrease in AIA for LUCIR on 5-task ImageNet-R. The benefits become more pronounced as the number of tasks increases, as reflected by the substantially larger improvements in AA and FM compared to AIA, indicating that Inf-SSM effectively captures the evolution of SSMs over longer task sequences. Notably, Inf-SSM remains effective when coupled with X-DER, the strongest baseline, where it improves average AA by **19.61%**, and reduces FM by **23.16%**.

Observability state parameter regularization. First, we regularize the state matrices ($\mathbf{A}$, $\mathbf{C}$) in all SSM blocks following each method’s strategy. For EWC [29], SI [61], and MAS [3], regularization is applied to the parameters that directly determine ($\mathbf{A}$, $\mathbf{C}$), while LwF [33] is implemented via distillation on ($\mathbf{A}$, $\mathbf{C}$) (denoted LwF-AC). As reported in Tab. 6 in §G.3, Inf-SSM achieves an average reduction in FM of **40.18%** and an average improvement in AA of **21.73%** over these baselines when regularizing ($\mathbf{A}$, $\mathbf{C}$).

Full State Parameter Regularization. To stress-test Inf-SSM, we extend EWC [29], SI [61], and MAS [3] to regularize all parameters involved in state-matrix formation and modify LwF [33] to distill on ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$) (denoted LwF-ABC), whereas Inf-SSM regularize ($\mathbf{A}$, $\mathbf{C}$) only, placing it in a deliberately disadvantaged setting. Even under this configuration, Tab. 2 shows that on average, Inf-SSM reduces FM by **14.56%** and improves AA by **6.79%**. These results highlight the strength of Inf-SSM in mitigating CF and empirically support our claim in Theorem 3.3, that regularizing the extended observability subspace is sufficient to control the evolution of the SSM’s behavior.

5.2. Additional studies

To complement our results in §5.1, we have conducted **five additional studies**: (1) As shown in Tab. 3, Inf-SSM is significantly faster than recent CL methods like X-DER [5]

while still on par with simple CL methods like EWC [29] without incurring any task-level overhead for computing the Fisher Information Matrix (FIM), which requires a forward-backward pass on the entire task dataset. (2) We applied CKD analysis in §H.1 and revealed that earlier blocks’ SSM states undergo fewer structural changes compared to deeper layers. (3) In §H.2, we applied Inf-SSM on different numbers of blocks in Vim-Small. Tab. 7 shows that increasing the number of Vim blocks regularized will improve the performance by preventing over-regularization in shallow layers. (4) §H.3 shows how our algorithm improves computational efficiency and stability. (5) Eq. (15) ignores an important aspect of SSM, which is the input mapping defined by $\mathbf{B}$. Thus, we add a Frobenius norm term to Inf-SSM to form Inf-SSM+ that regularizes $\mathbf{B}$. While Inf-SSM+ outperforms Inf-SSM in some scenarios, their performance is similar, as shown in Tab. 9 in §H.4, which agrees with Theorem 3.3.

Table 3. Latency comparison between Inf-SSM and simple (EWC [29]) and recent (X-DER [5]) CL methods with Vim-small on a single NVIDIA A40 GPU.

Operation	Mean (s)	Std. Dev (s)
EWC Loss (per batch)	0.0095	0.0124
EWC FIM (per task)	181.5038	28.8674
X-DER Loss (per batch)	1.2534	0.0766
Inf-SSM Loss (per batch)	0.0960	0.0384

6. Discussion and Conclusion

Limitations. Inf-SSM utilizes the fact that mainstream SSMs [18, 19, 38, 65] have $\mathbf{A}$ as a diagonal matrix. While a general $\mathbf{A}$ will prohibit the computational shortcut described in Eq. (14), Inf-SSM **remains valid** for general $\mathbf{A}$ as the observability Gramians can still be computed using the general Sylvester equation formulation described in Theorem C.4. Thus, the diagonal assumption is not essential for Inf-SSM, but a design choice of the mainstream SSMs’ architecture that enables faster computation.

Conclusion. We propose Inf-SSM to enable SSMs to learn from sequential tasks while preserving their prior knowledge without the need for previous task distributions. We constrain model updates by regularizing their extended observability subspace to retain the model’s knowledge while remaining faithful to the structure of SSMs. This is made possible by overcoming computational challenges for distances on the infinite Grassmannian. Beyond CL, Inf-SSM holds promise for broader applications. Future work may utilize Inf-SSM in knowledge distillation and model compression by integrating it with Schubert Varieties [60] to handle varying dimensionalities and kernel methods [27] within a Reproducing Kernel Hilbert Space for richer representations, potentially expanding its applicability.

Acknowledgments

Mehrtash Harandi and Trung Le are supported by the Australian Research Council (ARC) Discovery Program under grant DP230101176, “Exploiting Geometries of Learning for Fast, Adaptive and Robust AI.”

Isaac Ning Lee is supported by the NUS Research Scholarship from the School of Computing at the National University of Singapore. Leila Mahmoodi is supported by CSIRO’s Data61 Science Digital and the CSIRO Data61 Postgraduate Scholarship.

We thank the reviewers for their constructive feedback, which helped improve the paper.

References

- [1] Bijan Afsari and René Vidal. The alignment distance on spaces of linear dynamical systems. In *52nd IEEE Conference on Decision and Control*, pages 1162–1167, 2013. [2](#)
- [2] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. *Advances in neural information processing systems*, 32, 2019. [28](#)
- [3] 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. [1](#), [6](#), [7](#), [8](#), [23](#), [24](#), [28](#), [32](#)
- [4] Richard H. Bartels and G. W. Stewart. Solution of the matrix equation $AX + XB = C$. *Communications of the ACM*, 15(9): 820–826, 1972. [5](#), [18](#)
- [5] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. [1](#), [6](#), [7](#), [8](#), [32](#)
- [6] Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018. [6](#)
- [7] De Cheng, Yue Lu, Lingfeng He, Shizhou Zhang, Xi Yang, Nannan Wang, and Xinbo Gao. Mamba-CL: Optimizing selective state space model in null space for continual learning. *arXiv preprint arXiv:2411.15469*, 2024. [6](#), [23](#), [32](#)
- [8] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Forty-first International Conference on Machine Learning*, 2024. [1](#), [3](#), [19](#)
- [9] Katrien De Cock and Bart De Moor. Subspace angles between arma models. *Systems & Control Letters*, 46(4):265–270, 2002. [2](#), [31](#)
- [10] 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, 2009. [24](#)
- [11] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyang Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019. [6](#)
- [12] Inderjit S Dhillon, Jr RW Heath, Thomas Strohmer, and Joel A Tropp. Constructing packings in grassmannian manifolds via alternating projection. *Experimental mathematics*, 17(1):9–35, 2008. [29](#)
- [13] Gianfranco Doretto, Alessandro Chiuso, Ying Nian Wu, and Stefano Soatto. Dynamic textures. *International journal of computer vision*, 51:91–109, 2003. [16](#)
- [14] G. Golub, S. Nash, and C. Van Loan. A hessenberg-schur method for the problem $ax + xb = c$. *IEEE Transactions on Automatic Control*, 24(6):909–913, 1979. [5](#), [20](#)
- [15] Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D Bagdanov, Bartłomiej Twardowski, and Joost van de Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *European Conference on Computer Vision*, pages 473–490. Springer, 2024. [23](#)
- [16] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005. [25](#)
- [17] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech 256, 2022. [1](#), [6](#), [24](#)
- [18] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. [1](#), [2](#), [3](#), [5](#), [6](#), [8](#), [19](#)
- [19] Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022. [3](#), [5](#), [8](#), [19](#)
- [20] Albert Gu, Karan Goel, and Christopher Re. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. [1](#), [3](#), [15](#)
- [21] Ankit Gupta, Albert Gu, and Jonathan Berant. Diagonal state spaces are as effective as structured state spaces. *Advances in Neural Information Processing Systems*, 35:22982–22994, 2022. [3](#), [19](#)
- [22] Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone. *arXiv preprint arXiv:2407.08083*, 2024. [1](#), [6](#)
- [23] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. 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, 2021. [1](#), [6](#), [24](#)
- [24] 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. [1](#), [6](#), [7](#), [32](#)
- [25] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018. [6](#)

- [26] Wenbing Huang, Fuchun Sun, Lele Cao, Deli Zhao, Huaping Liu, and Mehrtash Harandi. Sparse coding and dictionary learning with linear dynamical systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3938–3947, 2016. 2
- [27] Wenbing Huang, Mehrtash Harandi, Tong Zhang, Lijie Fan, Fuchun Sun, and Junzhou Huang. Efficient optimization for linear dynamical systems with applications to clustering and sparse coding. *Advances in neural information processing systems*, 30, 2017. 2, 5, 6, 8, 14, 20, 26
- [28] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 699–715. Springer, 2020. 6
- [29] 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. 1, 6, 7, 8, 23, 24, 28, 32
- [30] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019. 25
- [31] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009. 1, 6, 24
- [32] Xiaojie Li, Yibo Yang, Jianlong Wu, Bernard Ghanem, Liqiang Nie, and Min Zhang. Mamba-fscil: Dynamic adaptation with selective state space model for few-shot class-incremental learning. *CoRR*, abs/2407.06136, 2024. 6, 23, 32
- [33] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 6, 7, 8, 23, 24, 28, 32
- [34] Jingyong Liang, Bernd Meyer, Isaac Ning Lee, and Thanh-Toan Do. Self-supervised learning for acoustic few-shot classification. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025. also as arXiv preprint arXiv:2409.09647 [cs.SD]. 1, 6
- [35] 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*, pages 23638–23647, 2024. 6
- [36] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Pengju An, Xiaoqi Li, Kaichen Zhou, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation. *Advances in Neural Information Processing Systems*, 37:40085–40110, 2024. 1, 6
- [37] Ruiqi Liu, Boyu Diao, Libo Huang, Zijia An, Zhulin An, and Yongjun Xu. Continual learning in the frequency domain. *Advances in Neural Information Processing Systems*, 37:85389–85411, 2024. 1, 6, 7, 32
- [38] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *Advances in neural information processing systems*, 37:103031–103063, 2025. 1, 3, 6, 8
- [39] Ekdeep Singh Lubana, Puja Trivedi, Danai Koutra, and Robert Dick. How do quadratic regularizers prevent catastrophic forgetting: The role of interpolation. In *Conference on Lifelong Learning Agents*, pages 819–837. PMLR, 2022. 6
- [40] 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. In *ICLR*, 2024. 23
- [41] Leila Mahmoodi, Mehrtash Harandi, and Peyman Moghadam. Flashback for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 3434–3443, 2023. 6
- [42] Leila Mahmoodi, Peyman Moghadam, Munawar Hayat, Christian Simon, and Mehrtash Harandi. Flashbacks to harmonize stability and plasticity in continual learning. *Neural Networks*, 190:107616, 2025. 6
- [43] Richar J. Martin. A metric for arma processes. *IEEE transactions on Signal Processing*, 48(4):1164–1170, 2000. 31
- [44] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995. 2
- [45] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 2
- [46] Zichong Meng, Jie Zhang, Changdi Yang, Zheng Zhan, Pu Zhao, and Yanzhi Wang. Diffclass: Diffusion-based class incremental learning. In *European Conference on Computer Vision*, pages 142–159. Springer, 2024. 6
- [47] 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. 6
- [48] Hao Phung, Quan Dao, Trung Dao, Viet Hoang Phan, Dimitris Metaxas, and Anh Tran. Dimsum: Diffusion mamba—a scalable and unified spatial-frequency method for image generation. *Advances in Neural Information Processing Systems*, 37:32947–32979, 2024. 1, 6
- [49] Avinash Ravichandran, Rizwan Chaudhry, and Rene Vidal. Categorizing dynamic textures using a bag of dynamical systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):342–353, 2012. 2, 26
- [50] 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. 23
- [51] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, , and Gerald Tesauro. Learning to learn

- without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*, 2019. [1](#), [6](#), [7](#), [32](#)
- [52] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. [6](#)
- [53] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021. [6](#)
- [54] Pavan Turaga, Ashok Veeraraghavan, Anuj Srivastava, and Rama Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2273–2286, 2011. [2](#)
- [55] Guido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. [6](#)
- [56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. [19](#)
- [57] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. [24](#)
- [58] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024. [2](#), [8](#), [14](#), [24](#)
- [59] 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, 2022. [1](#), [6](#), [7](#), [32](#)
- [60] Ke Ye and Lek-Heng Lim. Schubert varieties and distances between subspaces of different dimensions. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1176–1197, 2016. [3](#), [5](#), [8](#), [21](#), [26](#), [29](#), [30](#)
- [61] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. [1](#), [6](#), [7](#), [8](#), [23](#), [24](#), [28](#), [32](#)
- [62] Zeyu Zhang, Akide Liu, Ian Reid, Richard Hartley, Bohan Zhuang, and Hao Tang. Motion mamba: Efficient and long sequence motion generation. In *European Conference on Computer Vision*, pages 265–282. Springer, 2024. [1](#)
- [63] Chongyang Zhao and Dong Gong. Learning mamba as a continual learner. *CoRR*, abs/2412.00776, 2024. [6](#), [23](#), [32](#)
- [64] 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. [23](#)
- [65] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning*, 2024. [1](#), [2](#), [3](#), [5](#), [6](#), [8](#), [19](#), [20](#), [28](#), [33](#)