Exemplar-Free Continual Learning for State Space Models

Supplementary Material

Contents

A. Notations

Table 4. Summary of Mathematical Notations

Notation	Description
	General mathematical operations
$\mathbf{X}$	Matrix (bold capital letter)
$\boldsymbol{x}$	Column vector (bold lowercase letter)
$\mathbf{I}_n$	$n \times n$ identity matrix
$\mathbf{1}_n$	$n \times n$ ones matrix
$\mathbf{X}_{\text{diag}}$	Vector of diagonal elements of $\mathbf{X} \in \mathbb{R}^{n \times n}$, $\mathbf{X}_{\text{diag}} \in \mathbb{R}^{n \times 1}$
$\overline{\mathbf{X}}$	The discretized signal of $\mathbf{X}$
$\|\mathbf{A}\|_{\text{F}}$	Frobenius norm of $\mathbf{A} \in \mathbb{R}^{m \times n}$: $\sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{A}[i,j]^2}$
$\mathbf{A} \odot \mathbf{B}$	Hadamard product of two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$: $(\mathbf{A} \odot \mathbf{B})[i,j] = \mathbf{A}[i,j]\,\mathbf{B}[i,j]$.
$\det(\mathbf{A})$	Determinant of matrix $\mathbf{A}$
$\text{SN}(x)$	Soft-normalization function: $\frac{2}{1+\exp(-x)} - 1$ [27]
$\text{GL}(n)$	General linear group: $\{\mathbf{P} \in \mathbb{R}^{n \times n} \mid \det(\mathbf{P}) \neq 0\}$
$\text{Gr}(p, n)$	Grassmann manifold of the set of all p-dimensional linear subspaces of $\mathbb{R}^n$
Tr	Trace operator
$\boldsymbol{x}[\cdot]$	Discrete-time vector
$\boldsymbol{x}(\cdot)$	Continuous-time vector
	SSM notations
$\mathbf{A}$	State matrix of SSM in continuous time domain
$\mathbf{B}$	Input matrix of SSM in continuous time domain
$\mathbf{C}$	Output matrix of SSM
x	Input to SSM, where $x(t) \in \mathbb{R}$
$\boldsymbol{h}$	Hidden (state) vector of SSM, where $\boldsymbol{h}(t) \in \mathbb{R}^n$
y	output of SSM, where $y(t) \in \mathbb{R}$
$\mathbf{O}_\infty$	Extended Observability matrix of SSM
$\tilde{\mathbf{A}}$	$\overline{\mathbf{A}}$ averaged across outer dimension of SSM applied with SN Eq. (31)
$\tilde{\mathbf{B}}$	$\overline{\mathbf{B}}$ averaged across outer dimension of SSM applied with SN Eq. (31)
$\tilde{\mathbf{C}}$	$\mathbf{C}$ applied with SN Eq. (31)
$\mathbf{P}$	Any invertible $n \times n$ matrix for P-equivalence of LDS §C.1
	Vim specific notations
b	Batch size of input to SSM in Mamba and Vim
τ	Sequence length of SSM in Mamba and Vim
o	Outer dimension size of SSM in Mamba and Vim
	Grassmannian notations
θ_i	Principal angle for i-th dimension
$\mathbf{G}$	Gram matrix
$\mathcal{S}$	Notation of subspace, in particular $\mathcal{S} \in \text{Gr}(n, m)$
	Continual Learning notations
T	T-th task's identifier
$\mathcal{T}_T$	Sequential T-th task
N	Total number of task
$\text{L}_{\texttt{ISM}}$	The proposed Inf-SSM regularization loss function Eq. (15)
$\text{L}_{\texttt{ISM+}}$	The proposed Inf-SSM+ regularization loss function Eq. (39)
CKD	Centered Kernel-Disparity, Eq. (38)
AIA	Average Incremental Accuracy §G.2 [58]
AA	Average Accuracy §G.2 [58]
FM	Forgetting Measure §G.2 [58]

B. Preliminary

Definition B.1 (Principal Angles)**.** *For two subspaces in* $\mathrm{Gr}(n, d)$*, let* $\mathbf{X}$ *and* $\mathbf{Z}$ *be their orthonormal basis matrices. The* ***principal angles*** $\theta_1, \theta_2, \ldots, \theta_n$ *between the subspaces are defined as:*

$$\cos\theta_i = \sigma_i(\mathbf{X}^\top \mathbf{Z}), \tag{17}$$

where σ_i *is the* i*-th singular value of the matrix* $\mathbf{X}^\top \mathbf{Z}$*, respectively.*

The *geodesic distance* on $\mathrm{Gr}(n, d)$, inherited from its usual Riemannian metric, is given by

$$d_g(\mathbf{X}, \mathbf{Z}) = \sqrt{\theta_1^2 + \theta_2^2 + \cdots + \theta_n^2}\ . \tag{18}$$

Intuitively, θ_i measures how much the i-th principal direction of $\mathbf{X}$ deviates from that of $\mathbf{Z}$, and thus this distance quantifies the "angle" between two subspaces in a higher-dimensional setting.

B.1. Discretization of SSMs

A linear dynamic system is described by a classical **State-Space Model (SSM)** over time in the form of:

$$\begin{aligned} \dot{\boldsymbol{h}}(t) &= \mathbf{A}\boldsymbol{h}(t) + \mathbf{B}x(t), \\ y(t) &= \mathbf{C}\boldsymbol{h}(t)\ . \end{aligned} \tag{19}$$

Here, $\boldsymbol{h}(t) \in \mathbb{R}^n$ represents the **hidden state**, while $x(t), y(t) \in \mathbb{R}$ are the input and output of the SSM, respectively. The model is parametrized by: **1.** the state transition matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, **2.** the input mapping matrix $\mathbf{B} \in \mathbb{R}^{n\times 1}$, and **3.** the output mapping matrix $\mathbf{C} \in \mathbb{R}^{1\times n}$. To adopt SSM in machine learning, SSM needs to be discretized by introducing a sampling interval $\Delta \in \mathbb{R}$, which transforms the continuous parameters $\mathbf{A}$ and $\mathbf{B}$ into their discrete equivalents $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ using the Zero-Order Hold method [20]:

$$\overline{\mathbf{A}} = \exp(\Delta\mathbf{A})\ , \tag{20}$$

$$\overline{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}_n)\Delta\mathbf{B}\ . \tag{21}$$

The discrete time domain version of Eq. (19) can be written as

$$\mathbf{h}[t] = \overline{\mathbf{A}}\mathbf{h}[t-1] + \overline{\mathbf{B}}x[t], \tag{22}$$

$$y[t] = \mathbf{C}\mathbf{h}[t]. \tag{23}$$

This discretization allows the computation of y_t via a convolution operation rather than an explicit recurrence, greatly simplifying the computational complexity. **Note:** We represent $\overline{\mathbf{A}}, \overline{\mathbf{B}}$ as $\mathbf{A}, \mathbf{B}$ in the main text for better readability.

C. Proof

C.1. P-Equivalence

Lemma C.1 (P-Equivalence [13])**.** *Consider the SSM given by*

$$\begin{cases} \boldsymbol{h}[t] = \mathbf{A}\boldsymbol{h}[t-1] + \mathbf{B}x[t], \\ y[t] = \mathbf{C}\boldsymbol{h}[t], \end{cases}$$

where $\boldsymbol{h}[t] \in \mathbb{R}^n$ is the hidden (state) vector, $x[t] \in \mathbb{R}$ is the input, and $y[t] \in \mathbb{R}$ is the output. Let $\mathbf{P}$ be any invertible $n \times n$ matrix. Define

$$\mathbf{A}' \stackrel{\text{def}}{=} \mathbf{P}\mathbf{A}\mathbf{P}^{-1}, \quad \mathbf{B}' \stackrel{\text{def}}{=} \mathbf{P}\mathbf{B}, \quad \mathbf{C}' \stackrel{\text{def}}{=} \mathbf{C}\mathbf{P}^{-1},$$

and let the new state be $\tilde{\boldsymbol{h}}[t] = \mathbf{P}\boldsymbol{h}[t]$. Then the SSM

$$\begin{cases} \tilde{\boldsymbol{h}}[t] = \mathbf{A}'\tilde{\boldsymbol{h}}[t-1] + \mathbf{B}'x[t], \\ y[t] = \mathbf{C}'\tilde{\boldsymbol{h}}[t] \end{cases}$$

displays the same input-output behavior as the original system. Consequently, the triples $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $(\mathbf{A}', \mathbf{B}', \mathbf{C}')$ are said to be equivalent representations *of the same SSM.*

Proof. Since $\mathbf{P}$ is invertible, $\boldsymbol{h}[t] = \mathbf{P}^{-1}\tilde{\boldsymbol{h}}[t]$. We have

$$\boldsymbol{h}[t+1] = \mathbf{P}^{-1}\tilde{\boldsymbol{h}}[t+1] = \mathbf{P}^{-1}\Big(\mathbf{A}'\tilde{\boldsymbol{h}}[t] + \mathbf{B}'x[t]\Big) = \mathbf{P}^{-1}\mathbf{A}'\tilde{\boldsymbol{h}}[t] + \mathbf{P}^{-1}\mathbf{B}'x[t] \ .$$

Since $\tilde{\boldsymbol{h}}[t] = \mathbf{P}\boldsymbol{h}[t]$, we get

$$\boldsymbol{h}[t+1] = \mathbf{P}^{-1}\mathbf{A}'\mathbf{P}\boldsymbol{h}[t] + \mathbf{P}^{-1}\mathbf{B}'x[t] = \mathbf{A}\boldsymbol{h}[t] + \mathbf{B}x[t];,$$

where $\mathbf{A} = \mathbf{P}^{-1}\mathbf{A}'\mathbf{P}$ and $\mathbf{B} = \mathbf{P}^{-1}\mathbf{B}'$ by definition. For the output, from $y[t] = \mathbf{C}'\tilde{\boldsymbol{h}}[t]$ and again using $\tilde{\boldsymbol{h}}[t] = \mathbf{P}\boldsymbol{h}[t]$, we obtain

$$y[t] = \mathbf{C}'\tilde{\boldsymbol{h}}[t] = \mathbf{C}'\big(\mathbf{P}\boldsymbol{h}[t]\big) = \big(\mathbf{C}'\mathbf{P}\big)\boldsymbol{h}[t] = \mathbf{C}\boldsymbol{h}[t].$$

Therefore, at each time t, for the same input $x[t]$, the two systems $\big(\mathbf{A}', \mathbf{B}', \mathbf{C}'\big)$ on $\tilde{\boldsymbol{h}}[t]$ and $\big(\mathbf{A}, \mathbf{B}, \mathbf{C}\big)$ on $\boldsymbol{h}[t]$ generate the same output $y[t]$. □

C.2. Invariance of the subspace spanned by the observability matrix under P-equivalence

Theorem C.2 (Invariance of the extended Observability under P-equivalence)**.** *Let $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ and $(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}, \mathbf{P}\mathbf{B}, \mathbf{C}\mathbf{P}^{-1})$ be two equivalent representations for an SSM for $\mathbf{P} \in \mathrm{GL}(n)$. The extended observability subspaces satisfy:*

$$\mathcal{S}_\infty(\mathbf{A}', \mathbf{C}') = \mathcal{S}_\infty(\mathbf{A}, \mathbf{C}) \ . \tag{24}$$

Proof. The extended observability matrix for the transformed system is given by

$$\mathbf{O}_\infty(\mathbf{A}', \mathbf{C}') = \begin{bmatrix} \mathbf{C}' \\ \mathbf{C}'\mathbf{A}' \\ \mathbf{C}\mathbf{A}'^2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{C}\mathbf{P}^{-1} \\ \mathbf{C}\mathbf{P}^{-1}\big(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}\big) \\ \mathbf{C}\mathbf{P}^{-1}\big(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}\big)^2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{C}\mathbf{P}^{-1} \\ \mathbf{C}\mathbf{A}\mathbf{P}^{-1} \\ \mathbf{C}\mathbf{A}^2\mathbf{P}^{-1} \\ \vdots \end{bmatrix} = \boxed{\mathbf{O}_\infty(\mathbf{A}, \mathbf{C})\mathbf{P}^{-1}} \ .$$

Since $\mathbf{P}^{-1}$ is an invertible transformation, it does not change the span of the subspace. Therefore, we conclude that

$$\mathcal{S}_\infty(\mathbf{A}, \mathbf{C}) = \mathcal{S}_\infty(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}, \mathbf{C}\mathbf{P}^{-1}) = \mathcal{S}_\infty(\mathbf{A}', \mathbf{C}') \ .$$

□

C.3. Distances on Infinite Grassmannian

Let $\mathcal{S}, \mathcal{S}' \in \mathrm{Gr}(n, d)$. The chordal distance, aka projection distance, between $\mathcal{S}, \mathcal{S}'$ is defined as

$$d^2_{\text{chord}}(\mathcal{S}, \mathcal{S}') = \|\mathcal{S}\mathcal{S}^\top - \mathcal{S}'\mathcal{S}'^\top\|_{\text{F}}^2 = 2n - 2\|\mathcal{S}^\top\mathcal{S}'\|_{\text{F}}^2 \,. \tag{25}$$

Since in our case, $d \to \infty$, computing $d^2_{\text{chord}}(\mathcal{S}, \mathcal{S}')$ is not straightforward as one cannot compute $\|\mathcal{S}^\top\mathcal{S}'\|_{\text{F}}^2$ using explicit forms for $\mathcal{S}, \mathcal{S}'$. Below, we show how this can be done without the need to form $\mathcal{S}, \mathcal{S}' \in \mathrm{Gr}(n, \infty)$ explicitly. We start by stating a result from linear algebra.

Let $\mathbb{R}^{d\times n} \ni \mathbf{O} = \begin{bmatrix}\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots, \boldsymbol{o}_n\end{bmatrix}$ be a full rank matrix. The n-dimensional subspace spanned by the columns of $\mathbf{O}$ can be written as

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

This can be readily seen by verifying $\mathcal{S}^\top\mathcal{S} = (\mathbf{O}^\top\mathbf{O})^{-1/2}\mathbf{O}^\top\mathbf{O}(\mathbf{O}^\top\mathbf{O})^{-1/2} = \mathbf{I}_n$. As such, we can express $d^2_{\text{chord}}(\mathcal{S}, \mathcal{S}')$ as

$$\begin{aligned}
d^2_{\text{chord}}(\mathcal{S}, \mathcal{S}') &= 2n - 2\|\mathcal{S}^\top\mathcal{S}'\|_{\text{F}}^2 = 2n - 2\,\mathrm{Tr}\left\{\mathcal{S}^\top\mathcal{S}'\mathcal{S}'^\top\mathcal{S}\right\} \\
&= 2n - 2\,\mathrm{Tr}\left\{(\mathbf{O}^\top\mathbf{O})^{-1/2}\mathbf{O}^\top\mathbf{O}'(\mathbf{O}'^\top\mathbf{O}')^{-1/2}(\mathbf{O}'^\top\mathbf{O}')^{-1/2}\mathbf{O}'^\top\mathbf{O}(\mathbf{O}^\top\mathbf{O})^{-1/2}\right\} \\
&= 2n - 2\,\mathrm{Tr}\left\{(\mathbf{O}^\top\mathbf{O})^{-1}\mathbf{O}^\top\mathbf{O}'(\mathbf{O}'^\top\mathbf{O}')^{-1}\mathbf{O}'^\top\mathbf{O}\right\} \,.
\end{aligned}$$

As such, one needs to be able to compute terms such as $\mathbf{G}_1 = (\mathbf{O}^\top\mathbf{O})$, $\mathbf{G}_2 = (\mathbf{O}'^\top\mathbf{O}')$, $\mathbf{G}_3 = (\mathbf{O}^\top\mathbf{O}')$ and $\mathbf{G}_4 = (\mathbf{O}'^\top\mathbf{O})$ for observability matrices. The lemma below shows how this can be done.

Lemma C.3. *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*

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

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

$$\begin{aligned}
\mathbf{G} &= \mathbf{O}_\infty(\mathbf{A}, \mathbf{C})^\top\mathbf{O}_\infty(\mathbf{A}', \mathbf{C}') \\
&= \begin{bmatrix}\mathbf{C}, \mathbf{C}\mathbf{A}, \mathbf{C}\mathbf{A}^2, \ldots\end{bmatrix}\begin{bmatrix}\mathbf{C}' \\ \mathbf{C}'\mathbf{A}' \\ \mathbf{C}'(\mathbf{A}')^2 \\ \vdots\end{bmatrix} \\
&= \sum_{t=0}^{\infty}(\mathbf{A}^\top)^t\mathbf{C}^\top\mathbf{C}'(\mathbf{A}')^t \,.
\end{aligned}$$

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

$$\mathbf{A}^\top\mathbf{G}\mathbf{A}' - \mathbf{G} = -\mathbf{C}^\top\mathbf{C}' \,. \tag{26}$$

Proof. Using the definition of the observability matrices,

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

Multiplying both sides by $\mathbf{A}^\top$ on the left and $\mathbf{A}'$ on the right:

$$\begin{aligned}
\mathbf{A}^\top \mathbf{G} \mathbf{A}' &= \sum_{t=0}^{\infty} (\mathbf{A}^\top)^{t+1} \mathbf{C}^\top \mathbf{C}' (\mathbf{A}')^{t+1} \\
&= \sum_{t=1}^{\infty} (\mathbf{A}^\top)^{t} \mathbf{C}^\top \mathbf{C}' (\mathbf{A}')^{t} \\
&= \underbrace{\sum_{t=0}^{\infty} (\mathbf{A}^\top)^{t} \mathbf{C}^\top \mathbf{C}' (\mathbf{A}')^{t}}_{\mathbf{G}} - (\mathbf{A}^\top)^{0} \mathbf{C}^\top \mathbf{C}' (\mathbf{A}')^{0} \\
&= \boxed{\mathbf{G} - \mathbf{C}^\top \mathbf{C}'} \, .
\end{aligned}$$

□

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] to obtain $\mathbf{G}$. The computational complexity of solving the Sylvester algorithm is $\mathcal{O}(n^3)$. This is quite affordable as in our problem, n is typically very small ($n = 16$ in our experiments in Vim-small).

Definition C.4 (Sylvester Equation)**.** *A matrix problem in the form*

$$\mathbf{V}\mathbf{X} + \mathbf{X}\mathbf{U} = \mathbf{Q} \, ,$$

for $\mathbf{V} \in \mathbb{R}^{n \times n}$, $\mathbf{U} \in \mathbb{R}^{d \times d}$ and $\mathbf{Q} \in \mathbb{R}^{n \times d}$ over $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a Sylvester equation. A Sylvester equation has a unique solution if $\mathbf{V}$ and $-\mathbf{U}$ do not share any eigenvalue.

In our case,

$$\mathbf{V} = \mathbf{A}^\top \tag{27}$$

$$\mathbf{U} = -\mathbf{A}'^{-1} \tag{28}$$

$$\mathbf{Q} = -\mathbf{C}^\top \mathbf{C}' \mathbf{A}'^{-1} \tag{29}$$

D. SSMs, Vision Mamba and Inf-SSM

D.1. Selective State-Space Models.

Computation of S4 is slow due to $\mathbf{A}$ being parameterized as an $n \times n$ matrix. DSS [21] first shows that a diagonal state space always exists for any state space with a well-behaved matrix. S4D [19] further improves the computational efficiency by computing SSM only with real numbers and re-expressing the computation of the convolution kernel as a Vandermonde matrix. Mamba [18] and Vision Mamba [65] (Vim) inherit the diagonal $\mathbf{A}$ with further modifications on the model's structure.

A key limitation of SSMs and S4 is that they are time-invariant. As argued by Gu and Dao [18], such constant dynamics prevent the model from selecting or filtering relevant information based on the input context. To address this issue, Selective State-Space Models (S6) [18] introduce **input-dependent parameters**, making the model time-varying and adaptive. Specifically, instead of keeping $\Delta, \mathbf{B}, \mathbf{C}$ fixed, they are modeled as functions of the input. For example, $\mathbf{B}[t] = f_B(x(t)) = \mathbf{W}_B x(t)$.

This change significantly enriches the S6; however, it also loses the computational efficiency of the SSMs and S4, as the output can no longer be computed via convolution. The Mamba block is a selective SSM architecture inspired by S6, but with additional input-dependent transformations. Recent work in Mamba-2 [8], further shows that SSMs and Transformers [56] are indeed dual and linked via semi-separable matrices. This potentially allows algorithms developed independently in SSMs and Transformers could be mutually beneficial to each other.

D.2. Vision Mamba

As SSM is implemented to handle 1-D sequence data, Vision Mamba [65] transformed 2-D images into patches in 2-D. The sequence of patches is then linearly projected and added with a position embedding, forming a patch sequence. Similar to ViT, Vim utilizes a class token to capture the global information, and the entire patch sequence, including the class token, is fed into the SSM structure.

D.3. State approximation in S4D

Let $(\mathbf{A}, \mathbf{C})$ and $(\mathbf{A}', \mathbf{C}')$ be the tuples representing two SSMs. The SSMs could be described respectively by using the extended observability subspace formed by the tuples as discussed in Theorem C.3.

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

The discrete-time form of the SSM equation can be expressed as:

$$\begin{aligned} \mathbf{h}[t] &= \overline{\mathbf{A}}\mathbf{h}[t-1] + \overline{\mathbf{B}}x[t], \\ y[t] &= \mathbf{C}\mathbf{h}[t]. \end{aligned} \tag{30}$$

In Mamba, the dimensionality of each state is: $\overline{\mathbf{A}} \in \mathbb{R}^{\tau \times o \times n}$, $\overline{\mathbf{B}} \in \mathbb{R}^{\tau \times o \times n}$, and $\mathbf{C} \in \mathbb{R}^{\tau \times n}$. Since it is computationally infeasible to compute for the entire $\tau \times o$ pairs of states with dimensionality of n, we treat each τ as an independent trajectory applied over an infinite horizon. To justify our choice, as shown in Table 5, we measured the variance preserved when averaging over different axes. Averaging over o retains more informative variance than alternatives like averaging over τ or n, suggesting that it captures meaningful dynamics while enabling tractable computation.

Table 5. Mean and standard deviation of variance preserved in $\bar{\mathbf{A}}$ after averaging across different dimensions on ImageNet-R.

Dimension	Mean	Standard Deviation
τ	0.0117	0.0085
$\mathbf{o}$	0.0315	0.0307
n	0.0004	0.0003

We acknowledge that this approximation has limitations. Averaging over o may fail to capture fine-grained variations. However, our empirical results in §5 and §G suggest that this approximation is efficient without sacrificing performance.

Thus, we define:

$$\begin{aligned}\tilde{\mathbf{A}} &= \mathrm{SN}(\frac{1}{o}\sum_{i=1}^{o}\overline{\mathbf{A}}_{i,j}) \in \mathbb{R}^{\tau\times n},\\ \tilde{\mathbf{B}} &= \mathrm{SN}(\frac{1}{o}\sum_{i=1}^{o}\overline{\mathbf{B}}_{i,j}) \in \mathbb{R}^{\tau\times n},\\ \tilde{\mathbf{C}} &= \mathrm{SN}(\mathbf{C}) \in \mathbb{R}^{\tau\times n}.\end{aligned} \tag{31}$$

By following Huang *et al*. [27], Soft-Normalization $\mathrm{SN}(x) = 2/(1+\exp(-x)) - 1$ is applied to ensure Schur Stability. Note that $\mathbf{B}, \mathbf{C}$ do not necessarily need to have SN applied to ensure Schur Stability but SN is applied for the consistency in magnitude across $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$

D.4. Derivation of Gram matrix for Vim

For two S4D blocks represented by $\tilde{\mathbf{A}}, \tilde{\mathbf{C}}$ and $\tilde{\mathbf{A}}', \tilde{\mathbf{C}}'$. The Sylvester equation for $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{A}}', \tilde{\mathbf{C}}'$ is given by:

$$\tilde{\mathbf{A}}\mathbf{G}\tilde{\mathbf{A}}'^{\top} - \mathbf{G}_{ij} = -\tilde{\mathbf{C}}^{\top}\tilde{\mathbf{C}}'.$$

Thus, by multiplying by -1 on both sides:

$$\mathbf{G} - \tilde{\mathbf{A}}\mathbf{G}\tilde{\mathbf{A}}'^{\top} = \tilde{\mathbf{C}}^{\top}\tilde{\mathbf{C}}'.$$

Since $\mathbf{A}$ is diagonal and $\mathbf{C} \in \mathbb{R}^{n\times 1}$ as mentioned in §D, this allows simplification by using the Hadamard product. Thus, after simplification, the solution to the Gram matrix is:

$$\mathbf{G} - \tilde{\mathbf{A}}_{\text{diag}}\tilde{\mathbf{A}}'^{\top}_{\text{diag}} \odot \mathbf{G} = \tilde{\mathbf{C}}^{\top}\tilde{\mathbf{C}}'.$$

By collecting the element-wise factor of $\mathbf{G}$

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

Hence, by grouping the terms to form a solution for $\mathbf{G}$ will obtain:

$$\mathbf{G} = \tilde{\mathbf{C}}^{\top}\tilde{\mathbf{C}}' \odot \frac{1}{\mathbf{1}_n - \tilde{\mathbf{A}}_{\text{diag}}\tilde{\mathbf{A}}'^{\top}_{\text{diag}}} = \mathbf{O}^{\top}\mathbf{O}'. \tag{32}$$

For the formulation in Eq. (32), it could be broken down into four steps:

1. Matrix multiplication $\tilde{\mathbf{C}}^{\top}\tilde{\mathbf{C}}$
2. Matrix multiplication $\tilde{\mathbf{A}}_{\text{diag}}\tilde{\mathbf{A}}'^{\top}_{\text{diag}}$
3. Subtraction $\mathbf{1}_n - \tilde{\mathbf{A}}_{\text{diag}}\tilde{\mathbf{A}}'^{\top}_{\text{diag}}$
4. Element-wise division of $\tilde{\mathbf{C}}^{\top}\tilde{\mathbf{C}}$ over $\mathbf{1}_n - \tilde{\mathbf{A}}_{\text{diag}}\tilde{\mathbf{A}}'^{\top}_{\text{diag}}$

Note that $\odot$ and element-wise reciprocal could be combined as a single step by simply taking element-wise reciprocal of $\tilde{\mathbf{C}}^{\top}\tilde{\mathbf{C}}'$ by $\mathbf{1}_n - \tilde{\mathbf{A}}_{\text{diag}}\tilde{\mathbf{A}}'^{\top}_{\text{diag}}$. Thus, each outlined step have an FLOPS count of n^2, and hence, the total FLOPS count is $4n^2$ with computational complexity of $\mathcal{O}(n^2)$.

Hence, 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 $\mathbf{100}\times$.

D.5. Simplified Distance on Grassmannian

Empirically, we could compute distance on the Grassmannian using Eq. (26). However, we observed that due to $\mathbf{A}$ being diagonal in S4D, Mamba, and Vim, $\mathbf{G}_i, i \in \{1, 2, 3, 4\}$ normally have a dominant eigenvalue (principal angle) with small components of other principal directions. In particular, at $n = 16$, the majority of eigenvalues are close to 0, leading $\mathbf{O}$ to be very ill-conditioned and with zero determinant. Thus, we model $\mathbf{O}$ as a noisy rank-1 matrix.

Let $\mathbf{O}_1 = \boldsymbol{a}_1\boldsymbol{b}_1^\top \in \mathbb{R}^{\infty\times n}$ and $\mathbf{O}_2 = \boldsymbol{a}_2\boldsymbol{b}_2^\top \in \mathbb{R}^{\infty\times n}$ by approximating $\mathbf{O}$ as rank 1. We know that

$$\|\mathbf{O}_i\|_F = \sqrt{\mathrm{Tr}(\mathbf{O}_i^\top \mathbf{O}_i)},$$

where $\boldsymbol{a}_i \in \mathbb{R}^\infty$ is the column space and $\boldsymbol{b}_i \in \mathbb{R}^n$ and

$$\|\mathbf{O}_i^\top \mathbf{O}_j\|_F = \sqrt{\mathrm{Tr}(\mathbf{O}_j^\top \mathbf{O}_i \mathbf{O}_i^\top \mathbf{O}_j)}.$$

Hence, the principal angle between the two SSMs is

$$\cos\theta = \frac{\|\mathbf{O}_1^\top \mathbf{O}_2\|_F}{\|\mathbf{O}_1\|_F \|\mathbf{O}_2\|_F} = \sqrt{\frac{\mathrm{Tr}(\mathbf{O}_2^\top \mathbf{O}_1 \mathbf{O}_1^\top \mathbf{O}_2)}{\mathrm{Tr}(\mathbf{O}_1^\top \mathbf{O}_1)\,\mathrm{Tr}(\mathbf{O}_2^\top \mathbf{O}_2)}}.$$

Thus, the squared principal angle is

$$\cos^2\theta = \frac{\mathrm{Tr}(\mathbf{G}_3\mathbf{G}_4)}{\mathrm{Tr}(\mathbf{G}_1)\,\mathrm{Tr}(\mathbf{G}_2)}. \tag{33}$$

Note that this formulation does not necessarily allow $\cos\theta = 1$ when $\mathbf{O}_1 = \mathbf{O}_2$ if they are not rank 1. Thus, to ensure the simplification is reasonable, we have set up a Monte-Carlo simulation of 10,000 iterations with $n = 16$. For each iteration, we sample $\mathbf{A}_{\text{diag}}, \mathbf{B}, \mathbf{C} \sim \mathcal{N}(0, I_n)$. We then sample noise to simulate weight update during a sequential training scenario by sampling $\epsilon_i \sim \mathcal{N}(0, \frac{i \cdot I_n}{25})$ for $i \in \{0, \ldots, 99\}$. Then, we measure the correlation between i and our approximated $\cos\theta$. The Monte-Carlo test shows that for the average across 10,000 iterations, the Pearson correlation coefficient is -0.8962 with a standard deviation of 0.01515 and a mean p-value of 8.151e-30 with a standard deviation of 2.827e-28. This shows that the simplification is reasonable and applicable in Continual Learning regularization. For the problem $\cos\theta \neq 1$ when $\mathbf{O}_1 = \mathbf{O}_2$, we simply counteract it by defining:

$$\cos\theta = 1 \quad \text{if } |\mathbf{A} - \mathbf{A}'| \leq \epsilon \cap |\mathbf{C} - \mathbf{C}'| \leq \epsilon.$$

Formulation of Eq. (33) is clearly faster than other distance measures on Grassmannian outlined by Ye and Lim [60] as it does not involve inverse and determinant at all. Next, as $\mathbf{O}$ is rank deficient and ill-conditioned, computation of inverse and determinant is numerically unstable, while Eq. (33) only involves Trace operation, which is always numerically stable.

E. Inf-SSM Algorithm

In this section, we will discuss the Inf-SSM algorithm in the Continual Learning setting.

Algorithm 1 Inf-SSM State Regularization in EFCIL

Input: Frozen old model $f_{T-1}(\cdot; \mathbf{W}_{T-1})$, current-task dataset $\mathcal{D}_T$, regularization weight λ, learning rate α
Output: Updated model $f_T(\cdot; \mathbf{W}_T^*)$

```
for epoch = 1, ..., E do
    for mini-batch (X, Y) ~ D_T do
        /* Forward pass in f_{T-1}(·; W_{T-1}) and f_T(·; W_T) */
        Compute current-task logits Z_T ← f_T(X; W_T)
        /* Compute classification loss for current task */
        ℓ_cls ← L_cls(Z_T, Y)
        /* State extraction from past and current model */
        Extract M_old(Ã_old, C̃_old) from f_{T-1}(·; W_{T-1})
        Extract M_new(Ã_new, C̃_new) from f_T(·; W_T)
        /* Compute Inf-SSM loss for reg.*/                    ▷ Eq. (16)
        Let L_tot(W_T) = ℓ_cls + λ L_ISM(M_old, M_new)
        Update W_T ← W_T − α ∇L_tot(W_T)
    end for
end for
W_T^* ← W_T
```

For Inf-SSM, we require the previous task model $f_{T-1}(\cdot; \mathbf{W}_{T-1})$ and current task data $(\mathbf{x}, \mathbf{y}) \in (\mathcal{X}_T, \mathcal{Y}_T)$ for regularizing the new model $f_T(\cdot; \mathbf{W}_T)$. The current task data acts as a transport medium to extract the states of the previous task model and constrain the new model's weight update.

For every batch of training, we obtain the classification loss as normal. The classification loss is independent of the regularization step and thus can be set with any loss based on a specific training framework. During the forward pass, the Vim block intermediate states $\overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{C}$ are generated and extracted from the old and new model. This forms the input to the Inf-SSM loss function, which will penalize the weight update in the Infinite Observability subspace in the new model.

Limitations As the intermediate states $\overline{\mathbf{A}}, \overline{\mathbf{B}}, \mathbf{C}$ only exist during the computational scan process in SSM. This caused Inf-SSM to be "scan-breaking" as the intermediate state needed to be recomputed and saved outside of the scan for regularization purposes. This led to an increase in training time and VRAM requirements. However, if intermediate states could be extracted from the CUDA kernel efficiently, the computational speed of Inf-SSM should be similar to that of existing simple CL methods.

F. Additional Related Works

F.1. Hybrid Continual Learning Methods

In the main paper, we discussed foundational EFCIL methods of Elastic Weight Consolidation (EWC) [29], Synaptic Intelligence (SI) [61], and Memory-Aware Synapses (MAS) [3] from regularization approach and LwF [33, 50]. In this section, we focus on recent EFCIL methods that leverage complex hybrid architectures for better performance.

Self-sustaining representation expansion (SSRE) [64] utilizes dynamic structural reorganization to maintain old features. This is achieved by a dual-branch structure, a main branch for fusion and a side branch for updates to retain past knowledge. The main branch will undergo distillation to transfer shared knowledge across tasks with the help of the prototype selection algorithm to selectively incorporate new knowledge into the main branch. Meanwhile, LDC [15] compensates for semantic drift via a learnable projector network that aligns features across tasks, enabling compatibility with both supervised and semi-supervised settings. At the end of each task, the trained projector is utilized to correct and update the stored prototype. LDC corrects the drift in the prototype to improve performance in EFCIL settings. EFC [40] mitigates task-recency bias through prototype regularization and introduces a feature consolidation mechanism based on empirical feature drift. EFC combines the prototype pool replay, distillation, and regularization approach to mitigate catastrophic forgetting. By reducing feature drift, EFC improves the model's ability to learn new knowledge in EFCIL settings.

While SSRE, LDC, and EFC improve performance in exemplar-free scenarios, they introduce significantly more additional components and complexity compared to foundational CL algorithms like EWC [29], LwF [33], and Inf-SSM.

F.2. Continual Learning in Mamba

For CL in vision tasks using SSMs, Mamba-CL [7] enhances the stability of SSM outputs across past and current tasks by implementing orthogonality through null-space projection regularization. However, Mamba-CL needs to retain feature embeddings from all SSM modules to maintain consistency conditions for parameter updates. Meanwhile, MambaCL [63] incorporates meta-learning techniques to process an online data stream, enabling Mamba to function as a continual learner. MambaCL introduces selective regularization based on Mamba, linear transformers, and transformer connections. Although MambaCL recognizes the input-output relationship in SSMs, it does not account for the long-term evolution of SSM behavior, which is encoded in the extended observability subspace. Mamba-FSCIL [32] leverages a dual selective SSM projector to learn shifts in feature-space distribution and employs class-sensitive selective scan to improve model stability by reducing inter-class interference. Mamba-FSCIL requires the storage of intermediate feature embeddings from the old tasks distribution to utilize these features to improve the model's stability. From an SSM geometrical perspective, Mamba-FSCIL focuses on input-dependent parameters $\mathbf{B}, \mathbf{C}, \Delta$ in class-sensitive selective scan, where the key input state matrix $\mathbf{A}$ is not leveraged.

In short, while these recent works achieve impressive results within their respective settings, they are not directly comparable to Inf-SSM, as our focus lies in developing a fundamental continual learning algorithm that is flexible and complementary to mainstream CL methods rather than tailored to a specific scenario.

G. Experiments

G.1. Datasets

We conducted the experiments on four different datasets, namely: (1) **ImageNet-R** [23] with 30,000 images distributed unevenly in 200 classes of renditions of ImageNet [10]. (2) **CIFAR-100** [31], a balanced dataset of 100 classes consisting of 50,000 training images. (3) **Caltech-256** with 30,607 images from 256 classes [17]. We considered the first 250 classes for equal task partitioning. Each dataset is partitioned equally in terms of the number of classes across 5-task and 10-task scenarios.

In addition, we also utilized (4) **CUB-200-2011** [57] of the extended version 2011 with 11,788 images distributed unevenly across 200 classes for ablation studies purposes.

G.2. Continual Learning Evaluation Metrics

The notation of AA and AIA for test dataset of task $\mathcal{T}_j$ after training on $\mathcal{T}_k$ tasks where $j \leq k$ are as follow [58]:

$$\mathrm{AA}_k = \frac{1}{k}\sum_{j=1}^{k} a_{k,j}, \tag{34}$$

$$\mathrm{AIA}_k = \frac{1}{k}\sum_{i=1}^{k} \mathrm{AA}_i. \tag{35}$$

where $a_{k,j}$ is the accuracy of the test dataset for task $\mathcal{T}_j$ after the model is trained on task $\mathcal{T}_k$. Meanwhile FM at task k is defined as:

$$\mathrm{FM}_k = \frac{1}{k-1}\sum_{j=1}^{k-1} \max_{i\in 1,\ldots,k-1} (a_{i,j} - a_{k,j}). \tag{36}$$

G.3. Observability state parameter regularization

In this section, we present our experiment on observability state parameter regularization. For this experiment, all baseline methods, EWC [29], SI [61], and MAS [3] from the regularization-based category and LwF [33] from distillation methods are applied on weights directly contributing to the formation of state matrices $\mathbf{A}, \mathbf{C}$.

Table 6. AA(% ↑), AIA(% ↑), and FM(% ↓) of EFCIL methods on Vim-small are reported for ImageNet-R Dataset over 5 Tasks and 10 tasks benchmarks. **Note:** Regularization focus is on parameter sets $(\mathbf{A}, \mathbf{C})$ among all methods. Second-best results are underlined.

Method	ImageNet-R 5 task			ImageNet-R 10 task		
	$\mathrm{AA}_{\pm\mathrm{std}}$	$\mathrm{AIA}_{\pm\mathrm{std}}$	$\mathrm{FM}_{\pm\mathrm{std}}$	$\mathrm{AA}_{\pm\mathrm{std}}$	$\mathrm{AIA}_{\pm\mathrm{std}}$	$\mathrm{FM}_{\pm\mathrm{std}}$
Seq	$40.57_{\pm 1.36}$	$62.12_{\pm 0.42}$	$53.94_{\pm 1.63}$	$31.96_{\pm 1.82}$	$55.57_{\pm 0.42}$	$58.95_{\pm 1.15}$
EWC [29]	$42.25_{\pm 4.02}$	$63.55_{\pm 1.17}$	$51.20_{\pm 4.79}$	$\underline{33.92}_{\pm 1.82}$	$56.39_{\pm 0.23}$	$56.68_{\pm 2.34}$
SI [61]	$41.78_{\pm 1.64}$	$62.76_{\pm 1.10}$	$52.17_{\pm 1.63}$	$33.85_{\pm 0.64}$	$55.29_{\pm 0.27}$	$56.59_{\pm 0.77}$
MAS [3]	$40.32_{\pm 0.93}$	$62.31_{\pm 0.52}$	$53.87_{\pm 1.44}$	$33.39_{\pm 0.79}$	$55.67_{\pm 0.47}$	$57.60_{\pm 1.10}$
LwF-AC [33]	$\underline{43.18}_{\pm 1.57}$	$\underline{65.97}_{\pm 0.45}$	$\underline{47.66}_{\pm 1.81}$	$33.00_{\pm 1.79}$	$\underline{58.43}_{\pm 0.92}$	$\underline{54.33}_{\pm 1.95}$
Inf-SSM	$49.34_{\pm 3.36}$	$67.51_{\pm 1.47}$	$25.14_{\pm 3.86}$	$43.82_{\pm 1.55}$	$62.82_{\pm 1.29}$	$36.34_{\pm 1.54}$

As shown in Tab. 6, Inf-SSM achieves superior performance across both benchmarks. Compared to the best baseline, Inf-SSM reduces FM by 47.25% and 33.11%, while improving AA by 14.27% and 29.19% for 5- and 10-task settings, respectively. These results demonstrate that, under fair comparisons, Inf-SSM outperforms foundational EFCIL methods owing to its ability to capture the underlying geometry of Linear-Input-Varying SSMs.

H. Additional studies

H.1. Centered Kernel Disparity states analysis

Since the evolution of SSM internal states over the task sequence is not well understood, we first analyze their structural changes across EFCIL tasks using similarity measures. A prominent choice is Centered Kernel Alignment (CKA) [30].

$$\mathrm{CKA}(\mathbf{W}_1, \mathbf{W}_2) = \frac{\mathrm{HSIC}(\mathbf{W}_1, \mathbf{W}_2)}{\sqrt{\mathrm{HSIC}(\mathbf{W}_1, W_1)\mathrm{HSIC}(\mathbf{W}_2, \mathbf{W}_2)}}. \tag{37}$$

where HSIC is Hilbert-Schmidt Independence Criterion [16]. To achieve positive correlation with forgetting, we redefined the CKA of CL settings as Centered Kernel Disparity (CKD), where

$$\mathrm{CKD}(\mathbf{W}_1, \mathbf{W}_2) = 1 - \mathrm{CKA}(\mathbf{W}_1, \mathbf{W}_2). \tag{38}$$

Instead of measuring weights as commonly used, CKD will be utilized to analyze state evolution in CL settings across sequential tasks and across different SSM layers in VIM.

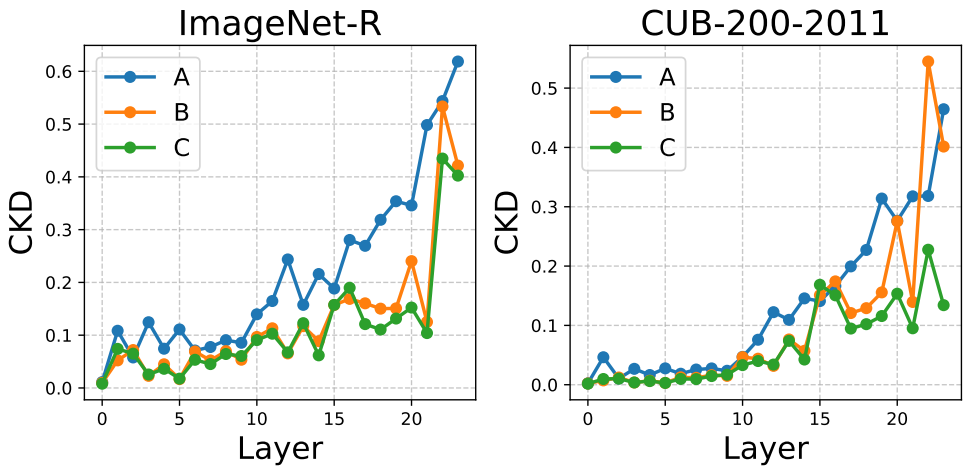

Figure 4. CKD analysis on Vim-small with ImageNet-R and CUB-200-2011 over 10 tasks, EFCIL settings for each of the 24 layers of SSM blocks.

Fig. 4 shows that SSM state changes most at the last few layers. However, regularization is applied across all layers as we deem that small changes in activation in unregulated early layers will be amplified across the subsequent layers and lead to large CF.

H.2. Inf-SSM ablation studies

In this study, we aim to investigate whether applying Inf-SSM to all Vim blocks is necessary. As discussed in §H.1, the SSM state matrices in the final Vim block undergo significant changes compared to the earlier blocks. Thus, in this ablation study, we apply Inf-SSM to different numbers of blocks in Vim-small. As presented in Tab. 7, AIA and AA improve with the number of blocks regularized. Interestingly, FM increases with the number of blocks regularized, as we expect the model stability to improve instead of degrading when more blocks are applied with Inf-SSM. This observation can be explained

Table 7. Ablation study on how many layers of Inf-SSM need to be applied in 10 tasks ImageNet-R and CUB-200-2011.

Vim Block applied	ImageNet-R			CUB-200-2011		
	AIA	AA	FM	AIA	AA	FM
Final Block	$60.69_{\pm 2.80}$	$42.60_{\pm 5.53}$	$23.18_{\pm 4.70}$	$28.03_{\pm 1.36}$	$11.04_{\pm 1.36}$	$1.19_{\pm 1.39}$
Final 12 Blocks	$62.66_{\pm 1.78}$	$43.24_{\pm 2.74}$	$36.16_{\pm 2.98}$	$40.16_{\pm 1.38}$	$17.56_{\pm 2.13}$	$24.20_{\pm 2.45}$
All 24 Blocks	$62.82_{\pm 1.29}$	$43.82_{\pm 1.55}$	$36.34_{\pm 1.54}$	$42.11_{\pm 0.83}$	$18.97_{\pm 0.90}$	$45.88_{\pm 0.50}$

by the fact that regularizing fewer layers means that earlier layers tend to over-regularize to minimize Inf-SSM loss in the regularized layers, leading to lower overall plasticity in the model. Thus, as the initial accuracy when a new task is learned is low, the final FM decreases as the number of blocks regularized decreases.

H.3. Simplified distance performance of Inf-SSM

In this part, Inf-SSM distance as derived in Eq. (33) is compared to other distances on the Grassmannian manifold in terms of speed. We randomly sampled $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{C} \in \mathbb{R}^{1 \times n}$ matrices from a Gaussian distribution with $n = 100$. The experiment is conducted on a single A40 40GB VRAM GPU with 10000 Monte Carlo simulation iterations. Four distances are compared: Inf-SSM, Fubini-Study [60], Martin [27, 49], Binet-Cauchy [60]. As shown in Tab. 8, Inf-SSM decreases

Table 8. Monte Carlo simulation of 10000 iterations on a single A40 GPU with $\mathbf{A} \in \mathbb{R}^{100 \times 100}$ and $\mathbf{C} \in \mathbb{R}^{1 \times 100}$ for comparisons of computational time between Inf-SSM and other distance metrics in §I

Metric	t (s)	±Std. Dev. (s)
Martin	0.0439	$8.3e{-3}$
Fubini-Study	0.0422	$5.9e{-3}$
Binet-Cauchy	0.0430	$5.9e{-3}$
Inf-SSM	0.0004	$4.5e{-3}$

computational time by 99.05% when compared to Fubini-Study distance with 23.73% less standard deviation. The benefits of computational speed will be even more significant when taking into account backpropagation during training of Vim, as computation of the gradient over determinant or inverse operations is costly.

H.4. Inf-SSM+

Inf-SSM, as shown in Eq. (15) ignores one important aspect of SSM, which is the input mapping defined by $\mathbf{B}$. This creates an unregularized path in the SSM algorithm and might lead to CF. Thus, we include $\mathbf{B}$[4] in the loss by adding a Frobenius norm regularization term. We define this variant of Inf-SSM as Inf-SSM+, more specifically:

$$\mathrm{L}_{\texttt{ISM+}} = \mathrm{L}_{\texttt{ISM}} + \gamma \mathbb{E}_{\mathcal{D}_T} \left\| \mathbf{B}_{T-1} - \mathbf{B}_T \right\|_{\mathrm{F}}^2 . \tag{39}$$

where γ controls the regularization strength on $\mathbf{B}$.

Table 9. AA(% ↑), AIA(% ↑), and FM(% ↓) of Inf-SSM and Inf-SSM+ on ImageNet-R, CIFAR-100, and Caltech-256 over 5-Tasks and 10-Tasks scenario on Vim-small.

Method	ImageNet-R			CIFAR-100			Caltech-256		
	$\mathrm{AA}_{\pm\mathrm{std}}$	$\mathrm{AIA}_{\pm\mathrm{std}}$	$\mathrm{FM}_{\pm\mathrm{std}}$	$\mathrm{AA}_{\pm\mathrm{std}}$	$\mathrm{AIA}_{\pm\mathrm{std}}$	$\mathrm{FM}_{\pm\mathrm{std}}$	$\mathrm{AA}_{\pm\mathrm{std}}$	$\mathrm{AIA}_{\pm\mathrm{std}}$	$\mathrm{FM}_{\pm\mathrm{std}}$
5-Tasks Scenario									
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$_{\pm 3.61}$
Inf-SSM+	47.43$_{\pm 6.57}$	66.36$_{\pm 3.12}$	27.81$_{\pm 8.86}$	46.87$_{\pm 2.85}$	68.04$_{\pm 2.02}$	31.08$_{\pm 3.64}$	51.10$_{\pm 2.68}$	68.17$_{\pm 0.98}$	49.44$_{\pm 3.27}$
10-Tasks Scenario									
Method	ImageNet-R			CIFAR-100			Caltech-256		
	$\mathrm{AA}_{\pm\mathrm{std}}$	$\mathrm{AIA}_{\pm\mathrm{std}}$	$\mathrm{FM}_{\pm\mathrm{std}}$	$\mathrm{AA}_{\pm\mathrm{std}}$	$\mathrm{AIA}_{\pm\mathrm{std}}$	$\mathrm{FM}_{\pm\mathrm{std}}$	$\mathrm{AA}_{\pm\mathrm{std}}$	$\mathrm{AIA}_{\pm\mathrm{std}}$	$\mathrm{FM}_{\pm\mathrm{std}}$
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$_{\pm 4.05}$
Inf-SSM+	43.62$_{\pm 1.48}$	63.29$_{\pm 0.74}$	37.69$_{\pm 1.49}$	24.70$_{\pm 3.33}$	53.38$_{\pm 1.82}$	23.85$_{\pm 2.26}$	39.24$_{\pm 1.85}$	62.33$_{\pm 1.96}$	56.50$_{\pm 3.07}$

Averaged over both task splits and all datasets, Inf-SSM+ only achieves a negligible AIA gain of 0.04% and a small FM reduction of 0.19% over Inf-SSM. Meanwhile, Inf-SSM+ incurs a 1.40% drop in AA when compared to Inf-SSM. Overall, explicitly regularizing $\mathbf{B}$ offers limited benefit while increasing the computational cost.

[4]Note for readability, $\mathbf{B}$ refers to $\tilde{\mathbf{B}}$ derived in §D.3.

H.5. Vim-tiny

To validate that Inf-SSM is adaptable to different model sizes, especially on small models, we validated Inf-SSM along with EFCIL baselines of EWC, SI, MAS, and LwF on Vim-tiny. Vim-tiny only has 7M parameters in comparison to 26M of Vim-small [65].

Table 10. AA(% ↑), AIA(% ↑), and FM(% ↓) of EFCIL methods on ImageNet-R 5-Task and 10-Task scenario on **Vim-tiny**. **Note:** Regularization focus is on parameter sets ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$) among all methods **except Inf-SSM**. Second-best results are underlined.

Method	ImageNet-R 5-task			ImageNet-R 10-task		
	$AA_{\pm std}$	$AIA_{\pm std}$	$FM_{\pm std}$	$AA_{\pm std}$	$AIA_{\pm std}$	$FM_{\pm std}$
Seq	$25.68_{\pm 1.47}$	$50.28_{\pm 1.35}$	$63.05_{\pm 2.45}$	$17.58_{\pm 1.34}$	$40.07_{\pm 1.21}$	$63.00_{\pm 1.35}$
EWC [29]	$35.73_{\pm 1.94}$	$57.62_{\pm 1.27}$	$48.34_{\pm 1.56}$	$23.14_{\pm 3.17}$	$47.38_{\pm 1.49}$	$55.45_{\pm 3.24}$
SI [61]	$25.55_{\pm 0.60}$	$50.05_{\pm 1.01}$	$63.38_{\pm 0.35}$	$17.15_{\pm 1.22}$	$39.41_{\pm 1.26}$	$64.65_{\pm 1.27}$
MAS [3]	$27.94_{\pm 0.85}$	$52.47_{\pm 1.24}$	$60.15_{\pm 0.38}$	$22.41_{\pm 1.48}$	$43.43_{\pm 1.19}$	$60.23_{\pm 1.76}$
LwF-ABC [33]	$37.17_{\pm 1.99}$	$58.87_{\pm 1.85}$	$35.10_{\pm 2.30}$	$26.17_{\pm 2.33}$	$49.25_{\pm 1.23}$	$32.25_{\pm 5.68}$
Inf-SSM	$39.85_{\pm 1.15}$	$58.74_{\pm 1.06}$	$23.33_{\pm 1.24}$	$27.92_{\pm 1.34}$	$49.25_{\pm 1.07}$	$25.18_{\pm 1.55}$

On average, Inf-SSM outperforms the previous best by 6.95% for AA and reduces FM by 27.74%. These results are consistent with our experiments on Vim-small, where Inf-SSM is particularly effective at retaining past knowledge, and its advantage grows as the number of learned tasks increases. Although LwF-ABC attains a slightly higher AIA than Inf-SSM, as reported in Tab. 10, the gap is marginal and well within the standard deviation.

H.6. Additional EFCIL Baseline

For an even more comprehensive empirical validation, we have adapted and re-implemented the more recent uncertainty-based method UCL [2] in Vim-small. The comparison below shows that Inf-SSM consistently outperforms UCL across both 5-task and 10-task settings for ImageNet-R, CIFAR-100, and Caltech-256.

Table 11. AA(% ↑), AIA(% ↑), and FM(% ↓) 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 ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$) 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]	$45.72_{\pm 3.04}$	$65.18_{\pm 1.48}$	$47.21_{\pm 3.34}$	$37.38_{\pm 1.40}$	$61.78_{\pm 1.05}$	$53.04_{\pm 1.35}$	$47.57_{\pm 0.21}$	$65.29_{\pm 1.04}$	$57.88_{\pm 0.43}$
MAS [3]	$44.70_{\pm 2.77}$	$65.59_{\pm 0.79}$	$48.23_{\pm 2.92}$	$37.59_{\pm 1.44}$	$61.95_{\pm 1.18}$	$53.13_{\pm 1.81}$	$44.87_{\pm 1.19}$	$66.44_{\pm 1.07}$	$61.00_{\pm 1.53}$
UCL [2]	$48.03_{\pm 1.15}$	$67.19_{\pm 0.22}$	$43.90_{\pm 1.29}$	$39.48_{\pm 4.95}$	$62.62_{\pm 2.89}$	$46.52_{\pm 7.15}$	$44.74_{\pm 2.66}$	$65.17_{\pm 0.42}$	$62.48_{\pm 3.80}$
LwF-ABC [33]	$45.09_{\pm 6.58}$	$65.69_{\pm 3.17}$	$40.77_{\pm 8.26}$	$44.62_{\pm 2.67}$	$66.81_{\pm 1.41}$	$38.68_{\pm 3.77}$	$46.52_{\pm 2.66}$	$66.58_{\pm 1.08}$	$59.03_{\pm 3.43}$
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]	$41.99_{\pm 2.28}$	$61.78_{\pm 2.24}$	$49.02_{\pm 2.10}$	$22.20_{\pm 1.11}$	$53.46_{\pm 0.35}$	$63.92_{\pm 1.28}$	$28.35_{\pm 0.34}$	$56.64_{\pm 0.74}$	$72.73_{\pm 0.47}$
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]	$40.10_{\pm 1.48}$	$61.30_{\pm 0.64}$	$48.18_{\pm 1.84}$	$20.44_{\pm 1.60}$	$49.69_{\pm 1.55}$	$37.99_{\pm 1.01}$	$28.15_{\pm 0.79}$	$55.25_{\pm 0.70}$	$73.50_{\pm 0.75}$
UCL [2]	$40.10_{\pm 1.87}$	$60.45_{\pm 1.01}$	$48.84_{\pm 1.94}$	$21.71_{\pm 1.19}$	$50.35_{\pm 0.50}$	$29.16_{\pm 1.00}$	$33.16_{\pm 3.43}$	$57.95_{\pm 1.77}$	$69.07_{\pm 4.05}$
LwF-ABC [33]	$41.85_{\pm 0.82}$	$62.63_{\pm 0.92}$	$40.10_{\pm 0.61}$	$24.39_{\pm 3.25}$	$53.48_{\pm 1.49}$	$25.29_{\pm 2.81}$	$35.45_{\pm 1.16}$	$59.63_{\pm 0.74}$	$64.32_{\pm 1.41}$
Inf-SSM	$43.82_{\pm 1.55}$	$62.82_{\pm 1.29}$	$36.34_{\pm 1.54}$	$26.53_{\pm 3.10}$	$54.24_{\pm 1.87}$	$24.00_{\pm 1.80}$	$39.88_{\pm 2.43}$	$62.28_{\pm 2.33}$	$55.85_{\pm 4.05}$

As shown in Tab. 11, Inf-SSM outperforms UCL in all metric instances. On average, Inf-SSM outperforms UCL [2] by 13.72% in AA, 5.00% in AIA, and 24.43% in FM.

I. Distance Equivalence on the Grassmannian

In this section, we consider two infinite observability subspaces of an SSM, $\mathcal{S}_1$ and $\mathcal{S}_2$, with principal angles $\theta_1, \ldots, \theta_n$ between them. The chordal distance [12] is

$$d_{\text{chord}}(\mathcal{S}_1, \mathcal{S}_2) = \sqrt{\sum_{i=1}^{n} \sin^2 \theta_i}.$$

Although the distances considered below are distinct metrics on the Grassmannian, they are locally equivalent near $\mathcal{S}_1 = \mathcal{S}_2$ because they share the same second-order behavior in the principal angles. To show that the Binet–Cauchy, Fubini–Study, and Martin distances are all equivalent to the chordal distance as $\mathcal{S}_2 \to \mathcal{S}_1$, it suffices to evaluate

$$\lim_{\mathcal{S}_2 \to \mathcal{S}_1} \frac{d_{\text{Gr}}^2}{d_{\text{chord}}^2}, \qquad d_{\text{Gr}} \in \{d_{\text{Binet}}, d_{\text{Fubini}}, d_{\text{Martin}}\}.$$

Let $\theta = (\theta_1, \ldots, \theta_n)$. Since the Taylor expansion of $\sin^2 x$ (Maclaurin series centered at $x = 0$) is

$$\sin^2 x = x^2 + \mathcal{O}(x^4) \quad \text{as } x \to 0,$$

we have

$$d_{\text{chord}}^2 = \sum_{i=1}^{n} \sin^2 \theta_i = \sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4). \tag{40}$$

Lemma I.1 (Binet–Cauchy distance and chordal distance equivalence)**.** *The Binet–Cauchy distance [60] is*

$$d_{\text{Binet}}(\mathcal{S}_1, \mathcal{S}_2) = \sqrt{1 - \prod_{i=1}^{n} \cos^2(\theta_i)},$$

and d_{Binet} is equivalent to d_{chord} as $\mathcal{S}_2 \to \mathcal{S}_1$.

Proof. As $\mathcal{S}_2 \to \mathcal{S}_1$, we have $\theta_i \to 0$ for all i, so it is enough to evaluate

$$\lim_{\theta \to 0} \frac{1 - \prod_{i=1}^{n} \cos^2(\theta_i)}{\sum_{i=1}^{n} \sin^2 \theta_i}.$$

Using

$$\cos^2 x = 1 - x^2 + \mathcal{O}(x^4),$$

we obtain

$$\begin{aligned} \prod_{i=1}^{n} \cos^2(\theta_i) &= \prod_{i=1}^{n} \left(1 - \theta_i^2 + \mathcal{O}(\theta_i^4)\right) \\ &= 1 - \sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4). \end{aligned}$$

Therefore,

$$1 - \prod_{i=1}^{n} \cos^2(\theta_i) = \sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4).$$

Combining this with Eq. (40),

$$\lim_{\theta \to 0} \frac{d_{\text{Binet}}^2}{d_{\text{chord}}^2} = \lim_{\theta \to 0} \frac{\sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4)}{\sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4)} = 1.$$

Hence, d_{Binet} and d_{chord} are locally equivalent. □

Lemma I.2 (Fubini–Study distance and chordal distance equivalence)**.** *The Fubini–Study distance [60] is*

$$d_{\text{Fubini}}(\mathcal{S}_1, \mathcal{S}_2) = \cos^{-1}\left(\prod_{i=1}^{n} \cos(\theta_i)\right),$$

and d_{Fubini} *is equivalent to* d_{chord} *as* $\mathcal{S}_2 \to \mathcal{S}_1$.

Proof. As $\mathcal{S}_2 \to \mathcal{S}_1$, we evaluate

$$\lim_{\theta \to 0} \frac{\left[\cos^{-1}(\prod_{i=1}^{n} \cos(\theta_i))\right]^2}{\sum_{i=1}^{n} \sin^2 \theta_i}.$$

Using the Taylor series expansion (Maclaurin series centered at $x = 0$),

$$\cos x = 1 - \frac{x^2}{2} + \mathcal{O}(x^4),$$

we have

$$\begin{aligned}\prod_{i=1}^{n} \cos(\theta_i) &= \prod_{i=1}^{n} \left(1 - \frac{\theta_i^2}{2} + \mathcal{O}(\theta_i^4)\right) \\ &= 1 - \frac{1}{2}\sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4).\end{aligned}$$

Let

$$\delta = 1 - \prod_{i=1}^{n} \cos(\theta_i) = \frac{1}{2}\sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4).$$

Since

$$\cos^{-1}(1 - \delta) = \sqrt{2\delta} + \mathcal{O}(\delta^{3/2}),$$

, which is derived from the Puiseux series, it follows that

$$\left[\cos^{-1}\left(\prod_{i=1}^{n} \cos(\theta_i)\right)\right]^2 = 2\delta + \mathcal{O}(\delta^2) = \sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4). \tag{41}$$

Using Eqs. (40) and (41), we conclude that

$$\lim_{\theta \to 0} \frac{d_{\text{Fubini}}^2}{d_{\text{chord}}^2} = \lim_{\theta \to 0} \frac{\sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4)}{\sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4)} = 1.$$

Hence, d_{Fubini} and d_{chord} are locally equivalent. □

Lemma I.3 (Martin distance and chordal distance equivalence)**.** *The Martin distance [9, 43] is*

$$d_{\mathrm{Martin}}(\mathcal{S}_1, \mathcal{S}_2) = \sqrt{-\log \prod_{i=1}^{n} \cos^2 \theta_i},$$

and d_{Martin} *is equivalent to* d_{chord} *as* $\mathcal{S}_2 \to \mathcal{S}_1$.

Proof. As $\mathcal{S}_2 \to \mathcal{S}_1$, we evaluate

$$\lim_{\theta \to 0} \frac{-\log \prod_{i=1}^{n} \cos^2 \theta_i}{\sum_{i=1}^{n} \sin^2 \theta_i}.$$

Using

$$\log \prod_{i=1}^{n} \cos^2 \theta_i = \sum_{i=1}^{n} \log(\cos^2 \theta_i),$$

together with Taylor series expansion of $\cos^2 x$ (Maclaurin series at $x = 0$),

$$\log(\cos^2 x) = -x^2 + \mathcal{O}(x^4) \quad \text{as } x \to 0,$$

we obtain

$$-\log \prod_{i=1}^{n} \cos^2 \theta_i = \sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4). \tag{42}$$

Combining Eqs. (40) and (42), we get

$$\lim_{\theta \to 0} \frac{d_{\mathrm{Martin}}^2}{d_{\mathrm{chord}}^2} = \lim_{\theta \to 0} \frac{\sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4)}{\sum_{i=1}^{n} \theta_i^2 + \mathcal{O}(\|\theta\|^4)} = 1.$$

Hence, d_{Martin} and d_{chord} are locally equivalent. □

J. Additional implementation details

J.1. Baselines

Our objective is to evaluate Inf-SSM under a continual learning (CL) protocol that does not rely on CNN- or Transformer-specific architectural assumptions, so that both Inf-SSM and all baselines can be instantiated fairly on the Vim backbone. Accordingly, we require baselines that:

1. Represent the main CL paradigms used in CIL or EFCIL.
2. Can be adapted to the Vim backbone without ad hoc, architecture-specific redesign.

We therefore adopt the following baselines, all implemented on Vim-Small under a shared training protocol (datasets, task splits, and metrics are described in §5, §G.2, and §J.2).

For replay-based methods:

- **ER** [51]is a canonical replay-based baseline. It measures how much Inf-SSM can further reduce forgetting when explicit rehearsal is allowed.
- **LUCIR** [24] and **X-DER** [5] are strong hybrid methods that combine replay with regularization or contrastive mechanisms. They are widely used in CIL evaluations and indicate whether Inf-SSM still brings gains on top of competitive replay-regularization pipelines.
- **L2P-R** [59] is a prompt-based method adapted to Vim-Small (see §J.3) to test compatibility of Inf-SSM with token-level adaptation strategies in SSM architectures.
- **CLFD** [37] is a frequency-domain method representing the latest CL designs. We include it to show Inf-SSM's benefit even when features are transformed into alternative domains.

For EFCIL methods:

- **EWC** [29] serves as a canonical example of sensitivity-based regularization.
- **SI** [61] serves as a baseline for the synaptic-level importance approach.
- **MAS** [3] offers comparison against methods based on Hebbian learning theory.
- **LwF** [33] is adapted to distill SSM state using the Frobenius norm applied explicitly on $(\mathbf{A}, \mathbf{C})$ for LwF-AC and $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ for LwF-ABC. LwF serves as a cornerstone distillation-based method that explicitly regularizes SSM states via the Frobenius norm and provides strong evidence for the importance of a $\mathbf{P}$-equivalence-aware distance measure in SSMs.

Together, these baselines span a wide range of CL families and are heavily used in prior CIL or EFCIL literature. We intentionally exclude complex hybrid or prototype-based EFCIL methods in our isolation test because they either violate the EFCIL assumptions (e.g., by storing prototypes or intermediate features) or require heavy, architecture-specific modifications that are not directly compatible with a clean Vim-SSM instantiation. More in-depth discussions on several such methods are included in §F.1.

Regarding Mamba-based CL methods like MambaCL [63], Mamba-CL [7], and Mamba-FSCIL [32] (see §F.2 for details), these works focus on scenario-specific goals (e.g., online CL, few-shot class-incremental learning, task-conditional adaptation) and do not exploit the rich geometrical structures of SSMs. We thus view them as future integration targets that are orthogonal to our research question, and not direct baselines for validating our core claim.

Summary. Our baseline set is chosen to cover prominent CL methods under a unified SSM backbone and evaluation protocol, while avoiding methods whose assumptions (stored features, heavily modified architectures, or different CL scenarios) are misaligned with the problem setting and goals of our work. Under these representative baselines, Inf-SSM consistently reduces forgetting and improves accuracy, supporting our claim that geometry-aware observability regularization is an effective and broadly compatible CL regularization algorithm.

J.2. Hyperparameters and Compute

For all experiments, we run on seeds 0, 10, and 100 with the same set of hyperparameters as shown below. All RGB images are resized to 224×224 before training and evaluations. For all datasets, we split each into 5 and 10 sequential tasks, where each task has an equal number of classes sampled from the corresponding datasets. For backbone, we utilized Vim-small [65] and kept all the hyperparameters in Vim as default unless mentioned otherwise.

The experiments are conducted on various machines available, which are A5500 GPU, A40 GPU, A100 GPU, and H100 GPU. All experiments are conducted on a single GPU only without distributed training.

Table 12. Hyperparameters for VIM-Small model on 5-task continual learning benchmark. All regularization coefficients (λ) are shown in base units. Learning rates (LR) follow cosine decay schedules.

Hyperparameter	**ImageNet-R**	**CIFAR-100**	**Caltech-256**
Batch Size	128	128	128
Training Epochs	40	40	40
Base LR	5.00×10^{-4}	1.00×10^{-5}	1.00×10^{-4}
Warmup LR	1.00×10^{-4}	1.00×10^{-6}	1.00×10^{-4}
Minimum LR	1.00×10^{-5}	1.00×10^{-7}	1.00×10^{-5}
Task LR Scaling	2.50×10^{-1}	5.00×10^{-1}	5.00×10^{-1}
Weight Decay	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
EWC-E-λ	2.00×10^{2}	2.50×10^{3}	1.00×10^{4}
EWC-γ	7.50×10^{-1}	7.50×10^{-1}	7.50×10^{-1}
SI-C	1.00×10^{3}	1.00×10^{5}	5.00×10^{4}
SI-ξ	9.00×10^{-1}	9.00×10^{-1}	9.00×10^{-1}
MAS-λ	1.00×10^{1}	1.00×10^{1}	1.00×10^{2}
MSE-γ	1.00×10^{2}	5.00×10^{1}	1.00×10^{2}
MSE-λ	5.00×10^{2}	2.50×10^{2}	5.00×10^{2}
Inf-SSM-λ	1.00×10^{6}	2.00×10^{5}	2.50×10^{6}
Inf-SSM+ -λ	1.00×10^{6}	2.00×10^{3}	2.50×10^{6}
Inf-SSM+ -γ	1.00×10^{0}	1.00×10^{2}	1.00×10^{2}

Table 13. Hyperparameters for VIM-Small model on 10-task continual learning benchmark. Configuration follows same conventions as Table 12.

Hyperparameter	**ImageNet-R**	**CIFAR-100**	**Caltech-256**
Batch Size	128	128	128
Training Epochs	40	40	40
Base LR	5.00×10^{-4}	1.00×10^{-5}	1.00×10^{-4}
Warmup LR	1.00×10^{-4}	1.00×10^{-6}	1.00×10^{-5}
Minimum LR	1.00×10^{-5}	1.00×10^{-7}	1.00×10^{-5}
Task LR Scaling	2.50×10^{-1}	5.00×10^{-1}	5.00×10^{-1}
Weight Decay	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
EWC-λ	5.00×10^{2}	1.00×10^{4}	1.00×10^{3}
EWC-γ	7.50×10^{-1}	7.50×10^{-1}	7.50×10^{-1}
SI-c	5.00×10^{4}	5.00×10^{4}	5.00×10^{4}
SI-ξ	9.00×10^{-1}	8.00×10^{-1}	9.00×10^{-1}
MAS-λ	1.00×10^{2}	1.00×10^{1}	1.00×10^{2}
MSE-γ	1.00×10^{2}	1.00×10^{1}	5.00×10^{2}
MSE-λ	5.00×10^{2}	5.00×10^{1}	5.00×10^{2}
Inf-SSM-λ	2.50×10^{5}	3.00×10^{4}	2.50×10^{6}
Inf-SSM+ -λ	1.50×10^{5}	2.00×10^{2}	1.00×10^{4}
Inf-SSM+ -γ	1.00×10^{2}	1.00×10^{1}	1.00×10^{1}

J.3. L2P in SSM

Adapting prompt-based methods in SSM is not straightforward. This is due to the recurrent nature of SSMs, leading to prompt tokens positioning being sensitive, unlike in attention layers. In our L2P-R implementation, we insert the prompt tokens by concatenating them at the start of the patch sequence. For classification, we have frozen the [CLS] token and instead utilize the prompt tokens for downstream classification by the final linear layer.

Table 14. Hyperparameters for ER, LUCIR, X-DER, L2P-R, and CLFD methods integration tests with Inf-SSM for ImageNet-R 5 tasks setting. Configuration includes both general training settings and method-specific parameters.

Hyperparameter	**ER**	**LUCIR**	**X-DER**	**L2P-R**	**CLFD**
Batch Size	128	128	32	32	64
Training Epochs	40	40	20	20	20
Base LR	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}
Warmup LR	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}
Minimum LR	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-5}	1.00×10^{-5}
Task LR Scaling	2.50×10^{-1}	2.50×10^{-1}	2.50×10^{-1}	2.50×10^{-1}	2.50×10^{-1}
Weight Decay	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
Buffer Size	5.00×10^{2}	5.00×10^{2}	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}
LUCIR-λ_{base}	–	5.00×10^{-1}	–	–	–
LUCIR-λ_{MR}	–	1.00×10^{-1}	–	–	–
LUCIR-K_{MR}	–	2.00×10^{0}	–	–	–
LUCIR-MR Margin	–	5.00×10^{-2}	–	–	–
X-DER-γ	–	–	8.50×10^{-1}	–	–
X-DER-Temp	–	–	7.00×10^{-2}	–	–
X-DER-Base Temp	–	–	7.00×10^{-2}	–	–
X-DER-α	–	–	3.00×10^{-1}	–	–
X-DER-β	–	–	1.80×10^{0}	–	–
X-DER-SimCLR Batch Size	–	–	3.20×10^{1}	–	–
X-DER-SimCLR Num Augs	–	–	2.00×10^{0}	–	–
X-DER-λ	–	–	5.00×10^{-2}	–	–
X-DER-dp Weight	–	–	1.00×10^{-1}	–	–
X-DER-Contr Margin	–	–	3.00×10^{-1}	–	–
X-DER-Constr η	–	–	1.00×10^{-1}	–	–
X-DER-Future Constr	–	–	1.00×10^{0}	–	–
X-DER-Part Constr	–	–	0.00×10^{0}	–	–
L2P-R-Pull-Constr	–	–	–	1.00×10^{-1}	–
Inf-SSM-λ	1.50×10^{5}	5.00×10^{2}	5.00×10^{2}	5.00×10^{3}	5.00×10^{3}

Table 15. Hyperparameters for ER, LUCIR, X-DER, L2P-R, and CLFD methods integration tests with Inf-SSM for ImageNet-R 10 tasks setting. Configuration includes both general training settings and method-specific parameters.

Hyperparameter	**ER**	**LUCIR**	**X-DER**	**L2P-R**	**CLFD**
Batch Size	128	128	32	32	64
Training Epochs	40	40	20	20	20
Base LR	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}
Warmup LR	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-5}	1.00×10^{-4}
Minimum LR	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-5}	1.00×10^{-5}
Task LR Scaling	2.50×10^{-1}	2.50×10^{-1}	2.50×10^{-1}	5.00×10^{-1}	2.50×10^{-1}
Weight Decay	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
Buffer Size	5.00×10^{2}	5.00×10^{2}	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}
LUCIR-λ_{base}	–	5.00×10^{-1}	–	–	–
LUCIR-λ_{MR}	–	1.00×10^{-1}	–	–	–
LUCIR-K_{MR}	–	2.00×10^{0}	–	–	–
LUCIR-MR Margin	–	5.00×10^{-2}	–	–	–
X-DER-γ	–	–	8.50×10^{-1}	–	–
X-DER-Temp	–	–	7.00×10^{-2}	–	–
X-DER-Base Temp	–	–	7.00×10^{-2}	–	–
X-DER-α	–	–	3.00×10^{-1}	–	–
X-DER-β	–	–	1.80×10^{0}	–	–
X-DER-SimCLR Batch Size	–	–	3.20×10^{1}	–	–
X-DER-SimCLR Num Augs	–	–	2.00×10^{0}	–	–
X-DER-λ	–	–	5.00×10^{-2}	–	–
X-DER-dp Weight	–	–	1.00×10^{-1}	–	–
X-DER-Contr Margin	–	–	3.00×10^{-1}	–	–
X-DER-Constr η	–	–	1.00×10^{-1}	–	–
X-DER-Future Constr	–	–	1.00×10^{0}	–	–
X-DER-Part Constr	–	–	0.00×10^{0}	–	–
L2P-R-Pull-Constr	–	–	–	1.00×10^{-1}	–
Inf-SSM-λ	1.50×10^{5}	1.00×10^{3}	5.00×10^{2}	5.00×10^{4}	1.00×10^{4}

Table 16. Hyperparameters for ER, LUCIR, X-DER, L2P-R, and CLFD methods integration tests with Inf-SSM for CIFAR-100 5 tasks setting. Configuration includes both general training settings and method-specific parameters.

Hyperparameter	**ER**	**LUCIR**	**X-DER**	**L2P-R**	**CLFD**
Batch Size	128	64	32	32	64
Training Epochs	40	40	20	20	20
Base LR	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}	1.00×10^{-4}
Warmup LR	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	5.00×10^{-5}
Minimum LR	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}
Task LR Scaling	2.50×10^{-1}	2.50×10^{-1}	2.50×10^{-1}	2.50×10^{-1}	5.00×10^{-1}
Weight Decay	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
Buffer Size	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}	2.00×10^{3}	1.00×10^{3}
LUCIR-λ_{base}	–	5.00×10^{-1}	–	–	–
LUCIR-λ_{MR}	–	1.00×10^{-1}	–	–	–
LUCIR-K_{MR}	–	2.00×10^{0}	–	–	–
LUCIR-MR Margin	–	5.00×10^{-2}	–	–	–
X-DER-γ	–	–	8.50×10^{-1}	–	–
X-DER-Temp	–	–	7.00×10^{-2}	–	–
X-DER-Base Temp	–	–	7.00×10^{-2}	–	–
X-DER-α	–	–	3.00×10^{-1}	–	–
X-DER-β	–	–	1.80×10^{0}	–	–
X-DER-SimCLR Batch Size	–	–	3.20×10^{1}	–	–
X-DER-SimCLR Num Augs	–	–	2.00×10^{0}	–	–
X-DER-λ	–	–	5.00×10^{-2}	–	–
X-DER-dp Weight	–	–	1.00×10^{-1}	–	–
X-DER-Contr Margin	–	–	3.00×10^{-1}	–	–
X-DER-Constr η	–	–	1.00×10^{-1}	–	–
X-DER-Future Constr	–	–	1.00×10^{0}	–	–
X-DER-Part Constr	–	–	0.00×10^{0}	–	–
L2P-R-Pull-Constr	–	–	–	1.00×10^{-1}	–
Inf-SSM-λ	1.00×10^{3}	1.00×10^{3}	5.00×10^{2}	1.00×10^{4}	1.00×10^{4}

Table 17. Hyperparameters for ER, LUCIR, X-DER, L2P-R, and CLFD methods integration tests with Inf-SSM for CIFAR-100 10 tasks setting. Configuration includes both general training settings and method-specific parameters.

Hyperparameter	**ER**	**LUCIR**	**X-DER**	**L2P-R**	**CLFD**
Batch Size	128	64	32	32	64
Training Epochs	40	40	20	40	20
Base LR	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}	5.00×10^{-4}	1.00×10^{-4}
Warmup LR	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	5.00×10^{-5}
Minimum LR	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}
Task LR Scaling	2.50×10^{-1}	2.50×10^{-1}	2.50×10^{-1}	2.50×10^{-1}	5.00×10^{-1}
Weight Decay	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
Buffer Size	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}	2.00×10^{3}	1.00×10^{3}
LUCIR-λ_{base}	–	5.00×10^{-1}	–	–	–
LUCIR-λ_{MR}	–	1.00×10^{-1}	–	–	–
LUCIR-K_{MR}	–	2.00×10^{0}	–	–	–
LUCIR-MR Margin	–	5.00×10^{-2}	–	–	–
X-DER-γ	–	–	8.50×10^{-1}	–	–
X-DER-Temp	–	–	7.00×10^{-2}	–	–
X-DER-Base Temp	–	–	7.00×10^{-2}	–	–
X-DER-α	–	–	3.00×10^{-1}	–	–
X-DER-β	–	–	1.80×10^{0}	–	–
X-DER-SimCLR Batch Size	–	–	3.20×10^{1}	–	–
X-DER-SimCLR Num Augs	–	–	2.00×10^{0}	–	–
X-DER-λ	–	–	5.00×10^{-2}	–	–
X-DER-dp Weight	–	–	1.00×10^{-1}	–	–
X-DER-Contr Margin	–	–	3.00×10^{-1}	–	–
X-DER-Constr η	–	–	1.00×10^{-1}	–	–
X-DER-Future Constr	–	–	1.00×10^{0}	–	–
X-DER-Part Constr	–	–	0.00×10^{0}	–	–
L2P-R-Pull-Constr	–	–	–	1.00×10^{-1}	–
Inf-SSM-λ	2.50×10^{2}	1.00×10^{0}	1.00×10^{3}	2.00×10^{3}	1.00×10^{4}

Table 18. Hyperparameters for ER, LUCIR, X-DER, L2P-R, and CLFD methods integration tests with Inf-SSM for Caltech-256 5 tasks setting. Configuration includes both general training settings and method-specific parameters.

Hyperparameter	**ER**	**LUCIR**	**X-DER**	**L2P-R**	**CLFD**
Batch Size	128	128	32	32	64
Training Epochs	40	40	20	40	20
Base LR	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}
Warmup LR	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}
Minimum LR	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}
Task LR Scaling	5.00×10^{-1}	5.00×10^{-1}	5.00×10^{-1}	5.00×10^{-1}	5.00×10^{-1}
Weight Decay	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
Buffer Size	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}
LUCIR-λ_{base}	–	5.00×10^{-1}	–	–	–
LUCIR-λ_{MR}	–	1.00×10^{-1}	–	–	–
LUCIR-K_{MR}	–	2.00×10^{0}	–	–	–
LUCIR-MR Margin	–	5.00×10^{-2}	–	–	–
X-DER-γ	–	–	8.50×10^{-1}	–	–
X-DER-Temp	–	–	7.00×10^{-2}	–	–
X-DER-Base Temp	–	–	7.00×10^{-2}	–	–
X-DER-α	–	–	3.00×10^{-1}	–	–
X-DER-β	–	–	1.80×10^{0}	–	–
X-DER-SimCLR Batch Size	–	–	3.20×10^{1}	–	–
X-DER-SimCLR Num Augs	–	–	2.00×10^{0}	–	–
X-DER-λ	–	–	5.00×10^{-2}	–	–
X-DER-dp Weight	–	–	1.00×10^{-1}	–	–
X-DER-Contr Margin	–	–	3.00×10^{-1}	–	–
X-DER-Constr η	–	–	1.00×10^{-1}	–	–
X-DER-Future Constr	–	–	1.00×10^{0}	–	–
X-DER-Part Constr	–	–	0.00×10^{0}	–	–
L2P-R-Pull-Constr	–	–	–	1.00×10^{-1}	–
Inf-SSM-λ	1.00×10^{3}	3.00×10^{3}	5.00×10^{2}	5.00×10^{2}	1.00×10^{3}

Table 19. Hyperparameters for ER, LUCIR, X-DER, L2P-R, and CLFD methods integration tests with Inf-SSM for Caltech-256 10 tasks setting. Configuration includes both general training settings and method-specific parameters.

Hyperparameter	**ER**	**LUCIR**	**X-DER**	**L2P-R**	**CLFD**
Batch Size	128	128	32	32	64
Training Epochs	40	40	20	40	20
Base LR	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}	1.00×10^{-4}
Warmup LR	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}
Minimum LR	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}	1.00×10^{-5}
Task LR Scaling	5.00×10^{-1}	5.00×10^{-1}	5.00×10^{-1}	5.00×10^{-1}	5.00×10^{-1}
Weight Decay	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}	1.00×10^{-1}
Buffer Size	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}	1.00×10^{3}
LUCIR-λ_{base}	–	5.00×10^{-1}	–	–	–
LUCIR-λ_{MR}	–	1.00×10^{-1}	–	–	–
LUCIR-K_{MR}	–	2.00×10^{0}	–	–	–
LUCIR-MR Margin	–	5.00×10^{-2}	–	–	–
X-DER-γ	–	–	8.50×10^{-1}	–	–
X-DER-Temp	–	–	7.00×10^{-2}	–	–
X-DER-Base Temp	–	–	7.00×10^{-2}	–	–
X-DER-α	–	–	3.00×10^{-1}	–	–
X-DER-β	–	–	1.80×10^{0}	–	–
X-DER-SimCLR Batch Size	–	–	3.20×10^{1}	–	–
X-DER-SimCLR Num Augs	–	–	2.00×10^{0}	–	–
X-DER-λ	–	–	5.00×10^{-2}	–	–
X-DER-dp Weight	–	–	1.00×10^{-1}	–	–
X-DER-Contr Margin	–	–	3.00×10^{-1}	–	–
X-DER-Constr η	–	–	1.00×10^{-1}	–	–
X-DER-Future Constr	–	–	1.00×10^{0}	–	–
X-DER-Part Constr	–	–	0.00×10^{0}	–	–
L2P-R-Pull-Constr	–	–	–	1.00×10^{-1}	–
Inf-SSM-λ	1.00×10^{3}	1.00×10^{3}	5.00×10^{2}	2.00×10^{3}	5.00×10^{3}