λ-JEPA: SPECTRAL ANTI-COLLAPSE REGULARIZATION FOR SELF-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Joint-embedding self-supervised learning typically combines an invariance objective across augmented views with additional mechanisms to prevent representational collapse. These objectives are often applied after a projection head, while downstream tasks use the backbone representation before the projector. We find that this mismatch does not necessarily prevent dimensional collapse in the backbone, which can retain low effective rank and potentially limit downstream transfer. To address this, we introduce `SACReg`, a spectral anti-collapse regularizer motivated by an analysis of λ-balance, which captures the relative scale of weight matrices across layers. In a two-layer linear network, we show that *(i)* λ-balance prevents collapse, and *(ii)* our regularizer applied to the backbone induces λ-balance. In the nonlinear case, this regularizer leads to anti-collapse as well and, in realistic architectures on ImageNet100, it empirically increases the representations' ranks. We apply `SACReg` to JEPA and propose λ-JEPA, which improves over LeJEPA and VISReg on ImageNet-1k classification and in average linear-probe transfer performance across eight downstream image datasets. On video self-supervised learning, λ-JEPA improves over LeVJEPA and V-JEPA 2 on the Something-Something-v2 and Kinetics-400 benchmarks.

1 INTRODUCTION

Self-supervised learning aims to learn transferable representations from unlabeled data without requiring manual annotations (Jing & Tian, 2021). In computer vision, joint-embedding self-supervised learning (JE-SSL) has become one of the most promising approaches, encouraging representations of related views or regions of an image to agree while additional mechanisms prevent representational collapse (Jing et al., 2022; Bardes et al., 2022). These mechanisms include using contrastive negatives (Chen et al., 2020; He et al., 2020), asymmetric teacher-student training (Caron et al., 2021; Zhou et al., 2022; Grill et al., 2020) and explicit anti-collapse regularization (Balestriero & LeCun, 2025; Bardes et al., 2022; Wu et al., 2026).

JE-SSL methods typically optimize a projected representation during pretraining, while downstream tasks discard the non-linear projection head and use the backbone representation instead. These two representation spaces can differ substantially in both geometry and downstream performance (Bordes et al., 2023). In contrastive self-supervised learning, this has also been studied through dimensional collapse, where learned representations span only a lower dimensional part of the available feature space (Jing et al., 2022). We observe an analogous phenomenon in explicitly regularized JE-SSL methods. Although their anti-collapse objectives successfully maintain high rank projected representations, the corresponding backbone representations can remain low rank. This matters for task-agnostic representation learning, where different downstream tasks can benefit from different invariances (Ericsson et al., 2021; Tian et al., 2020b). Dimensional collapse in the backbone can therefore limit transfer by reducing the set of feature directions available for downstream tasks. Consistent with this view, effective rank has been shown to predict downstream performance in JE-SSL (Garrido et al., 2023).

The rank of learned representations has been studied extensively in the feature-learning literature, with prior work linking representation rank to distinct learning regimes in linear and nonlinear networks (Saxe et al., 2014; Atanasov et al., 2022). Building on this perspective, we analyze a two-layer linear model with λ-balance initialization, which captures the relative scale of adjacent layers, and

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

connects it to representation rank. The resulting spectral anti-collapse mechanism motivates a covariance regularizer for nonlinear networks, applied directly to the backbone features to restore their representation rank. In other words, our λ-balance theory prescribes both the form of the regularizer and where it should be applied. We then apply it accordingly to JE-SSL in a method we call λ-JEPA, using view-averaged backbone representations to prevent collapse and promote variation across images (see Figure 6). In controlled experiments, we show that λ-JEPA improves class-relevant separation while preserving augmentation dependent variation. At larger scale, it outperforms both LeJEPA (Balestriero & LeCun, 2025) and VISReg (Wu et al., 2026) on ImageNet-1K (Russakovsky et al., 2015) trained from scratch, with performance approaching DINO (Caron et al., 2021) in both linear probing and transfer learning. The same approach applies off the shelf to video JEPA models, where λ-JEPA trained from scratch outperforms V-JEPA 2 (Assran et al., 2025) and LeVJEPA (Kuhn et al., 2026) on Something-Something-v2 (Goyal et al., 2017) and Kinetics-400 (Kay et al., 2017).

We summarize our main contributions as:

- We show that standard JE-SSL methods designed to prevent collapse in the projected representation do not ensure a high-rank backbone, potentially limiting its representational capacity for downstream tasks.
- We derive SACReg, a spectral anti-collapse regularizer that guarantees encoder non-collapse in a two-layer linear network, motivating encoder-side spectral regularization. We extend this to nonlinear encoders by directly regularizing representation covariance and apply it to JE-SSL using view-averaged backbone representations to prevent collapse and promote variation across images.
- We propose λ-JEPA, a standalone SSL method that applies SACReg to both the backbone and projected representations. Across image JE-SSL benchmarks, λ-JEPA improves frozen transfer while remaining competitive on ImageNet-1k classification, and extends to video SSL with large gains on temporal recognition.

2 Problem and Related Works

Joint-embedding SSL Modern joint-embedding self-supervised learning (JE-SSL) methods differ primarily in how they avoid collapsed solutions. We focus on a family of explicitly regularized methods (VICReg (Bardes et al., 2022), LeJEPA (Balestriero & LeCun, 2025), VISReg (Wu et al., 2026)). In the projected space, VICReg controls feature variance and covariance, while LeJEPA regularizes the embedding distribution toward an isotropic Gaussian. VISReg similarly combines variance regularization with an additional distributional constraint. Following the use of projection heads in SimCLR (Chen et al., 2020), these objectives act after a non-linear projection head, which is discarded for downstream tasks.

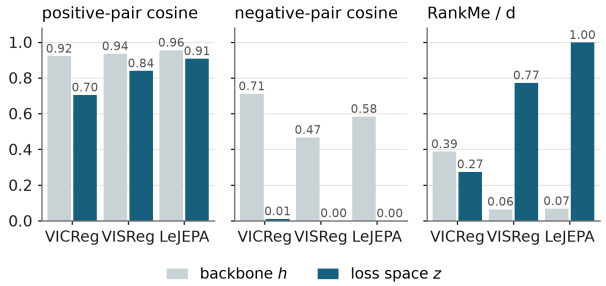

Figure 1: Backbone and loss space geometry for explicitly regularized JE-SSL methods. The projection head changes both pairwise similarities and RankMe.

We focus on a specific consequence of this separation: in these explicitly regularized methods, anti-collapse constraints are imposed on $\mathbf{z}$, yet with nonlinear projectors, preventing dimensional collapse in $\mathbf{z}$ does not in general prevent it in the backbone representation $\mathbf{h}$.[1]. Figure 1 shows how two key aspects of explicitly regularized JE-SSL methods change across the projection head: (i) agreement between augmented views and (ii) prevention of representational collapse. Positive pairs remain highly aligned in both the backbone and projected spaces. In contrast, different images have very similar representations in the backbone, indicating substantial concentration. The same discrepancy appears in representation rank. RankMe (Garrido et al., 2023) measures how evenly variance is distributed across feature directions and is typically higher in the projected space, whereas backbone

[1]For example, let $\mathbf{h} = (X, 0)^\top$ with X uniform on $\{-1, 0, 1\}$, and let $p(\mathbf{h}) = (h_1, |h_1|)^\top$. Then $\mathrm{Cov}(\mathbf{h})$ has rank one, while $\mathrm{Cov}(p(\mathbf{h})) = \mathrm{diag}(2/3, 2/9)$ has full rank.

remains low rank, especially for VISReg and LeJEPA. For task-agnostic representation learning, low rank backbones can be restrictive because different downstream tasks may rely on different feature directions (Ericsson et al., 2021; Tian et al., 2020b). Consistently, Garrido et al. (2023) have shown that it correlates with downstream performance in JE-SSL, motivating backbone-level anti-collapse regularization.

Feature learning and anti-collapse of the backbone representations The question of what determines the rank of learned representations has been studied extensively in the feature learning literature which has so far had little contact with SSL but offers directly relevant tools. In particular, deep linear networks provide a tractable setting for studying these effects. Despite their linear end-to-end mapping, their factorized parameterization induces nonlinear optimization dynamics (Baldi & Hornik, 1989; Fukumizu, 1998; Saxe et al., 2014; Du et al., 2019; Jacot et al., 2018; Chizat et al., 2019; Braun et al., 2022) and captures phenomena observed in nonlinear networks (Saxe et al., 2019; Kunin et al., 2024; Nam et al., 2025; Anguita et al., 2026). Initialization plays a central role in these dynamics: small initializations favor low-rank or sparse solutions, whereas appropriate large-scale limits yield kernel-like behavior (Saxe et al., 2014; Gunasekar et al., 2018; Chizat & Bach, 2020; Woodworth et al., 2020; Li et al., 2021). In nonlinear networks, initialization scale likewise influences the learning regime (Luo et al., 2021; Atanasov et al., 2022). Beyond overall initialization scale, the relative scales of adjacent layers, as studied through λ-balanced initializations, also influence learning dynamics and implicit bias (Azulay et al., 2021; Kunin et al., 2024; Dominé et al., 2025; Jarvis et al., 2025). Their consequences for representation rank, however, remain less characterized, particularly in the nonlinear regime and in larger scale networks. Our work addresses this gap by investigating how layer balance shapes representation rank and using these insights to derive a regularizer that prevents collapse in the backbone representations of SSL models.

3 METHOD

Our goal is to prevent collapse in the backbone representations retained for downstream transfer. We first study a tractable two-layer linear network, where we show that negative layer balance yields a non-collapse guarantee and motivates an encoder-side spectral regularizer. We extend this principle to nonlinear encoders by directly regularizing representation covariance, then apply it to JE-SSL on view-averaged backbone representations to promote variation across images. Figure 6 illustrates the resulting λ-JEPA objective and the placement of SACReg in the backbone and projected spaces.

3.1 DERIVING A SPECTRAL ANTI-COLLAPSE MECHANISM IN LINEAR NETWORKS

Linear-network setting. Consider a supervised dataset $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{P}$, $\mathbf{x}_n \in \mathbb{R}^{N_i}$, $\mathbf{y}_n \in \mathbb{R}^{N_o}$. We model the input–output mapping using a two-layer linear network, $\widehat{\mathbf{y}}_n = \mathbf{W}_2\mathbf{W}_1\mathbf{x}_n$, where $\mathbf{W}_1 \in \mathbb{R}^{N_h \times N_i}$ and $\mathbf{W}_2 \in \mathbb{R}^{N_o \times N_h}$ denote the encoder and decoder weight matrices, respectively. The network is trained on the mean-squared-error loss $\mathcal{L}_{\text{task}}(\mathbf{W}_1, \mathbf{W}_2) = \frac{1}{2}\left\langle \|\mathbf{W}_2\mathbf{W}_1\mathbf{x} - \mathbf{y}\|_2^2 \right\rangle$, where $\langle\cdot\rangle$ denotes the empirical average over the training set. Although the task loss depends only on the end-to-end mapping $\mathbf{M} = \mathbf{W}_2\mathbf{W}_1$, its factorization across the two layers affects the parameter and representation dynamics. We characterize this factorization through the layer-imbalance matrix $\mathbf{\Delta} = \mathbf{W}_2^\top\mathbf{W}_2 - \mathbf{W}_1\mathbf{W}_1^\top$. A network is said to be λ-balanced when $\mathbf{\Delta} = \lambda\mathbf{I}_{N_h}$. Previous work has shown that λ-balanced initializations provide an analytically tractable framework for characterizing the transition between rich and lazy learning dynamics (Dominé et al., 2025; Nam et al., 2025). Under unregularized gradient flow, $\mathbf{\Delta}$ is conserved, so the layer balance remains fixed at its initial value. Prior studies have also linked these learning regimes to the rank of the representations Saxe et al. (2014); Atanasov et al. (2022). Motivated by this connection, we investigate whether a suitable layer balance can prevent representation collapse. We show that an appropriately chosen negative balance provides such a guarantee, and use this result to derive a regularizer that dynamically promotes the same anti-collapse property during training.

Balancedness anti-collapse property. A negative layer balance provides an explicit anti-collapse guarantee. If $\mathbf{\Delta} = \lambda_{\text{bal}}\mathbf{I}_{N_h}$ with $\lambda_{\text{bal}} < 0$, then $\mathbf{W}_1\mathbf{W}_1^\top \succeq -\lambda_{\text{bal}}\mathbf{I}_{N_h}$, and $\sigma_{\min}(\mathbf{W}_1) \geq \sqrt{-\lambda_{\text{bal}}}$, so the encoder has full row rank. This also yields a guarantee for the hidden representations under the assumption that, for some $\kappa_x > 0$, the input covariance $\mathbf{\Sigma}_x = \frac{1}{P}\sum_{n=1}^{P}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top \succeq \kappa_x\mathbf{I}_{N_i}$, with $\bar{\mathbf{x}} = \frac{1}{P}\sum_{n=1}^{P}\mathbf{x}_n$. In fact, the centered covariance of the hidden representations $\mathbf{h}_n =$

$\mathbf{W}_1\mathbf{x}_n$ satisfies $\mathbf{C}_h = \mathbf{W}_1\mathbf{\Sigma}_x\mathbf{W}_1^\top \succeq -\kappa_x\lambda_{\text{bal}}\mathbf{I}_{N_h}$. Therefore, negative layer balance prevents representation collapse, see Lemma A.4 in Appendix A.1.3 for a formal statement and proof.

Although a negative balanced initialization prevents collapse, extending the same initialization strategy to deep nonlinear networks is non-trivial. We therefore seek a regularized training objective that dynamically selects a negative layer balance, independently of its initial value. Motivated by this objective, we derive a regularizer that follows from a constrained optimization problem.

Layer balance through regularization. Among all factorizations of a fixed end-to-end mapping $\mathbf{M} = \mathbf{W}_2\mathbf{W}_1$, we look for a constrained minimizer that satisfies $\mathbf{W}_2^\top\mathbf{W}_2 - \mathbf{W}_1\mathbf{W}_1^\top = -\frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h}$.

Theorem 3.1. *Fix an end-to-end mapping $\mathbf{M} \in \mathbb{R}^{N_o \times N_i}$, $\gamma > 0$ and $\lambda_{\text{reg}} > 0$. Let $(\mathbf{W}_1^\star, \mathbf{W}_2^\star)$ be a solution of*

$$\min_{\mathbf{W}_1,\mathbf{W}_2} \quad \frac{\lambda_{\text{reg}}}{2}\left(\|\mathbf{W}_1\|_F^2 + \|\mathbf{W}_2\|_F^2\right) - \frac{\gamma}{2}\log\det\left(\mathbf{W}_1\mathbf{W}_1^\top\right) \quad \textit{subject to} \quad \mathbf{W}_2\mathbf{W}_1 = \mathbf{M}, \tag{1}$$

with $\mathbf{W}_1^\star(\mathbf{W}_1^\star)^\top \succ 0$. Then, $(\mathbf{W}_2^\star)^\top\mathbf{W}_2^\star - \mathbf{W}_1^\star(\mathbf{W}_1^\star)^\top = -\frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h}$

The proof of Theorem 3.1 is provided in Appendix A.1.4. This motivates the following definition.

Definition 3.2 (Linear SACReg)**.** *Assume that $\mathbf{W}_1\mathbf{W}_1^\top \succ 0$, $\gamma > 0$ and $\lambda_{\text{reg}} > 0$. We define*

$$\mathcal{R}_{\text{LSAC}}(\mathbf{W}_1, \mathbf{W}_2) = \frac{\lambda_{\text{reg}}}{2}\left(\|\mathbf{W}_1\|_F^2 + \|\mathbf{W}_2\|_F^2\right) - \frac{\gamma}{2}\log\det\left(\mathbf{W}_1\mathbf{W}_1^\top\right),$$

and the training objective $\mathcal{L}_{\text{LSAC}}(\mathbf{W}_1, \mathbf{W}_2) = \mathcal{L}_{\text{task}}(\mathbf{W}_1, \mathbf{W}_2) + \mathcal{R}_{\text{LSAC}}(\mathbf{W}_1, \mathbf{W}_2)$.

The regularizer combines symmetric L_2 weight decay with a log-determinant term acting specifically on the encoder Gram matrix. This placement is motivated by the non-collapse condition: a negative layer balance bounds the encoder Gram matrix away from being singular and under the input-covariance assumption, guarantees non-collapse of the hidden representations. The log-determinant term penalizes vanishing encoder singular values, while weight decay controls the overall weight scale.

We next note that the balance identified by the constrained optimization problem also emerges dynamically during training. If $N_h \leq N_i$ and $\mathbf{W}_1(0)\mathbf{W}_1(0)^\top \succ 0$, the regularized gradient-flow solution for the mean-squared-error objective exists for all $t \geq 0$, and the encoder retains full row rank throughout training. The task-loss terms cancel in the imbalance dynamics, giving $\tau\dot{\mathbf{\Delta}} = -2\lambda_{\text{reg}}\mathbf{\Delta} - 2\gamma\mathbf{I}_{N_h}$. Consequently, $\mathbf{\Delta}(t) + \frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h} = e^{-2\lambda_{\text{reg}}t/\tau}\left(\mathbf{\Delta}(0) + \frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h}\right)$. Thus, independently of the dataset and initial imbalance, the network converges exponentially with rate $2\lambda_{\text{reg}}/\tau$ to $\lambda_{\text{bal}}^{\text{enc}}$-balanced solution, with $\lambda_{\text{bal}}^{\text{enc}} = -\gamma/\lambda_{\text{reg}}$ (see Theorem A.7 in Appendix A.1.5). The network preserves this balance throughout training (Corollary A.8). Unlike λ-balanced initialization, which fixes the balance only through the starting condition, the proposed objective continuously attracts the network towards the selected balance. The ratio $\gamma/\lambda_{\text{reg}}$ determines the fixed point, while λ_{reg} controls the rate of convergence towards it. The ratio $\gamma/\lambda_{\text{reg}}$ determines the selected balance and the corresponding lower bound on the encoder spectrum and yields a non-collapse guarantee. This motivates applying the penalty directly to the representation covariance in nonlinear encoders, as developed next.

3.2 Spectral regularization of nonlinear representations

Although full matrix imbalance is not conserved in nonlinear networks, prior work has shown that it influences their learning dynamics (Kunin et al., 2024; Anguita et al., 2026; Jarvis et al., 2025). Motivated by this connection and our linear analysis, we apply the spectral anti-collapse principle directly to the encoder's representation covariance, preserving the encoder-side placement without relying on exact balance dynamics. Let $\mathbf{h} = f_\theta(\mathbf{x}) \in \mathbb{R}^{N_h}$ denote the representation produced by a differentiable encoder f_θ. For a minibatch of size B, collect the representations row-wise in $\mathbf{H} \in \mathbb{R}^{B\times N_h}$ and define $\boldsymbol{\mu} = \frac{1}{B}\mathbf{H}^\top\mathbf{1}_B$, $\mathbf{P}_B = \mathbf{I}_B - \frac{1}{B}\mathbf{1}_B\mathbf{1}_B^\top$, $\widetilde{\mathbf{H}} = \mathbf{P}_B\mathbf{H}$, $\mathbf{C}_h = \frac{1}{B}\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}}$. Here, $\boldsymbol{\mu}$ and $\mathbf{C}_h$ are the empirical representation mean and centered covariance, respectively.

Definition 3.3 (Nonlinear SACReg). *Let $\lambda_{\text{mean}} > 0$, $\lambda_{\text{cov}} > 0$, $\gamma > 0$, and $\varepsilon \geq 0$. Assume that $\mathbf{C}_h + \varepsilon \mathbf{I}_{N_h} \succ 0$. We define*

$$\mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) = \frac{\lambda_{\text{mean}}}{2} \|\boldsymbol{\mu}\|_2^2 + \frac{\lambda_{\text{cov}}}{2} \operatorname{tr}(\mathbf{C}_h) - \frac{\gamma}{2} \log\det\left(\mathbf{C}_h + \varepsilon \mathbf{I}_{N_h}\right), \tag{2}$$

and the corresponding training objective $\mathcal{L}_{\text{SAC}} = \mathcal{L}_{\text{task}} + \mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h)$.

The covariance terms play the same roles as in the linear setting: λ_{cov} controls the total representation variance, while γ controls the strength of the log-determinant penalty on small covariance eigenvalues. The parameter ε provides numerical stabilization. The nonlinear regularizer also includes a mean penalty $\|\boldsymbol{\mu}\|_2^2$, weighted by λ_{mean}, which pulls the representation mean towards the origin. For a bias-free linear encoder, centered inputs produce centered representations, so no separate mean penalty is needed. For a linear encoder with whitened inputs, the covariance penalty reduces exactly to the encoder portion of the linear regularizer, with $\lambda_{\text{cov}} = \lambda_{\text{reg}}$. This connection is derived in Appendix A.2.1.

Anti-collapse property. We next characterize the regularizer's inductive bias and anti-collapse properties. The mean and trace penalties control representation location and scale, while the negative log-determinant penalizes small covariance eigenvalues.

Theorem 3.4 (Optimal mean and covariance). *The unique minimizer of $\mathcal{R}_{\text{SAC}}$ over $\boldsymbol{\mu} \in \mathbb{R}^{N_h}$, $\mathbf{C}_h \succeq 0$ such that $\mathbf{C}_h + \varepsilon \mathbf{I}_{N_h} \succ 0$ is $\boldsymbol{\mu}^\star = \mathbf{0}$, $\mathbf{C}_h^\star = \left(\frac{\gamma}{\lambda_{\text{cov}}} - \varepsilon\right)_+ \mathbf{I}_{N_h}$, where $(a)_+ = \max\{a, 0\}$.*

Thus, the regularizer favors zero-mean representations with isotropic covariance, which is full rank when $\gamma > \lambda_{\text{cov}}\varepsilon$. For $\varepsilon = 0$, the preferred variance is $\gamma/\lambda_{\text{cov}}$. These are the preferred moments of the regularizer alone; the task loss, network parameterization, and minibatch rank constraints may prevent their attainment.

When $\varepsilon = 0$, the negative log-determinant is a strict barrier against covariance collapse: $\lambda_{\min}(\mathbf{C}_h) \to 0$ implies $\mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \to +\infty$. At the other extreme, the trace term prevents unbounded growth: $\lambda_{\max}(\mathbf{C}_h) \to +\infty$ implies $\mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \to +\infty$. The mean term supplies a third, independent barrier: $\|\boldsymbol{\mu}\| \to \infty$ implies $\mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \to +\infty$, so the representation cannot escape the regularizer by drifting off in mean instead of collapsing or exploding in covariance. Consequently, every finite sublevel set has bounded mean and covariance eigenvalues bounded above and away from zero; see Theorem A.15 in Appendix A.2.2. We evaluate this property experimentally in nonlinear ReLU networks, following Kunin et al. (2024). Consistent with our predictions, the regularizer increases representation rank across initialization scales (Fig. 4 in Appendix A.2.3).

Altogether, these results suggest that SACReg promotes higher-rank representations in nonlinear networks, supporting its role as an anti-collapse regularizer while also preserving feature learning.

3.3 Spectral anti-collapse for JE-SSL

As shown in Fig. 1, standard JE-SSL objectives can leave backbone representations dimensionally collapsed. To address this, we apply the spectral anti-collapse regularizer (SACReg), introduced in Section 3.2, to joint-embedding self-supervised learning (JE-SSL), targeting variation between images to prevent representation collapse. Standard JE-SSL objectives enforce invariance and prevent collapse in the projected space $\mathbf{z}$. Motivated by our analysis of λ-balance and the resulting anti-collapse guarantees, our standalone SSL method, λ-JEPA, applies SACReg to both backbone and projected representations. For a batch of B images, each with V augmented views, let $\mathbf{h}_i^{(j)} = f_\theta(\mathbf{v}_i^{(j)})$, and $\mathbf{z}_i^{(j)} = p_\phi(\mathbf{h}_i^{(j)})$ denote the backbone and projected representations, respectively. Define their view centers by $\bar{\mathbf{h}}_i = \frac{1}{V}\sum_{j=1}^V \mathbf{h}_i^{(j)}$, $\bar{\mathbf{z}}_i = \frac{1}{V}\sum_{j=1}^V \mathbf{z}_i^{(j)}$, and write $\bar{\mathbf{H}} = \{\bar{\mathbf{h}}_i\}_{i=1}^B$, $\bar{\mathbf{Z}} = \{\bar{\mathbf{z}}_i\}_{i=1}^B$, and $\mathbf{Z} = \{\mathbf{z}_i^{(j)}\}_{i=1,\ldots,B;\, j=1,\ldots,V}$. For a batch of view centers $\mathbf{R} = \{\mathbf{r}_i\}_{i=1}^B \subset \mathbb{R}^d$, define its empirical mean and centered covariance as $\boldsymbol{\mu}_R = \frac{1}{B}\sum_{i=1}^B \mathbf{r}_i$, and $\boldsymbol{\Sigma}_R = \frac{1}{B}\sum_{i=1}^B (\mathbf{r}_i - \boldsymbol{\mu}_R)(\mathbf{r}_i - \boldsymbol{\mu}_R)^\top$. We average over V views so that SACReg mainly encourages variation across images rather than augmentations.

Definition 3.5 (SACReg for JE-SSL). *For $\varepsilon \geq 0$ such that $\mathbf{\Sigma}_R + \varepsilon \mathbf{I}_d \succ 0$, we define*

$$\mathrm{SACReg}(\mathbf{R}) = \frac{1}{2}\left[\|\boldsymbol{\mu}_R\|_2^2 + \mathrm{tr}(\mathbf{\Sigma}_R) - \log\det(\mathbf{\Sigma}_R + \varepsilon \mathbf{I}_d) - d\right].$$

Given an invariance objective $\mathcal{L}_{\mathrm{SSL}}(\mathbf{Z})$ and weights $\beta_h, \beta_z > 0$, the λ-JEPA objective is

$$\mathcal{L}_{\lambda\text{-JEPA}} = \mathcal{L}_{\mathrm{SSL}}(\mathbf{Z}) + \beta_h\ \mathrm{SACReg}(\bar{\mathbf{H}}) + \beta_z\ \mathrm{SACReg}(\bar{\mathbf{Z}}). \tag{3}$$

The mean term centers the representation, the trace controls its overall scale, and the negative log-determinant penalizes vanishing covariance directions. The projected-space term provides anti-collapse where the SSL objective is optimized, while the backbone term directly protects the representation retained for downstream transfer. When applying our regularizer to an existing SSL method that already contains its own projected-space anti-collapse mechanism, we leave its original objective unchanged and add only the backbone term, $\beta_h\,\mathrm{SACReg}(\bar{\mathbf{H}})$. In practice, when the feature dimension exceeds the number of image centers in a minibatch, we evaluate the regularizer over orthogonal lower-dimensional slices of the representation, details are given in Appendix B.

4 Experiments

We evaluate whether SACReg improves the performance and transferability of representations. We first report the performance of the standalone λ-JEPA objective on image and video benchmarks, then use controlled ImageNet-100 (Tian et al., 2020a) experiments to isolate the effect of backbone regularization across diverse SSL objectives and analyze how it changes representation geometry.

4.1 Setup

We consider two experimental settings. Our main evaluation covers large-scale image and video SSL. For images, we train ViT-S and ViT-B (Dosovitskiy et al., 2021) on ImageNet-1k (Russakovsky et al., 2015) for 100 and 400 epochs from scratch. For video, we follow LeVJEPA (Kuhn et al., 2026) and train ViT-S and ViT-B from scratch on a class-balanced 20% subset of Kinetics-710 (Li et al., 2023), for up to 1,085 epochs. Our controlled experiments use ViT-S on ImageNet-100 (Tian et al., 2020a) to study the effect of SACReg across different SSL objectives.

Due to compute limitations, we do not retune hyperparameters for the 400-epoch ImageNet-1k or 1,085-epoch video runs; we use these primarily to study scaling with extended training. For evaluation, we follow standard frozen feature protocols: the Lightly (Susmelj et al., 2020) linear probe benchmark on ImageNet-1k, the VISReg (Wu et al., 2026) pipeline for transfer and the LeVJEPA pipeline for video. Details are given in Appendix C. Whenever public checkpoints trained under a common codebase are available (OpenKnowledge AI[2]), we evaluate them under the same pipeline, otherwise we report published numbers. We set the loss weights using the gradient-based calibration procedure described in Appendix C.3.

4.2 Performance on image self-supervised learning

Table 1a reports our main 100-epoch results on ImageNet-1k linear probing and average transfer across eight classification datasets. We compare most directly with LeJEPA (Balestriero & LeCun, 2025) and VISReg (Wu et al., 2026), the related explicitly regularized JE-SSL methods. We include DINO (Caron et al., 2021) and iBOT (Zhou et al., 2022) as established SSL baselines for broader comparison which, until now, have significantly outperformed JEPA-based models. At 100 epochs, where training length is matched, λ-JEPA substantially improves over prior explicitly regularized methods on both ImageNet-1k linear probing and transfer. The gains are particularly strong on transfer, improving over LeJEPA by 8.9 points with ViT-S and over VISReg by 4.5 points with ViT-B. λ-JEPA is also competitive with DINO and iBOT: its ImageNet-1k performance approaches these methods, while both model sizes achieve the highest average transfer among the 100-epoch results.

[2]https://huggingface.co/collections/OK-AI/imagenet-1k-self-supervised-vit-baselines

(a) Main: 100 epochs

Method	Backbone	Linear	Transfer
LeJEPA	ViT-S/16	62.5	68.7
DINO	ViT-S/16	70.0	75.7
iBOT	ViT-S/16	70.9	75.9
λ-JEPA	ViT-S/16	69.7	77.6
LeJEPA	ViT-B/16	69.7	74.2
DINO	ViT-B/16	73.9	78.3
iBOT	ViT-B/16	77.0	80.7
VISReg	ViT-B/16	70.3	76.4
λ-JEPA	ViT-B/16	74.2	80.9

(b) Longer training

Method	Backbone	Ep.	Linear	Transfer
LeJEPA	ViT-S/16	300	66.4	71.9
DINO	ViT-S/16	300	73.8	79.4
iBOT	ViT-S/16	300	75.2	79.5
λ-JEPA	ViT-S/16	400	72.3	79.3
LeJEPA	ViT-B/16	300	72.4	75.2
DINO	ViT-B/16	300	75.2	79.3
iBOT	ViT-B/16	300	78.6	82.3
MoCo v3‡	ViT-B/16	300	75.9	80.5
DINO‡	ViT-B/16	400	77.2	83.1
iBOT‡	ViT-B/16	400	78.5	83.2
VISReg‡	ViT-B/16	400	74.6	79.1
λ-JEPA	ViT-B/16	400	76.0	82.0

Table 1: ImageNet-1k linear probing and mean linear-probe transfer over eight datasets. **(a)** Main 100-epoch comparison. **(b)** Longer-training results, including public SSL checkpoints for context. ‡ Transfer results are reported by VISReg; all others are evaluated by us under the matched VISReg protocol. Detailed results are in Appendix C.6.

Table 1b reports results with extended pretraining: we use the same recipe and hyperparameters as the 100-epoch experiments, without any additional tuning for the longer runs. Extending λ-JEPA to 400 epochs improves both ImageNet-1k linear probing and transfer performance. Since the available checkpoints use different training lengths, we include them as longer training comparisons rather than strictly matched baselines. With ViT-B, λ-JEPA reaches 82.0 average transfer, outperforming the 400-epoch VISReg result by 2.9 points and coming within 1.2 points of DINO and iBOT. With ViT-S, it reaches 79.3 average transfer, essentially matching the 300-epoch DINO and iBOT results.

Takeaway. λ-JEPA substantially improves over prior explicitly regularized JE-SSL methods on both ImageNet-1k classification and transfer. To the best of our knowledge, it is the first JEPA-based method to approach DINO and iBOT on both evaluations, while achieving the highest average transfer performance at 100 epochs.

4.3 Performance on video self-supervised learning

We apply the same objective to video by replacing the loss inside the LeVJEPA (Kuhn et al., 2026) training pipeline while keeping the model, optimizer and data pipeline unchanged. We train these models from scratch and freeze the encoders after pretraining. The frozen encoders are evaluated on three benchmarks with different characteristics: ImageNet-1k for recognition from a single image, Something-Something-v2 (SSv2) (Goyal et al., 2017) for temporal reasoning and Kinetics-400 (Kay et al., 2017) for action recognition.

Method	IN-1k	SSv2	K400
ViT-S/16, 240 epochs			
LeVJEPA	39.4	—	—
V-JEPA 2	38.7	—	—
λ-JEPA	46.8	37.3	40.0
ViT-B/16, 240 epochs			
VideoMAEv2	47.1	—	—
V-JEPA 2	51.6	—	—
LeVJEPA	50.7	30.4	—
λ-JEPA	52.9	43.9	44.2
ViT-B/16, longer training			
VideoMAEv2	53.4	43.6	37.4
V-JEPA 2	51.6	42.5	40.7
LeVJEPA	61.0	40.4	44.6
λ-JEPA	57.1	48.3	45.7

Table 2: Video SSL on IN-1k, SSv2, and K400. Longer training LeVJEPA and λ-JEPA use 1085 epochs.

Table 2 compares λ-JEPA with LeVJEPA, V-JEPA 2 (Assran et al., 2025) and VideoMAEv2 (Wang et al., 2023). At 240 epochs, λ-JEPA improves over LeVJEPA and V-JEPA 2 on ImageNet-1k for both ViT-S and ViT-B. The largest gains are on SSv2: our ViT-B improves over LeVJEPA by 13.5 points at 240 epochs and by 7.9 points at 1,085 epochs. For ImageNet-1k and SSv2, LeVJEPA evaluates frozen features with an attentive probe followed by a linear classifier, whereas on K400 it reports a linear probe on mean-pooled features. In our setting, SACReg is applied to the CLS representation, so we report a CLS probe in Table 2, evaluating the representation directly regularized during pretraining. It reaches 45.7 at 1,085 epochs compared with 44.6 for LeVJEPA's probe over the mean-pooled features. For completeness, we also evalu-

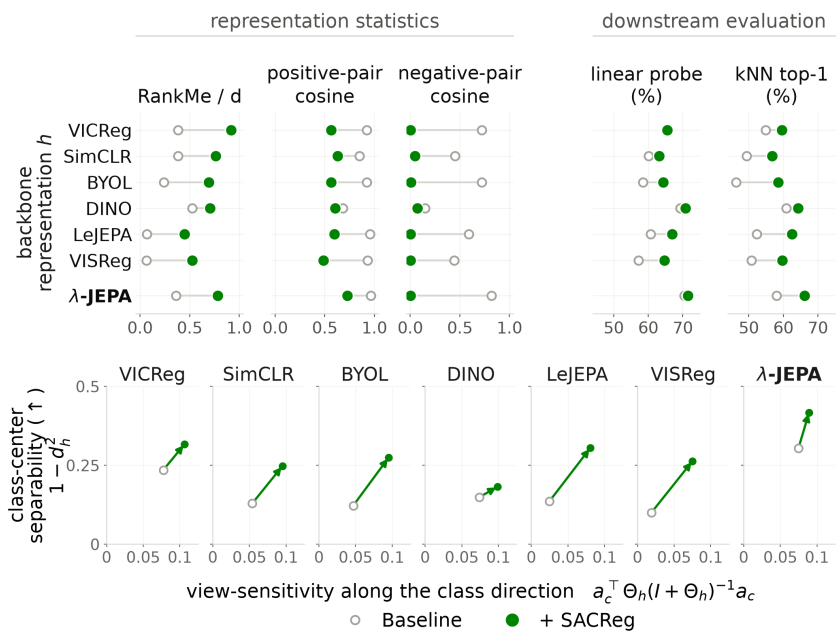

Figure 2: Effect of backbone SACReg across SSL objectives on ImageNet-100. **Top:** backbone statistics and downstream evaluation. **Bottom:** class-center separability versus view sensitivity along class-prediction directions. Open and green markers denote baseline and +SACReg, respectively; arrows show their change. Detailed results are in Appendix C.6.

ate the same frozen encoder using mean-pooled features and an attentive probe, obtaining 43.4 and 62.0, respectively. The lower mean-pooled result is consistent with these features not being directly regularized in our setup.

Takeaway. λ-JEPA extends effectively to video SSL, with its clearest gains on temporal evaluations, showing that the benefits of backbone spectral regularization transfer beyond image pretraining.

4.4 Controlled experiments on ImageNet-100

We next use controlled ImageNet-100 experiments to better understand the effects of SACReg across SSL objectives. Our analysis focuses on two questions: how SACReg changes representation geometry and downstream performance, and what variation is recovered in the additional representation directions. To this end, we apply SACReg to the backbone representations of six existing augmentation-based SSL methods on ImageNet-100 (Tian et al., 2020a): LeJEPA (Balestriero & LeCun, 2025), VICReg (Bardes et al., 2022), VISReg (Wu et al., 2026), SimCLR (Chen et al., 2020), DINO (Caron et al., 2021), and BYOL (Grill et al., 2020). For each method, we keep its original objective and training recipe unchanged and add only $\beta_{\mathrm{h}}\,\mathrm{SACReg}$ to the backbone representation. The regularizer weight is set using the gradient based calibration procedure (see Appendix C.3 for details) and is not tuned individually to the methods except DINO. For DINO we tune the weight separately due to its sensitivity to the gradient ratios.

Figure 2 (top) shows that applying SACReg to the backbones produces a consistent change in the representations across objectives. At the backbone, RankMe increases substantially for every method, while the cosine geometry becomes less concentrated. Negative-pair cosine similarities approach zero while positive-pairs generally decrease as well. In contrast, corresponding statistics in the projection/loss space change much less. This suggests that SACReg primarily reshapes the backbone representation without substantially impeding the learning after the projection. The same intervention also improves downstream performance. kNN accuracy improves across all methods convincingly, while linear probe accuracy also improves, with smaller gains for VICReg. Notably, the gains therefore do not come from making positive pairs more invariant.

Augmentation thickness We next distinguish variation between images from variation across augmented views of the same image. Let Q denote a source image and $\mathbf{h}$ its backbone representation under a random augmentation. We define $\mathbf{B}_h = \mathrm{Cov}\left(\mathbb{E}[\mathbf{h} \mid Q]\right)$ and $\mathbf{A}_h = \mathbb{E}\left[\mathrm{Cov}(\mathbf{h} \mid Q)\right]$, where $\mathbf{B}_h$ captures variation between image centers, with $r_h = \mathrm{rank}(\mathbf{B}_h)$, and $\mathbf{A}_h$ captures variation across views of the same image. We then define the *augmentation thickness*

$$\mathbf{\Theta}_h = \mathbf{B}_h^{\dagger/2}\mathbf{A}_h\mathbf{B}_h^{\dagger/2}, \qquad \theta_h(\mathbf{a}) = \frac{\mathbf{a}^\top\mathbf{A}_h\mathbf{a}}{\mathbf{a}^\top\mathbf{B}_h\mathbf{a}}, \tag{4}$$

so thickness measures augmentation variation relative to the separation of image centers along that direction. Appendix E shows that thickness has a two-sided role: too little can discard meaningful variation across augmentations, while too much allows within-image variation to dominate the separation between images.

To study thickness, we use the class indicators. For each class c, let $\mathbf{a}_c$ be its linear prediction from whitened image centers and $d_h(c)^2$ the corresponding error. We measure class-center separability by $1 - d_h(c)^2$ and view sensitivity by $\mathbf{a}_c^\top\mathbf{\Theta}_h(\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1}\mathbf{a}_c$. Details are in Appendix C.

Figure 2 (bottom) shows that, averaged across classes, all six SSL methods have greater class-center separability and greater view sensitivity under SACReg. Therefore, SACReg's downstream gains are not obtained by suppressing augmentation-dependent variation. Specifically, the more consistent effect is an improvement in how image centers are organized along downstream-relevant directions. For λ-JEPA, class-center separability improves for 96 of 100 classes, whereas view sensitivity moves in both directions. SACReg therefore consistently improves class-center separation without uniformly moving the representation toward either invariance or augmentation sensitivity.

Takeaway. Across diverse SSL objectives, SACReg consistently increases representation rank and improves downstream performance. The augmentation thickness analysis shows that these gains are not explained by increased augmentation invariance. SACReg improves class-center separation while preserving augmentation variation.

5 Conclusion

In this paper, we targeted the asymmetry between anti-collapse regularization in joint embedding self-supervised models and the representation used for downstream tasks. First, we showed that preventing collapse in the projected representation does not guarantee a high rank backbone representation. This matters because a low rank backbone reduces the representational capacity, which can harm downstream performance. Our solution is directly motivated by a linear analysis of negative layer balance, which is known as λ balance in the feature learning literature. From this, we introduced SACReg, which directly regularizes the representation covariance, and used it to build λ-JEPA. In controlled experiments, SACReg consistently increases backbone rank and improves downstream performance while preserving augmentation-dependent variation. At scale, λ-JEPA narrows the gap between explicitly regularized JEPA methods and DINO, while improving frozen transfer across image and video SSL benchmarks.

SACReg regularizes CLS tokens and uses global views. Extending it to patch-level features and local crops is non-trivial: local views introduce substantially larger augmentation-induced variation, making the per-image view centers noisier and the covariance across centers harder to regularize reliably. In our experiments, this led to unstable training. Understanding how different augmentations shape this variation and which augmentation-sensitive directions are useful for downstream tasks remains an open direction for future work. Developing a formulation that more robustly regularizes spatially local representations while separating augmentation variation from between-image variation could further improve performance on dense prediction tasks. Additionally, JEPA-style models have already been connected with world modeling and identifiabilty theory for learning causal representations (Klindt et al., 2026). This is conceptually natural, as JEPA models optimize the same principles of causal representations: invariance to relevant transformations and anti-collapse (Yao et al., 2025). However, methods in causal representation learning leverage mostly negative samples (and sometimes reconstruction) to implement the anti-collapse properties. We encourage future work to explore the role of our regularizer in the identifiability of world models.

AI USE STATEMENT

We used generative AI tools as coding assistants to support the implementation and modification of the codebase and for language editing. These tools were not used to autonomously generate research results or replace the authors' scientific judgment. The authors determined the research methodology and experimental design, reviewed and validated all AI-assisted implementations.

ETHICS STATEMENT

This work studies methods for self-supervised representation learning using established image and video datasets and does not involve human-subject experiments or the collection of new personal or sensitive data. We are not aware of specific ethical concerns introduced by the proposed method beyond those generally associated with large-scale representation learning and computer vision models. The work is intended as a general-purpose methodological contribution and we encourage appropriate consideration of dataset provenance, bias and privacy risks when deploying learned representations.

REPRODUCIBILITY STATEMENT

We provide detailed information required to reproduce our theoretical and empirical results throughout the paper and appendices. The assumptions, statements and proofs supporting the theoretical results are provided in Appendix A. For the empirical results, implementation details for SACReg are given in Appendix B, Appendix C includes information about datasets, architectures, training procedures, loss-weight calibration, evaluation pipelines and baselines used in our experiments. Additional analyses and their definitions are provided in Appendices D and E. We reuse public baseline implementations and checkpoints whenever available, and explicitly document differences in training and evaluation protocols when exact matching is not possible. In particular, for video SSL, some baselines use different frozen-feature evaluation protocols (e.g., CLS, mean-pooled, or attentive probes); we report the corresponding protocols and evaluate our representations under multiple probe choices to facilitate comparison. Furthermore, we provide supplementary material containing the code used to run every experiment reported in the paper. These materials are intended to enable reproduction of both the proposed method and the empirical evaluations.

REFERENCES

Nicolas Anguita, Francesco Locatello, Andrew M Saxe, Marco Mondelli, Flavia Mancini, Samuel Lippl, and Clementine Domine. A theory of how pretraining shapes inductive bias in fine-tuning. *arXiv preprint arXiv:2602.20062*, 2026.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning, 2025. URL https://arxiv.org/abs/2506.09985.

Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1NvflqAdoom.

Shahar Azulay, Edward Moroshko, Mor Shpigel Nacson, Blake E Woodworth, Nathan Srebro, Amir Globerson, and Daniel Soudry. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. In *International Conference on Machine Learning*, pp. 468–477. PMLR, 2021.

Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

A. Bardes, J. Ponce, and Y. LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022.

Florian Bordes, Randall Balestriero, Quentin Garrido, Adrien Bardes, and Pascal Vincent. Guillotine regularization: Why removing layers is needed to improve generalization in self-supervised learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=ZgXfXSz51n.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Lukas Braun, Clémentine Carla Juliette Dominé, James E Fitzgerald, and Andrew M Saxe. Exact learning dynamics of deep linear networks with prior knowledge. In *Advances in Neural Information Processing Systems*, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, October 2021.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*, 2020.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on learning theory*, pp. 1305–1338. PMLR, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

Clémentine Carla Juliette Dominé, Nicolas Anguita, Alexandra Maria Proca, Lukas Braun, Daniel Kunin, Pedro A. M. Mediano, and Andrew M Saxe. From lazy to rich: Exact learning dynamics in deep linear networks. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZXaocmXc6d.

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

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pp. 1675–1685. PMLR, 2019.

Linus Ericsson, Henry Gouk, and Timothy M. Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5414–5423, June 2021.

Kenji Fukumizu. Effect of batch learning in multilayer neural networks. In *International Conference on Neural Information Processing*, 1998. URL https://api.semanticscholar.org/CorpusID:605683.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. RankMe: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *Proceedings of the International Conference on Machine Learning*, 2023.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Devon Jarvis, Sebastian Lee, Clémentine Carla Juliette Dominé, Andrew M Saxe, and Stefano Sarao Mannelli. A theory of initialisation's impact on specialisation. *Journal of Statistical Mechanics: Theory and Experiment*, 2025(11):114001, 2025.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=YevsQ05DEN7.

Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2021.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

David Klindt, Yann LeCun, and Randall Balestriero. When does lejepa learn a world model? *arXiv preprint arXiv:2605.26379*, 2026.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *https://cave.cs.toronto.edu/kriz/learning-features-2009-TR.pdf*, 2009.

Lukas Kuhn, Lucas Maes, Giuseppe Serra, Quentin Le Lidec, Yann LeCun, Randall Balestriero, and Florian Buettner. Levjepa: Efficient & scalable video pretraining without the heuristics. *arXiv preprint arXiv:2608.27395*, 2026.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vt5mnLVIVo.

Daniel Kunin, Allan Raventós, Clémentine Dominé, Feng Chen, David Klindt, Andrew Saxe, and Surya Ganguli. Get rich quick: exact solutions reveal how unbalanced initializations promote rapid feature learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 81157–81203. Curran Associates, Inc., 2024. doi: 10.52202/079017-2580. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/94074dd5a072d28ff75a76dabed43767-Paper-Conference.pdf.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Unlocking the potential of image vits for video understanding. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1632–1643. IEEE, 2023.

Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=AHOs7Sm5H7R.

Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang. Phase diagram for two-layer relu neural networks at infinite-width limit. *Journal of Machine Learning Research*, 22(71):1–47, 2021.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

Yoonsoo Nam, Seok Hyeong Lee, Clémentine Carla Juliette Dominé, Yeachan Park, Charles London, Wonyl Choi, Niclas Alexander Göring, and Seungjai Lee. Position: Solve layer-wise linear models first to understand neural dynamical phenomena (Neural collapse, emergence, Lazy/Rich regime, and grokking). In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 81897–81929. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/nam25a.html.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3498–3505. IEEE, 2012.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://openreview.net/forum?id=_wzZwKpTDF_9C.

Andrew M Saxe, James L McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116 (23):11537–11546, 2019.

Igor Susmelj, Matthias Heller, Philipp Wirth, Jeremy Prescott, and Malte Ebner. Lightly, 2020. URL https://github.com/lightly-ai/lightly.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020b.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14549–14560. IEEE, 2023.

Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673. PMLR, 2020.

Haiyu Wu, Randall Balestriero, and Morgan Levine. VISReg: Variance-invariance-sketching regularization for JEPA training. *arXiv preprint arXiv:2606.02572*, 2026.

Dingling Yao, Dario Rancati, Riccardo Cadei, Marco Fumero, and Francesco Locatello. Unifying causal representation learning with the invariance principle. In *International Conference on Learning Representations*, volume 2025, pp. 53847–53890, 2025.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ydopy-e6Dg.

A A NON-COLLAPSE REGULARIZER INSPIRED BY THE FEATURE-LEARNING LITERATURE

A.1 LINEAR NETWORK

A.1.1 PROBLEM SETUP

Consider a supervised learning problem with training data $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^{P}, \mathbf{x}_n \in \mathbb{R}^{N_i}, \mathbf{y}_n \in \mathbb{R}^{N_o}$. Let $\mathbf{X} = [\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_P] \in \mathbb{R}^{N_i \times P}$, and $\mathbf{Y} = [\mathbf{y}_1 \quad \cdots \quad \mathbf{y}_P] \in \mathbb{R}^{N_o \times P}$. We consider a two-layer linear network $\widehat{\mathbf{y}}_n = \mathbf{W}_2\mathbf{W}_1\mathbf{x}_n$, with encoder and decoder matrices $\mathbf{W}_1 \in \mathbb{R}^{N_h \times N_i}$, $\mathbf{W}_2 \in \mathbb{R}^{N_o \times N_h}$. The corresponding end-to-end mapping is $\mathbf{M} = \mathbf{W}_2\mathbf{W}_1$.

The empirical mean-squared error is

$$\mathcal{L}_{\text{task}}(\mathbf{W}_1, \mathbf{W}_2) = \frac{1}{2P} \left\| \mathbf{W}_2\mathbf{W}_1\mathbf{X} - \mathbf{Y} \right\|_F^2 . \tag{5}$$

Throughout this section, we consider continuous-time gradient flow, with time constant $\tau > 0$:

$$\tau\dot{\mathbf{W}}_1 = -\nabla_{\mathbf{W}_1}\mathcal{L}_{\text{task}}, \qquad \tau\dot{\mathbf{W}}_2 = -\nabla_{\mathbf{W}_2}\mathcal{L}_{\text{task}}. \tag{6}$$

Definition A.1 (λ-balancedness)**.** *Define the layer imbalance matrix by*

$$\mathbf{\Delta}(\mathbf{W}_1, \mathbf{W}_2) = \mathbf{W}_2^\top\mathbf{W}_2 - \mathbf{W}_1\mathbf{W}_1^\top \in \mathbb{R}^{N_h \times N_h}. \tag{7}$$

The network is said to be λ_{bal}-balanced if $\mathbf{\Delta}(\mathbf{W}_1, \mathbf{W}_2) = \lambda_{\text{bal}}\mathbf{I}_{N_h}$. The case $\lambda_{\text{bal}} = 0$ corresponds to standard balancedness.

A.1.2 BALANCE CONSERVATION WITHOUT EXPLICIT REGULARIZATION

For completeness we first recall the balance-conservation property of two-layer linear networks trained without explicit regularization, also derived in Dominé et al. (2025).

Lemma A.2 (Conservation of layer imbalance)**.** *Suppose that the network is trained by gradient flow on a differentiable loss of the form $\mathcal{L}_{\text{task}}(\mathbf{W}_1, \mathbf{W}_2) = \ell(\mathbf{W}_2\mathbf{W}_1)$, without weight decay or a log-determinant regularizer. Then*

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{W}_2^\top\mathbf{W}_2 - \mathbf{W}_1\mathbf{W}_1^\top\right) = \mathbf{0}. \tag{8}$$

Consequently, if the network is λ_{bal}-balanced at initialization, then it remains λ_{bal}-balanced throughout gradient-flow training.

Proof. Let $\mathbf{G} = \nabla_{\mathbf{M}}\ell(\mathbf{M})\big|_{\mathbf{M}=\mathbf{W}_2\mathbf{W}_1}$. The gradient-flow equations for the unregularized task loss are

$$\tau\dot{\mathbf{W}}_1 = -\mathbf{W}_2^\top\mathbf{G}, \qquad \tau\dot{\mathbf{W}}_2 = -\mathbf{G}\mathbf{W}_1^\top.$$

Therefore,

$$\tau \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{W}_2^\top \mathbf{W}_2\right) = -\mathbf{W}_1 \mathbf{G}^\top \mathbf{W}_2 - \mathbf{W}_2^\top \mathbf{G} \mathbf{W}_1^\top, \tag{9}$$

$$\tau \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{W}_1 \mathbf{W}_1^\top\right) = -\mathbf{W}_2^\top \mathbf{G} \mathbf{W}_1^\top - \mathbf{W}_1 \mathbf{G}^\top \mathbf{W}_2. \tag{10}$$

The two expressions are identical. Subtracting them proves equation 8. □

Remark A.3 (Gradient flow versus gradient descent). The conservation law in Lemma A.2 is exact under continuous-time gradient flow. For finite-step gradient descent, additional terms of order η^2 generally appear, so exact conservation need not hold for a nonzero learning rate η.

A.1.3 Non-collapse property of balanced initialization

We first show that a negative layer balance is sufficient to prevent encoder collapse. Under a positive-definiteness assumption on the input covariance, it also prevents collapse of the hidden representations.

Lemma A.4 (Non-collapse under negative layer balance). *Let $\mathbf{W}_1 \in \mathbb{R}^{N_h \times N_i}$ and $\mathbf{W}_2 \in \mathbb{R}^{N_o \times N_h}$ satisfy*

$$\mathbf{W}_2^\top \mathbf{W}_2 - \mathbf{W}_1 \mathbf{W}_1^\top = \lambda_{\mathrm{bal}} \mathbf{I}_{N_h}, \qquad \lambda_{\mathrm{bal}} < 0. \tag{11}$$

Then $\mathbf{W}_1 \mathbf{W}_1^\top \succeq -\lambda_{\mathrm{bal}} \mathbf{I}_{N_h}$. Consequently, $\mathbf{W}_1$ has full row rank, with $\sigma_{\min}(\mathbf{W}_1) \geq \sqrt{-\lambda_{\mathrm{bal}}}$.

Furthermore, let $\bar{\mathbf{x}} = P^{-1} \sum_{n=1}^{P} \mathbf{x}_n$ and suppose that the centered empirical input covariance satisfies $\mathbf{\Sigma}_x = \frac{1}{P} \sum_{n=1}^{P} (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^\top \succeq \kappa_x \mathbf{I}_{N_i}$ for some $\kappa_x > 0$. For the hidden representations $\mathbf{h}_n = \mathbf{W}_1 \mathbf{x}_n$, let $\bar{\mathbf{h}} = P^{-1} \sum_{n=1}^{P} \mathbf{h}_n$. Their centered empirical covariance then satisfies

$$\boxed{\mathbf{C}_h = \frac{1}{P} \sum_{n=1}^{P} (\mathbf{h}_n - \bar{\mathbf{h}})(\mathbf{h}_n - \bar{\mathbf{h}})^\top \succeq -\kappa_x \lambda_{\mathrm{bal}} \mathbf{I}_{N_h}}. \tag{12}$$

Proof. Rearranging equation 11 gives

$$\mathbf{W}_1 \mathbf{W}_1^\top = \mathbf{W}_2^\top \mathbf{W}_2 - \lambda_{\mathrm{bal}} \mathbf{I}_{N_h} \succeq -\lambda_{\mathrm{bal}} \mathbf{I}_{N_h} \succ 0,$$

since $\mathbf{W}_2^\top \mathbf{W}_2 \succeq 0$ and $\lambda_{\mathrm{bal}} < 0$. This establishes the encoder rank and singular-value bounds.

By linearity, $\bar{\mathbf{h}} = \mathbf{W}_1 \bar{\mathbf{x}}$, so $\mathbf{h}_n - \bar{\mathbf{h}} = \mathbf{W}_1 (\mathbf{x}_n - \bar{\mathbf{x}})$. Therefore,

$$\mathbf{C}_h = \mathbf{W}_1 \mathbf{\Sigma}_x \mathbf{W}_1^\top \succeq \kappa_x \mathbf{W}_1 \mathbf{W}_1^\top \succeq -\kappa_x \lambda_{\mathrm{bal}} \mathbf{I}_{N_h} \succ 0,$$

which proves the representation non-collapse bound. □

Thus, the representation covariance is positive definite, with variance at least $-\kappa_x \lambda_{\mathrm{bal}} > 0$ in every unit direction. This rules out *complete collapse*, in which all hidden representations coincide ($\mathbf{C}_h = \mathbf{0}$), and, more generally, *dimensional collapse*, in which the hidden representations lie in a proper affine subspace of $\mathbb{R}^{N_h}$ ($\mathrm{rank}(\mathbf{C}_h) < N_h$).

Combining Lemma A.2 with Lemma A.4, we conclude that a network initialized with a negative layer balance retains the corresponding non-collapse bounds throughout unregularized gradient-flow training. This guarantee depends on the balance imposed at initialization.

A.1.4 Constrained optimization characterization

Lemma A.4 identifies negative layer balance as a sufficient condition for non-collapse. We next show that the proposed regularizer selects precisely such a balance among factorizations of a fixed end-to-end mapping. The regularizer arises from a constrained minimum-regularization problem.

Theorem A.5 (Balancedness via Constrained Optimization). *Fix an end-to-end mapping $\mathbf{M} \in \mathbb{R}^{N_o \times N_i}$, $\gamma > 0$, $\lambda_{\mathrm{reg}} > 0$ and consider*

$$\begin{aligned} \min_{\mathbf{W}_1, \mathbf{W}_2} \quad & \frac{\lambda_{\mathrm{reg}}}{2}\left(\|\mathbf{W}_1\|_F^2 + \|\mathbf{W}_2\|_F^2\right) - \frac{\gamma}{2} \log\det\left(\mathbf{W}_1 \mathbf{W}_1^\top\right) \\ \textit{subject to} \quad & \mathbf{W}_2 \mathbf{W}_1 = \mathbf{M}. \end{aligned} \tag{13}$$

Let $(\mathbf{W}_1^\star, \mathbf{W}_2^\star)$ *be a constrained minimizer with* $\mathbf{W}_1^\star(\mathbf{W}_1^\star)^\top \succ 0$*. Then*

$$\boxed{(\mathbf{W}_2^\star)^\top \mathbf{W}_2^\star - \mathbf{W}_1^\star(\mathbf{W}_1^\star)^\top = -\frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h}}. \tag{14}$$

Proof. Introduce a Lagrange multiplier $\mathbf{\Lambda} \in \mathbb{R}^{N_o \times N_i}$ for the constraint $\mathbf{W}_2\mathbf{W}_1 = \mathbf{M}$. The Lagrangian is

$$\begin{aligned}\mathscr{L}(\mathbf{W}_1, \mathbf{W}_2, \mathbf{\Lambda}) = \frac{\lambda_{\text{reg}}}{2}\left(\|\mathbf{W}_1\|_F^2 + \|\mathbf{W}_2\|_F^2\right) - \frac{\gamma}{2}\log\det\left(\mathbf{W}_1\mathbf{W}_1^\top\right) \\ + \langle \mathbf{\Lambda}, \mathbf{W}_2\mathbf{W}_1 - \mathbf{M}\rangle_F .\end{aligned} \tag{15}$$

The first-order stationarity conditions are

$$\lambda_{\text{reg}}\mathbf{W}_1 - \gamma\left(\mathbf{W}_1\mathbf{W}_1^\top\right)^{-1}\mathbf{W}_1 + \mathbf{W}_2^\top\mathbf{\Lambda} = \mathbf{0}, \tag{16}$$

$$\lambda_{\text{reg}}\mathbf{W}_2 + \mathbf{\Lambda}\mathbf{W}_1^\top = \mathbf{0}. \tag{17}$$

Multiplying equation 16 on the right by $\mathbf{W}_1^\top$ gives

$$\lambda_{\text{reg}}\mathbf{W}_1\mathbf{W}_1^\top - \gamma\mathbf{I}_{N_h} + \mathbf{W}_2^\top\mathbf{\Lambda}\mathbf{W}_1^\top = \mathbf{0}. \tag{18}$$

Multiplying equation 17 on the left by $\mathbf{W}_2^\top$ gives

$$\lambda_{\text{reg}}\mathbf{W}_2^\top\mathbf{W}_2 + \mathbf{W}_2^\top\mathbf{\Lambda}\mathbf{W}_1^\top = \mathbf{0}. \tag{19}$$

Eliminating the common multiplier term $\mathbf{W}_2^\top\mathbf{\Lambda}\mathbf{W}_1^\top$ yields $\lambda_{\text{reg}}\left(\mathbf{W}_1\mathbf{W}_1^\top - \mathbf{W}_2^\top\mathbf{W}_2\right) = \gamma\mathbf{I}_{N_h}$, which is equivalent to equation 14. □

Corollary A.6 (Non-collapse at constrained regularizer minima)**.** *Under the assumptions of Theorem A.5,*

$$\boxed{\mathbf{W}_1^\star(\mathbf{W}_1^\star)^\top \succeq \frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h}}. \tag{20}$$

Consequently, the encoder has full row rank, with $\sigma_{\min}(\mathbf{W}_1^\star) \geq \sqrt{\frac{\gamma}{\lambda_{\text{reg}}}}$.

If, in addition, the centered empirical input covariance $\mathbf{\Sigma}_x$ *defined in Lemma A.4 satisfies* $\mathbf{\Sigma}_x \succeq \kappa_x\mathbf{I}_{N_i}$ *for some* $\kappa_x > 0$*, then the centered empirical covariance of the hidden representations* $\mathbf{h}_n^\star = \mathbf{W}_1^\star\mathbf{x}_n$ *satisfies*

$$\boxed{\mathbf{C}_h^\star = \mathbf{W}_1^\star\mathbf{\Sigma}_x(\mathbf{W}_1^\star)^\top \succeq \kappa_x\frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h}}. \tag{21}$$

Proof. Theorem A.5 gives

$$(\mathbf{W}_2^\star)^\top\mathbf{W}_2^\star - \mathbf{W}_1^\star(\mathbf{W}_1^\star)^\top = -\frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h}.$$

Since $\gamma > 0$ and $\lambda_{\text{reg}} > 0$, this is a negative layer balance. Both bounds therefore follow directly from Lemma A.4 with $\lambda_{\text{bal}} = -\gamma/\lambda_{\text{reg}} < 0$. □

Thus, the representation covariance is positive definite, preventing both complete and dimensional collapse.

A.1.5 Balance selected by the encoder-side regularizer

For arbitrary $\mathbf{\Delta}$, the regularizer exponentially drives the network toward the selected balance.

Theorem A.7 (Global existence and exponential convergence of layer balance)**.** *Consider gradient flow with time scale* $\tau > 0$ *on the regularized mean-squared-error objective in equation 3.2, with* $\lambda_{\text{reg}} > 0$ *and* $\gamma > 0$*. If* $\mathbf{W}_1(0)\mathbf{W}_1(0)^\top \succ 0$*, then the solution exists for all* $t \geq 0$ *and the encoder remains full row rank.*

Define

$$\mathbf{\Delta}(t) = \mathbf{W}_2(t)^\top \mathbf{W}_2(t) - \mathbf{W}_1(t)\mathbf{W}_1(t)^\top, \qquad \mathbf{\Delta}_\star = -\frac{\gamma}{\lambda_{\text{reg}}}\mathbf{I}_{N_h}.$$

The imbalance satisfies

$$\tau\dot{\mathbf{\Delta}}(t) = -2\lambda_{\text{reg}}\big(\mathbf{\Delta}(t) - \mathbf{\Delta}_\star\big),$$

and hence

$$\boxed{\mathbf{\Delta}(t) - \mathbf{\Delta}_\star = e^{-2\lambda_{\text{reg}}t/\tau}\big(\mathbf{\Delta}(0) - \mathbf{\Delta}_\star\big)}. \tag{22}$$

Thus, for every admissible initialization, the layer imbalance converges exponentially to $\mathbf{\Delta}_\star$ at rate $2\lambda_{\text{reg}}/\tau$, selecting the asymptotic balance $\lambda_{\text{bal}}^{\text{enc}} = -\gamma/\lambda_{\text{reg}}$.

Proof. We first establish global existence and preservation of the encoder full row rank. Along gradient flow,

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathcal{L}_{\text{LSAC}} = -\frac{1}{\tau}\left(\|\nabla_{\mathbf{W}_1}\mathcal{L}_{\text{LSAC}}\|_F^2 + \|\nabla_{\mathbf{W}_2}\mathcal{L}_{\text{LSAC}}\|_F^2\right) \le 0.$$

Writing σ_i for the encoder singular values, the regularizer is

$$\mathcal{R}_{\text{LSAC}} = \frac{\lambda_{\text{reg}}}{2}\|\mathbf{W}_2\|_F^2 + \sum_{i=1}^{N_h}\left(\frac{\lambda_{\text{reg}}}{2}\sigma_i^2 - \gamma\log\sigma_i\right).$$

Each scalar summand is bounded below and diverges at both zero and infinity. Since the mean-squared-error task loss is nonnegative, the initial objective sublevel set bounds the weights and keeps every encoder singular value away from zero. The trajectory thus remains in a compact subset of the full-row-rank domain, where the gradient field is smooth. Standard ODE continuation gives global existence and $\mathbf{W}_1(t)\mathbf{W}_1(t)^\top \succ 0$ for all $t \ge 0$.

Now define

$$\mathbf{G} = \nabla_{\mathbf{M}}\ell(\mathbf{M})\big|_{\mathbf{M}=\mathbf{W}_2\mathbf{W}_1}, \qquad \mathbf{H} = \mathbf{W}_1\mathbf{W}_1^\top.$$

Since $\mathbf{H} \succ 0$, it is symmetric and invertible. The gradient of the encoder-side log-determinant term is

$$\nabla_{\mathbf{W}_1}\left[-\frac{\gamma}{2}\log\det\left(\mathbf{W}_1\mathbf{W}_1^\top\right)\right] = -\gamma\mathbf{H}^{-1}\mathbf{W}_1. \tag{23}$$

The gradient-flow equations are therefore

$$\tau\dot{\mathbf{W}}_1 = -\mathbf{W}_2^\top\mathbf{G} - \lambda_{\text{reg}}\mathbf{W}_1 + \gamma\mathbf{H}^{-1}\mathbf{W}_1, \tag{24}$$

$$\tau\dot{\mathbf{W}}_2 = -\mathbf{G}\mathbf{W}_1^\top - \lambda_{\text{reg}}\mathbf{W}_2. \tag{25}$$

We first compute the dynamics of the decoder Gram matrix. By the product rule,

$$\tau\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{W}_2^\top\mathbf{W}_2\right) = \left(\tau\dot{\mathbf{W}}_2\right)^\top\mathbf{W}_2 + \mathbf{W}_2^\top\left(\tau\dot{\mathbf{W}}_2\right).$$

Substituting equation 25 gives

$$\begin{aligned}\tau\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{W}_2^\top\mathbf{W}_2\right) &= \left(-\mathbf{G}\mathbf{W}_1^\top - \lambda_{\text{reg}}\mathbf{W}_2\right)^\top\mathbf{W}_2 + \mathbf{W}_2^\top\left(-\mathbf{G}\mathbf{W}_1^\top - \lambda_{\text{reg}}\mathbf{W}_2\right)\\ &= -\mathbf{W}_1\mathbf{G}^\top\mathbf{W}_2 - \mathbf{W}_2^\top\mathbf{G}\mathbf{W}_1^\top - 2\lambda_{\text{reg}}\mathbf{W}_2^\top\mathbf{W}_2.\end{aligned} \tag{26}$$

Similarly, the encoder Gram matrix satisfies

$$\tau\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{W}_1\mathbf{W}_1^\top\right) = \left(\tau\dot{\mathbf{W}}_1\right)\mathbf{W}_1^\top + \mathbf{W}_1\left(\tau\dot{\mathbf{W}}_1\right)^\top.$$

Substituting equation 24, we obtain

$$\begin{aligned}\tau\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{W}_1\mathbf{W}_1^\top\right) = &\left(-\mathbf{W}_2^\top\mathbf{G} - \lambda_{\text{reg}}\mathbf{W}_1 + \gamma\mathbf{H}^{-1}\mathbf{W}_1\right)\mathbf{W}_1^\top\\ &+ \mathbf{W}_1\left(-\mathbf{W}_2^\top\mathbf{G} - \lambda_{\text{reg}}\mathbf{W}_1 + \gamma\mathbf{H}^{-1}\mathbf{W}_1\right)^\top\\ = &-\mathbf{W}_2^\top\mathbf{G}\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{G}^\top\mathbf{W}_2 - 2\lambda_{\text{reg}}\mathbf{W}_1\mathbf{W}_1^\top\\ &+ \gamma\mathbf{H}^{-1}\mathbf{W}_1\mathbf{W}_1^\top + \gamma\mathbf{W}_1\mathbf{W}_1^\top\mathbf{H}^{-1}.\end{aligned} \tag{27}$$

Because $\mathbf{H} = \mathbf{W}_1\mathbf{W}_1^\top$, we have

$$\mathbf{H}^{-1}\mathbf{W}_1\mathbf{W}_1^\top = \mathbf{I}_{N_h}, \qquad \mathbf{W}_1\mathbf{W}_1^\top\mathbf{H}^{-1} = \mathbf{I}_{N_h}.$$

Therefore,

$$\tau\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{W}_1\mathbf{W}_1^\top\right) = -\mathbf{W}_2^\top\mathbf{G}\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{G}^\top\mathbf{W}_2 - 2\lambda_{\mathrm{reg}}\mathbf{W}_1\mathbf{W}_1^\top + 2\gamma\mathbf{I}_{N_h}. \tag{28}$$

Recall that

$$\mathbf{\Delta} = \mathbf{W}_2^\top\mathbf{W}_2 - \mathbf{W}_1\mathbf{W}_1^\top.$$

Subtracting equation 28 from equation 26 yields

$$\begin{aligned}\tau\dot{\mathbf{\Delta}} = &\left(-\mathbf{W}_1\mathbf{G}^\top\mathbf{W}_2 - \mathbf{W}_2^\top\mathbf{G}\mathbf{W}_1^\top - 2\lambda_{\mathrm{reg}}\mathbf{W}_2^\top\mathbf{W}_2\right)\\ &-\left(-\mathbf{W}_2^\top\mathbf{G}\mathbf{W}_1^\top - \mathbf{W}_1\mathbf{G}^\top\mathbf{W}_2 - 2\lambda_{\mathrm{reg}}\mathbf{W}_1\mathbf{W}_1^\top + 2\gamma\mathbf{I}_{N_h}\right)\\ = &-2\lambda_{\mathrm{reg}}\left(\mathbf{W}_2^\top\mathbf{W}_2 - \mathbf{W}_1\mathbf{W}_1^\top\right) - 2\gamma\mathbf{I}_{N_h}.\end{aligned} \tag{29}$$

The task-loss terms cancel exactly, leaving

$$\boxed{\tau\dot{\mathbf{\Delta}} = -2\lambda_{\mathrm{reg}}\mathbf{\Delta} - 2\gamma\mathbf{I}_{N_h}}.$$

It remains to solve this linear matrix differential equation. Dividing by τ gives

$$\dot{\mathbf{\Delta}} + \frac{2\lambda_{\mathrm{reg}}}{\tau}\mathbf{\Delta} = -\frac{2\gamma}{\tau}\mathbf{I}_{N_h}.$$

Multiplying both sides by the integrating factor $e^{2\lambda_{\mathrm{reg}}t/\tau}$ yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[e^{2\lambda_{\mathrm{reg}}t/\tau}\mathbf{\Delta}(t)\right] = -\frac{2\gamma}{\tau}e^{2\lambda_{\mathrm{reg}}t/\tau}\mathbf{I}_{N_h}.$$

Integrating from 0 to t, we obtain

$$\begin{aligned}e^{2\lambda_{\mathrm{reg}}t/\tau}\mathbf{\Delta}(t) - \mathbf{\Delta}(0) &= -\frac{2\gamma}{\tau}\int_0^t e^{2\lambda_{\mathrm{reg}}s/\tau}\,\mathrm{d}s\,\mathbf{I}_{N_h}\\ &= -\frac{\gamma}{\lambda_{\mathrm{reg}}}\left(e^{2\lambda_{\mathrm{reg}}t/\tau} - 1\right)\mathbf{I}_{N_h}.\end{aligned}$$

Multiplying by $e^{-2\lambda_{\mathrm{reg}}t/\tau}$ gives

$$\mathbf{\Delta}(t) = e^{-2\lambda_{\mathrm{reg}}t/\tau}\mathbf{\Delta}(0) - \frac{\gamma}{\lambda_{\mathrm{reg}}}\left(1 - e^{-2\lambda_{\mathrm{reg}}t/\tau}\right)\mathbf{I}_{N_h}.$$

Equivalently, defining $\mathbf{\Delta}_\star = -(\gamma/\lambda_{\mathrm{reg}})\mathbf{I}_{N_h}$,

$$\boxed{\mathbf{\Delta}(t) - \mathbf{\Delta}_\star = e^{-2\lambda_{\mathrm{reg}}t/\tau}\left(\mathbf{\Delta}(0) - \mathbf{\Delta}_\star\right)}.$$

Since $\lambda_{\mathrm{reg}} > 0$ and $\tau > 0$, the imbalance converges exponentially to $\mathbf{\Delta}_\star$ at rate $2\lambda_{\mathrm{reg}}/\tau$. □

Corollary A.8 (Persistence of the selected balance)**.** *If the network is initialized such that* $\mathbf{\Delta}(0) = -\frac{\gamma}{\lambda_{\mathrm{reg}}}\mathbf{I}_{N_h}$*, then* $\mathbf{\Delta}(t) = -\frac{\gamma}{\lambda_{\mathrm{reg}}}\mathbf{I}_{N_h}$ *for all* $t \geq 0$*. Thus, the balance condition selected by the regularizer is invariant under the regularized gradient-flow dynamics.*

Proof. Substituting the assumed initial condition into equation 22 gives the result immediately. □

A.1.6 SINGULAR-MODE CHARACTERIZATION

The previous results establish the selected balance and the anti-collapse property without requiring an explicit factorization. For completeness, we now give the corresponding allocation of each end-to-end singular mode across the two layers, also derived in Dominé et al. (2025). We later show that the regularization leads to the balanced factorization (see Appendix A.1.7).

Theorem A.9 (Singular-mode allocation)**.** *Assume $N_h \leq \min\{N_i, N_o\}$, and let $\mathbf{M} = \mathbf{USV}^\top$, where $\mathbf{S} = \mathrm{diag}(s_1, \ldots, s_{N_h})$,with $s_i \geq 0$, defines an N_h-mode singular value decomposition, padded with zero singular values when $\mathrm{rank}(\mathbf{M}) < N_h$. Define $\lambda_{bal} = -\frac{\gamma}{\lambda_{\mathrm{reg}}}$. Up to an orthogonal transformation $\mathbf{R} \in \mathbb{R}^{N_h \times N_h}$ of the hidden layer and rotations within degenerate singular subspaces, a minimizer of Theorem A.5 can be written as $\mathbf{W}_1^\star = \mathbf{RAV}^\top$, $\mathbf{W}_2^\star = \mathbf{UBR}^\top$, where $\mathbf{A} = \mathrm{diag}(a_1, \ldots, a_{N_h})$, $\mathbf{B} = \mathrm{diag}(b_1, \ldots, b_{N_h})$, and*

$$a_i^2 = \frac{-\lambda_{bal} + \sqrt{\lambda_{bal}^2 + 4s_i^2}}{2}, \qquad b_i^2 = \frac{\lambda_{bal} + \sqrt{\lambda_{bal}^2 + 4s_i^2}}{2}. \tag{30}$$

In particular, $a_i b_i = s_i$, and $b_i^2 - a_i^2 = \lambda_{bal}$.

Proof. According to Eq. 14 $(\mathbf{W}_2^\star)^\top \mathbf{W}_2^\star - \mathbf{W}_1^\star (\mathbf{W}_1^\star)^\top = \lambda_{bal} \mathbf{I}_{N_h}$. The two Gram matrices therefore commute and share an orthogonal eigenbasis R. Consequently, the factors admit the aligned representation with $\mathbf{B}^2 - \mathbf{A}^2 = \lambda_{bal} \mathbf{I}_{N_h}$.

The product constraint gives

$$\mathbf{W}_2^\star \mathbf{W}_1^\star = \mathbf{UBAV}^\top = \mathbf{USV}^\top,$$

and hence $\mathbf{BA} = \mathbf{S}$. For each singular direction i, we therefore have

$$a_i b_i = s_i, \qquad b_i^2 - a_i^2 = \lambda_{bal}.$$

Setting $x_i = b_i^2$ gives $x_i(x_i - \lambda_{bal}) = s_i^2$. The solution is

$$b_i^2 = \frac{\lambda_{bal} + \sqrt{\lambda_{bal}^2 + 4s_i^2}}{2},$$

and subtracting λ_{bal} gives

$$a_i^2 = \frac{-\lambda_{bal} + \sqrt{\lambda_{bal}^2 + 4s_i^2}}{2}.$$

□

A.1.7 Rich and lazy regime

The ratio $-\gamma/\lambda_{\mathrm{reg}}$ determines the selected layer balance and thereby influences the learning regime. For a fixed end-to-end mapping, it governs how each singular mode is distributed between the encoder and decoder. Zero balance ($\lambda_{\mathrm{bal}} = 0$) corresponds to equal layer singular values, whereas nonzero balance introduces an asymmetry. The effect of this balance on learning depends on the architecture, initialization scale, and sign of λ_{bal} (Dominé et al., 2025). In the linear setting considered here, sufficiently strong negative imbalance can favor lazy dynamics while preserving the anti-collapse guarantee. By contrast, zero-balanced initialization can support rich dynamics at sufficiently small initialization scales. The relationship between imbalance and learning regime is more complex in nonlinear networks, where full matrix imbalance is not conserved and does not necessarily induce lazy learning. Appropriate asymmetric parameter scalings can instead accelerate the onset of rich feature-learning dynamics while preserving the non-collapse property (Anguita et al., 2026; Kumar et al., 2024; Kunin et al., 2024). We discuss this distinction further in Appendix A.2.3.

A.1.8 Experimental verification

We empirically verify Theorem 3.1 (and its dynamical counterpart, Theorem A.7) in a minimal two-layer linear network with no architectural bottleneck, $N_i = N_h = N_o = 8$, so that $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{8\times 8}$ are both square and (generically) invertible.

Task. We use an orthogonal input design $\mathbf{X} = \sqrt{N_i}\,\mathbf{I}_{N_i}$ and a fixed $\{0, \pm 1/\sqrt{N_i}\}$-valued target matrix $\mathbf{Y}$ with hierarchical block structure. The resulting least-squares mapping has three distinct singular values (multiplicities $2, 2, 4$, reflecting the hierarchy), letting us check the predicted balance across multiple directions at once.

Training protocol. In Fig. 3 (A), both layers are initialized with unbalanced i.i.d. Gaussian weights, $\mathbf{W}_1(0), \mathbf{W}_2(0) \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 0.5$. The network is trained by full-batch gradient descent on

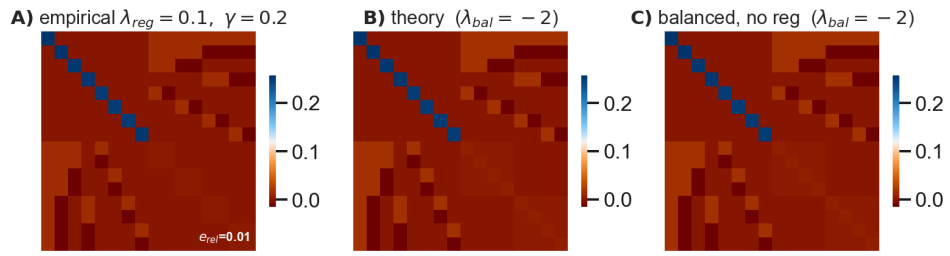

Figure 3: Block Gram matrix $\mathbf{Q} = \begin{pmatrix} \mathbf{W}_1^\top \mathbf{W}_1 & M^\top \\ M & \mathbf{W}_2 \mathbf{W}_2^\top \end{pmatrix}$, $M = \mathbf{W}_2 \mathbf{W}_1$, for a two-layer linear network. **(A)** Training with symmetric L_2 weight decay ($\lambda_{\text{reg}} = 0.1$) and an encoder-side negative log-determinant penalty ($\gamma = 0.2$). The regularizer drives the layer imbalance $\mathbf{\Delta} = \mathbf{W}_2^\top \mathbf{W}_2 - \mathbf{W}_1 \mathbf{W}_1^\top$ towards $\mathbf{\Delta}_\star = -(\gamma/\lambda_{\text{reg}})\mathbf{I}_{N_h} = -2\mathbf{I}_{N_h}$, corresponding to $\lambda_{\text{bal}} = -2$. **(B)** Closed-form prediction from Appendix A.1.6. **(C)** Unregularized control network initialized at $\lambda_{\text{bal}} = -2$. This balance is conserved under gradient flow, providing a comparison with the balance selected dynamically in (A).

$\mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{task}} + \mathcal{R}_{\text{LSAC}}$ ($\eta = 10^{-3}, T = 3 \times 10^4$ steps), with the log-determinant term on the *encoder* Gram matrix, $-\gamma \log\det(\mathbf{W}_1 \mathbf{W}_1^T)$, and weight decay λ_{reg} on both layers ($(\lambda_{reg}, \gamma) = (0.1, 0.2)$).

Verification protocol. At $t = T$, we compare three block Gram matrices $\mathbf{Q} = \begin{pmatrix} \mathbf{W}_1^\top \mathbf{W}_1 & M^\top \\ M & \mathbf{W}_2 \mathbf{W}_2^\top \end{pmatrix}$, where $M = \mathbf{W}_2 \mathbf{W}_1$, **(A)** $\mathbf{Q}_{\text{emp}}$ from the regularized network; **(B)** $\mathbf{Q}_{\text{theory}}$, obtained by Theorem A.9 Appendix A.1.6, with $\lambda_{\text{bal}} = -\gamma/\lambda_{reg} = -2$; and **(C)** $\mathbf{Q}_{\text{bal}}$, from a network initialized exactly at this λ_{bal}-balanced solution and trained unregularized ($\lambda_{reg} = \gamma = 0$), whose balance is conserved by construction under unregularized gradient flow.

Fig. 3 shows near-perfect agreement across all three panels. The empirical imbalance eigenvalues concentrate tightly around the predicted, negative fixed point, $\lambda_i(\mathbf{\Delta}(T)) \in [-2.005, -1.984] \approx \lambda_{\text{bal}}$, and the small relative error,

$$e_{rel} = \frac{\|\mathbf{Q}_{\text{emp}} - \mathbf{Q}_{\text{theory}}\|_F}{\|\mathbf{Q}_{\text{theory}}\|_F} = 0.0095,$$

confirms that the encoder-side log-determinant regularizer drives the network towards the predicted, collapse-preventing balanced fixed point, without impeding task fit (final MSE $\approx 5.47 \times 10^{-4}$).

A.2 From linear to nonlinear covariance regularization

A.2.1 From linear balance to nonlinear covariance regularization

For a linear encoder with centered, whitened inputs and $\varepsilon = 0$, the covariance penalty reduces exactly to the encoder portion of the linear regularizer, with $\lambda_{\text{cov}} = \lambda_{\text{reg}}$.

Lemma A.10 (Linear encoder mean and covariance)**.** *Let $\mathbf{x} \in \mathbb{R}^{N_i}$ have finite second moments, mean $\boldsymbol{\mu}_x = \mathbb{E}[\mathbf{x}]$, and covariance $\mathbf{\Sigma}_x = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu}_x)(\mathbf{x} - \boldsymbol{\mu}_x)^\top]$. For the bias-free linear representation $\mathbf{h} = \mathbf{W}_1 \mathbf{x}$, its mean and covariance satisfy $\boldsymbol{\mu} = \mathbf{W}_1 \boldsymbol{\mu}_x$, $\mathbf{C}_h = \mathbf{W}_1 \mathbf{\Sigma}_x \mathbf{W}_1^\top$.*

In particular, for centered, whitened inputs, $\boldsymbol{\mu}_x = \mathbf{0}$ and $\mathbf{\Sigma}_x = \mathbf{I}_{N_i}$, so $\boldsymbol{\mu} = \mathbf{0}$, $\mathbf{C}_h = \mathbf{W}_1 \mathbf{W}_1^\top$, and $\operatorname{tr}(\mathbf{C}_h) = \|\mathbf{W}_1\|_F^2$.

Proof. By linearity of expectation, $\boldsymbol{\mu} = \mathbb{E}[\mathbf{W}_1 \mathbf{x}] = \mathbf{W}_1 \boldsymbol{\mu}_x$. Consequently, $\mathbf{h} - \boldsymbol{\mu} = \mathbf{W}_1(\mathbf{x} - \boldsymbol{\mu}_x)$, giving $\mathbf{C}_h = \mathbb{E}[(\mathbf{h} - \boldsymbol{\mu})(\mathbf{h} - \boldsymbol{\mu})^\top] = \mathbf{W}_1 \mathbf{\Sigma}_x \mathbf{W}_1^\top$. For centered, whitened inputs, these identities reduce to $\boldsymbol{\mu} = 0$ and $\mathbf{C}_h = \mathbf{W}_1 \mathbf{W}_1^\top$. Taking the trace yields $\operatorname{tr}(\mathbf{C}_h) = \|\mathbf{W}_1\|_F^2$. □

Recovery of the linear encoder regularizer. Recall that the nonlinear regularizer is

$$\mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) = \frac{\lambda_{\text{mean}}}{2} \|\boldsymbol{\mu}\|_2^2 + \frac{\lambda_{\text{cov}}}{2} \operatorname{tr}(\mathbf{C}_h) - \frac{\gamma}{2} \log\det(\mathbf{C}_h + \varepsilon \mathbf{I}_{N_h}).$$

Assume $\mathbf{W}_1\mathbf{W}_1^\top \succ 0$, centered and whitened inputs, $\varepsilon = 0$, and $\lambda_{\text{cov}} = \lambda_{\text{reg}}$. Lemma A.10 then gives

$$\mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) = \frac{\lambda_{\text{reg}}}{2}\|\mathbf{W}_1\|_F^2 - \frac{\gamma}{2}\log\det(\mathbf{W}_1\mathbf{W}_1^\top).$$

The mean penalty vanishes because $\boldsymbol{\mu} = \mathbf{0}$, regardless of the positive value of λ_{mean}. Thus, the nonlinear regularizer reduces exactly to the encoder portion of the linear regularizer:

$$\mathcal{R}_{\text{LSAC}}(\mathbf{W}_1, \mathbf{W}_2) = \mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) + \frac{\lambda_{\text{reg}}}{2}\|\mathbf{W}_2\|_F^2.$$

Role of the mean penalty. For nonlinear encoders, centered inputs need not produce centered representations. Moreover, covariance is invariant under a common translation of all representations, so neither the trace nor the log-determinant term controls their mean. The additional $\|\boldsymbol{\mu}\|_2^2$ penalty controls this translation by favoring zero-mean representations. It therefore extends the regularizer without changing its reduction to the bias-free linear setting with centered, whitened inputs. The same identities hold for empirical means and covariances.

A.2.2 Log-determinant covariance regularizer with the mean

Let $\mathbf{h} = f_\theta(\mathbf{x}) \in \mathbb{R}^{N_h}$ denote the representation produced by a possibly nonlinear encoder. For a minibatch of size B, collect the representations row-wise in $\mathbf{H} \in \mathbb{R}^{B\times N_h}$. Define the empirical mean, centering matrix, centered representations, and empirical covariance as $\boldsymbol{\mu} = \frac{1}{B}\mathbf{H}^\top\mathbf{1}$, $\mathbf{P}_B = \mathbf{I}_B - \frac{1}{B}\mathbf{1}\mathbf{1}^\top$, $\widetilde{\mathbf{H}} = \mathbf{P}_B\mathbf{H}$, $\mathbf{C}_h = \frac{1}{B}\widetilde{\mathbf{H}}^\top\widetilde{\mathbf{H}} \succeq 0$. The linear analysis never needs to control the representation's mean: for a bias-free linear encoder $\mathbf{h} = \mathbf{W}_1\mathbf{x}$ with centered input, $\mathbb{E}[\mathbf{h}] = \mathbf{W}_1\,\mathbb{E}[\mathbf{x}] = \mathbf{0}$ identically, so the mean carries no independent degree of freedom for $\mathcal{R}_{\text{SAC}}$ to regularize. This guarantee does not survive the transfer to a general nonlinear (or biased) encoder: a bias term, or the nonlinearity itself (e.g. a ReLU clipping one side of an otherwise symmetric pre-activation), can give $\mathbf{h}$ a nonzero mean that moves independently of $\mathbf{C}_h$ – invisible to both $\text{tr}(\mathbf{C}_h)$ and $\log\det(\mathbf{C}_h)$, since $\mathbf{C}_h$ is computed from the *centered* representation by construction. We therefore regularize $\boldsymbol{\mu}$ directly alongside $\mathbf{C}_h$. Motivated by the linear encoder regularizer, we give the following definition.

Definition A.11 (Nonlinear SACReg)**.** *Let $\lambda_{\text{mean}} > 0$, $\lambda_{\text{cov}} > 0$, $\gamma > 0$, and $\varepsilon \geq 0$. Assume that $\mathbf{C}_h + \varepsilon\mathbf{I}_{N_h} \succ 0$. We define*

$$\mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) = \frac{\lambda_{\text{mean}}}{2}\|\boldsymbol{\mu}\|_2^2 + \frac{\lambda_{\text{cov}}}{2}\,\text{tr}(\mathbf{C}_h) - \frac{\gamma}{2}\log\det\left(\mathbf{C}_h + \varepsilon\mathbf{I}_{N_h}\right). \tag{31}$$

The mean term controls the representation's location, the trace term controls its overall scale, and the negative log-determinant penalizes small covariance eigenvalues. The regularizer constrains only the representation mean and covariance; it does not directly constrain higher-order moments or enforce Gaussianity.

To understand the inductive bias of the regularizer, we consider its minimizer in isolation. This does not imply that the mean and covariance of the complete learning objective must attain this value, since the task loss may introduce competing constraints. First, we write the regularizer in spectral form.

Proposition A.12 (Spectral form)**.** *Let $\lambda_1(\mathbf{C}_h), \ldots, \lambda_{N_h}(\mathbf{C}_h)$ be the eigenvalues of $\mathbf{C}_h$. Then*

$$\mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) = \frac{\lambda_{\text{mean}}}{2}\|\boldsymbol{\mu}\|^2 + \frac{1}{2}\sum_{i=1}^{N_h}\left[\lambda_{\text{cov}}\lambda_i(\mathbf{C}_h) - \gamma\log\left(\lambda_i(\mathbf{C}_h) + \varepsilon\right)\right]. \tag{32}$$

Proof. Because $\mathbf{C}_h$ is symmetric and positive semidefinite, it admits an eigendecomposition $\mathbf{C}_h = \mathbf{Q}\,\text{diag}(\lambda_1, \ldots, \lambda_{N_h})\mathbf{Q}^\top$. Consequently, $\text{tr}(\mathbf{C}_h) = \sum_{i=1}^{N_h}\lambda_i$ and $\log\det(\mathbf{C}_h + \varepsilon\mathbf{I}_{N_h}) = \sum_{i=1}^{N_h}\log(\lambda_i + \varepsilon)$. The mean term does not depend on $\mathbf{C}_h$. Substitution into equation 31 proves the result. □

We can now write the optimal mean and covariance.

Theorem A.13 (Optimal mean and covariance)**.** *The unique minimizer of* $\mathcal{R}_{\mathrm{SAC}}$ *over* $\boldsymbol{\mu} \in \mathbb{R}^{N_h}$, $\mathbf{C}_h \succeq 0$ *such that* $\mathbf{C}_h + \varepsilon \mathbf{I}_{N_h} \succ 0$ *is*

$$\boxed{\boldsymbol{\mu}^\star = \mathbf{0}, \qquad \mathbf{C}_h^\star = \left(\frac{\gamma}{\lambda_{\mathrm{cov}}} - \varepsilon\right)_+ \mathbf{I}_{N_h}}, \tag{33}$$

where $(a)_+ = \max\{a, 0\}$*. Therefore, the covariance-level optimum is full-rank if and only if* $\gamma > \lambda_{\mathrm{cov}}\varepsilon$*, independently of* $\boldsymbol{\mu}^\star$*.*

Proof. By Proposition A.12, $\mathcal{R}_{\mathrm{SAC}}$ is a sum of a term depending only on $\boldsymbol{\mu}$ and a term depending only on $\mathbf{C}_h$, so the two may be minimized separately. The mean term $\frac{\lambda_{\mathrm{mean}}}{2}\|\boldsymbol{\mu}\|^2$ is uniquely minimized at $\boldsymbol{\mu}^\star = \mathbf{0}$. For the covariance term, each eigenvalue independently minimizes the strictly convex function

$$\phi(s) = \frac{\lambda_{\mathrm{cov}}}{2} s - \frac{\gamma}{2}\log(s + \varepsilon), \qquad s \geq 0 \quad , s + \varepsilon \geq 0$$

Its derivatives are $\phi'(s) = \frac{\lambda_{\mathrm{cov}}}{2} - \frac{\gamma}{2(s+\varepsilon)}$ and $\phi''(s) = \frac{\gamma}{2(s+\varepsilon)^2} > 0$. The unconstrained stationary point satisfies $\phi'(s) = 0 \Leftrightarrow s = \frac{\gamma}{\lambda_{\mathrm{cov}}} - \varepsilon$. If $\gamma/\lambda_{\mathrm{cov}} - \varepsilon > 0$, this stationary point is the unique minimizer. Otherwise, $\varepsilon > 0$, so $s = 0$ is admissible and is the unique minimizer. □

Corollary A.14 (Isotropic non-collapsed target, centered)**.** *In the idealized case* $\varepsilon = 0$*,*

$$\boxed{\boldsymbol{\mu}^\star = \mathbf{0}, \qquad \mathbf{C}_h^\star = \frac{\gamma}{\lambda_{\mathrm{cov}}}\mathbf{I}_{N_h}}. \tag{34}$$

Thus, the regularizer simultaneously:

1. *centers the representation at the origin;*
2. *prevents the covariance eigenvalues from vanishing;*
3. *equalizes the covariance spectrum;*
4. *controls the overall representation scale; and*
5. *selects a condition number equal to one.*

Next we derive the non-collapse property of the regularizer.

Theorem A.15 (Mean boundedness and strict covariance non-collapse)**.** *Suppose that* $\varepsilon = 0$*. Then*

$$\lambda_{\min}(\mathbf{C}_h) \to 0 \quad \Longrightarrow \quad \mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \to +\infty, \tag{35}$$

$$\lambda_{\max}(\mathbf{C}_h) \to +\infty \quad \Longrightarrow \quad \mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \to +\infty, \tag{36}$$

$$\|\boldsymbol{\mu}\| \to \infty \quad \Longrightarrow \quad \mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \to +\infty. \tag{37}$$

Consequently, any fixed finite upper bound on the regularizer bounds the mean and keeps all covariance eigenvalues bounded above and away from zero. Specifically, for every finite K*, there exist constants* $0 < m_K \leq M_K < \infty$ *and* $0 \leq M_K' < \infty$ *such that*

$$R_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \leq K \quad \Longrightarrow \quad m_K \mathbf{I}_{N_h} \preceq \mathbf{C}_h \preceq M_K \mathbf{I}_{N_h}, \qquad \|\boldsymbol{\mu}\|_2 \leq M_K'. \tag{38}$$

Proof. Suppose that $\varepsilon = 0$ and $\mathbf{C}_h \succ 0$. Let $\lambda_1(\mathbf{C}_h), \ldots, \lambda_{N_h}(\mathbf{C}_h) > 0$ denote the eigenvalues of $\mathbf{C}_h$. Since $\mathrm{tr}(\mathbf{C}_h) = \sum_{i=1}^{N_h} \lambda_i(\mathbf{C}_h)$ and $\log\det(\mathbf{C}_h) = \sum_{i=1}^{N_h} \log \lambda_i(\mathbf{C}_h)$, write

$$\mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) = \frac{\lambda_{\mathrm{mean}}}{2}\|\boldsymbol{\mu}\|^2 + \sum_{i=1}^{N_h} \phi\left(\lambda_i(\mathbf{C}_h)\right), \qquad \phi(s) = \frac{\lambda_{\mathrm{cov}}}{2} s - \frac{\gamma}{2}\log s, \quad s > 0.$$

We first show that ϕ is bounded below. Its derivatives are $\phi'(s) = \frac{\lambda_{\mathrm{cov}}}{2} - \frac{\gamma}{2s}$ and $\phi''(s) = \frac{\gamma}{2s^2} > 0$, so ϕ is strictly convex, with unique stationary point $s = \gamma/\lambda_{\mathrm{cov}}$ and global minimum

$$b := \min_{s>0} \phi(s) = \phi\left(\frac{\gamma}{\lambda_{\mathrm{cov}}}\right) = \frac{\gamma}{2} - \frac{\gamma}{2}\log\left(\frac{\gamma}{\lambda_{\mathrm{cov}}}\right).$$

In particular, $\phi(s) \geq b$ for every $s > 0$, so $\sum_{i=1}^{N_h} \phi(\lambda_i(\mathbf{C}_h)) \geq N_h b$ for every $\mathbf{C}_h \succ 0$.

Covariance collapse. Without loss of generality order the eigenvalues so that $\lambda_1(\mathbf{C}_h) = \lambda_{\min}(\mathbf{C}_h)$. Then

$$\mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \;\geq\; \phi\left(\lambda_{\min}(\mathbf{C}_h)\right) + (N_h - 1)b.$$

As $s \downarrow 0$, $\frac{\lambda_{\mathrm{cov}}}{2} s \to 0$ and $-\frac{\gamma}{2} \log s \to +\infty$, so $\lim_{s \downarrow 0} \phi(s) = +\infty$. Since $(N_h - 1)b$ is a fixed finite constant, $\lambda_{\min}(\mathbf{C}_h) \to 0 \Rightarrow \mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \to +\infty$, proving equation 35.

Covariance blowup. As $s \to +\infty$, $\frac{\lambda_{\mathrm{cov}}}{2} s \to +\infty$ dominates $-\frac{\gamma}{2} \log s$, so $\phi(s) \to +\infty$; the same argument with $\lambda_{\max}(\mathbf{C}_h)$ in place of $\lambda_{\min}(\mathbf{C}_h)$ proves equation 36.

Mean blowup. Since $\phi(s) \geq b$ for every $s > 0$, we have

$$\mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \geq \frac{\lambda_{\mathrm{mean}}}{2} \|\boldsymbol{\mu}\|_2^2 + N_h b \qquad \text{for every } \mathbf{C}_h \succ 0.$$

Because $\lambda_{\mathrm{mean}} > 0$ and b is finite, $\|\boldsymbol{\mu}\|_2 \to \infty$ implies $\mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \to +\infty$, even when $\mathbf{C}_h$ varies. This proves equation 37.

The preceding limits imply that, for any fixed finite K, the condition $\mathcal{R}_{\mathrm{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h) \leq K$ keeps the covariance eigenvalues bounded above and away from zero, and bounds $\|\boldsymbol{\mu}\|$. Otherwise, a sequence satisfying this condition would approach covariance collapse, unbounded covariance, or unbounded mean, forcing the regularizer to diverge and contradicting its upper bound K. This proves equation 38. □

Remark A.16 (Effect of the numerical stabilizer). The regularizer prevents covariance collapse when $\varepsilon = 0$, but only discourages it when $\varepsilon > 0$. In the latter case, its unique covariance-level minimizer is still full-rank provided $\gamma > \lambda_{\mathrm{cov}} \varepsilon$ (Theorem A.13). Mean boundedness remains guaranteed in both cases.

Remark A.17 (Scope of the nonlinear extension). The nonlinear covariance regularizer is motivated by the exact linear balance analysis, but it does not imply that the linear balance identity $\mathbf{W}_2^\top \mathbf{W}_2 - \mathbf{W}_1 \mathbf{W}_1^\top = -\frac{\gamma}{\lambda_{\mathrm{reg}}} \mathbf{I}_{N_h}$ continues to hold for a general nonlinear encoder. The nonlinear parameter dynamics are modified by the encoder Jacobian, and there is generally no conserved difference between the decoder Gram matrix and the representation covariance.

A.2.3 Feature learning: Two-layer ReLU experiment

Task and data. We study nonlinear teacher–student regression with a fixed, untrained two-layer ReLU teacher of hidden width $N_{h_0} = 3$. The teacher is

$$f_\star(\mathbf{x}) = \sum_{j=1}^{N_{h_0}} a_j^\star \sigma\big((\mathbf{w}_j^\star)^\top \mathbf{x}\big), \qquad \sigma(z) = \max\{0, z\},$$

where $\mathbf{w}_j^\star$ are sampled independently and uniformly from the unit sphere $\mathbb{S}^{d-1}$, and $a_j^\star$ are independent Rademacher random variables. For each run, we sample $n = 1{,}000$ training inputs independently from $\mathbb{S}^{N_i - 1}$ and assign noiseless labels $y_i = f_\star(\mathbf{x}_i)$. A separate test set of $n_{\mathrm{test}} = 10{,}000$ inputs, generated with seed 201, is reused at every checkpoint for both test-loss and representation-rank measurements.

Student architecture. The student is a two-layer, bias-free ReLU network with hidden width $N_h = 50$:

$$\mathbf{h}_\theta(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x}) \in \mathbb{R}^{N_h}, \qquad f_\theta(\mathbf{x}) = \mathbf{a}^\top \mathbf{h}_\theta(\mathbf{x}).$$

We measure the rank of the post-ReLU representation $\mathbf{h}_\theta(\mathbf{x})$, the network's only hidden layer. The student uses a symmetric initialization: hidden units form $N_h/2$ pairs with identical input weights and opposite readout weights,

$$\mathbf{w}_{j+N_h/2} = \mathbf{w}_j, \qquad a_{j+N_h/2} = -a_j, \qquad j = 1, \ldots, N_h/2.$$

This ensures $f_{\theta_0}(\mathbf{x}) = \mathbf{0}$ for every input while retaining a nontrivial initial neural tangent kernel (NTK).

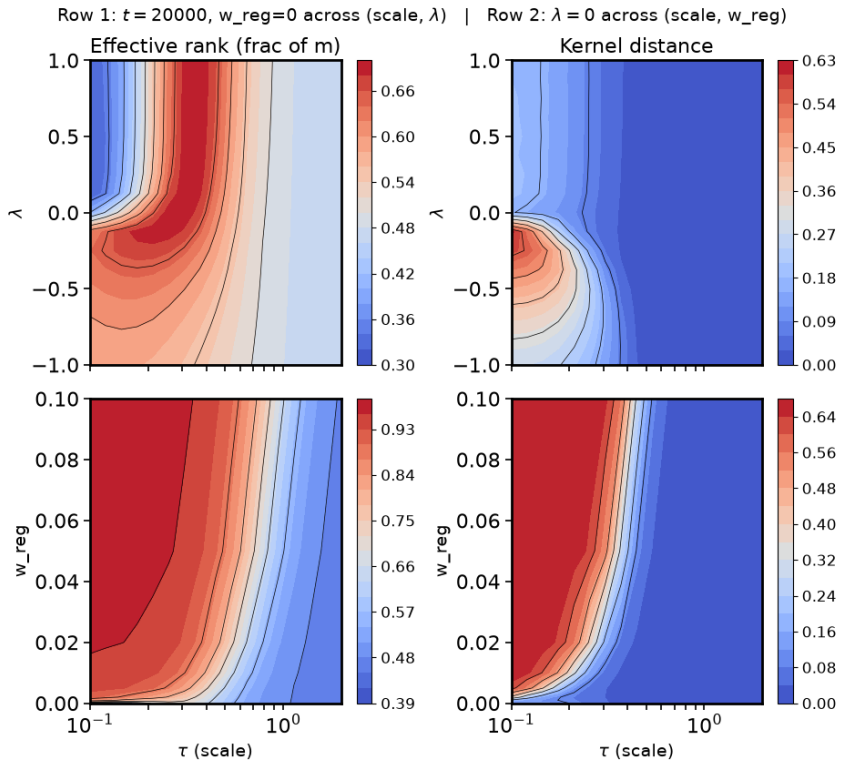

Figure 4: Effective rank and kernel distance of the NTK from initialization. **Top row:** Both metrics as functions of the scale τ and relative scale λ at step $t = 20{,}000$, without regularization ($w_{\text{reg}} = 0$). **Bottom row:** The same metrics at the final checkpoint as functions of the scale τ and regularization weight w_{reg}, with $\lambda = 0$ fixed.

Initialization. Two parameters control initialization: an overall scale $\tau > 0$ and a per-neuron imbalance λ. For each independently sampled pair, we initialize

$$\mathbf{w}_j = \frac{\tau}{\alpha}\mathbf{u}_j, \qquad a_j = \tau\alpha s_j,$$

where $\mathbf{u}_j$ is uniform on $\mathbb{S}^{N_i-1}$, $s_j \in \{-1, +1\}$ is a Rademacher random variable, and $\alpha > 0$ satisfies

$$\lambda = a_j^2 - \|\mathbf{w}_j\|_2^2 = \tau^2\left(\alpha^2 - \alpha^{-2}\right).$$

Equivalently,

$$\alpha = \left(\frac{\lambda/\tau^2 + \sqrt{(\lambda/\tau^2)^2 + 4}}{2}\right)^{1/2}.$$

Thus, negative λ corresponds to larger encoder weights, whereas positive λ corresponds to larger readout weights.

Objective and optimization. We train the student using full-batch gradient descent on the mean squared error,

$$\mathcal{L}_{\text{MSE}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(f_\theta(\mathbf{x}_i) - y_i\right)^2,$$

with an additional regularization term in the regularized experiments. The learning rate is scaled as $\eta = 5 \times 10^{-3}/\tau^2$. We apply the log-determinant covariance regularizer directly to the hidden representation of the toy two-layer ReLU student network,

$$\mathbf{h} = \text{ReLU}(\mathbf{W}_1\mathbf{x}) \in \mathbb{R}^{N_h}$$

Table 3: Effective rank fraction across training steps for different values of layer imbalance λ.

	Training step					
λ	0	100	1000	5000	10000	20000
-1.00	0.467	0.467	0.484	0.554	0.574	0.581
-0.75	0.467	0.467	0.487	0.563	0.587	0.595
-0.50	0.467	0.467	0.492	0.577	0.606	0.616
-0.25	0.467	0.469	0.500	0.596	0.627	0.639

(here $N_h = 50$), immediately after the nonlinearity and before the readout layer. We use the implementation of the regularizer described in Appendix B.1.

We use $\lambda_{\text{cov}} = \gamma$. Therefore, the regularized training objective is

$$\mathcal{L} = \mathcal{L}_{\text{task}} + w_{\text{reg}} \cdot \mathcal{R}_{\text{SAC}}(\boldsymbol{\mu}, \mathbf{C}_h),$$

with $\mathcal{L}_{\text{task}}$ the mean-squared error between the student's output and the teacher's labels, and $w_{\text{reg}} \in \{0, 0.001, 0.005, 0.02, 0.1\}$ swept as the regularizer strength.

Feature learning. Using the two-layer ReLU network setup of Kunin et al. (2024), we investigate how the rank effects and the learning regime of the initial imbalance λ relate to those of the SAC regularizer. Linear theory suggests that a negative relative scale should favor the same anti-collapse behavior promoted by SAC regularization.

- *Rank.* There is no exact correspondence between the initial imbalance λ and the regularization weight w_{reg}. The regularizer appears to exert a substantially stronger effect on rank (see Fig. 4). Nevertheless, both negative λ (relative to $\lambda \geq 0$) and positive w_{reg} favor higher rank in the Rich regime, supporting a qualitative connection between initial imbalance and anti-collapse regularization (see Fig. 4). Interestingly, the rank effect of negative λ emerges during training rather than reflecting a decaying initial offset. All negative-λ runs with $w_{\text{reg}} = 0$ begin at approximately the same effective rank fraction, `effrank_frac` $= 0.467$, consistent with their shared initialization (see Table 3). This quantity then increases monotonically throughout training, indicating that the higher final rank develops through the training dynamics.
- *Learning regime.* We characterize the learning regime by measuring the departure of the neural tangent kernel (NTK) from its value at initialization, quantified by kernel distance. Increasing w_{reg} leads to an increase in kernel distance across the initialization scales considered, indicating that stronger regularization promotes greater kernel evolution (see Fig. 4). It also extends the range of initialization scales over which feature learning occurs, although kernel evolution remains limited at sufficiently large scales (see Fig. 4). At small initialization scales, varying λ alone yields kernel distance values between 0.16 and 0.58, comparable to the range obtained by varying w_{reg}. At large initialization scales, however, both parameters have little effect on kernel evolution. These results suggest that λ primarily modulates the degree of feature learning within an already rich regime, whereas w_{reg} can also promote departures from otherwise lazy dynamics. Altogether, we find that regularization promotes feature learning across a range of initialization scales, with stronger regularization associated with more pronounced feature learning (see Fig. 4 in Appendix A.2.3). A similar trend emerges in the self-supervised setting. We show that both the backbone representations and the empirical NTK move substantially from initialization during training, consistent with rich learning dynamics (Fig. 5 and Appendix A.2.4).

A.2.4 Feature learning in the ViT

We additionally measure feature and kernel drift during self-supervised training on ImageNet-100. We train a ViT-S/16 for 100 epochs with SACReg applied to the backbone representation and evaluate all checkpoints on fixed held-out images.

Feature drift. For 4,096 validation images, let $\mathbf{X}_0, \mathbf{X}_t \in \mathbb{R}^{4096\times 384}$ denote the CLS representations at initialization and epoch t, respectively, at a given transformer block. We measure their

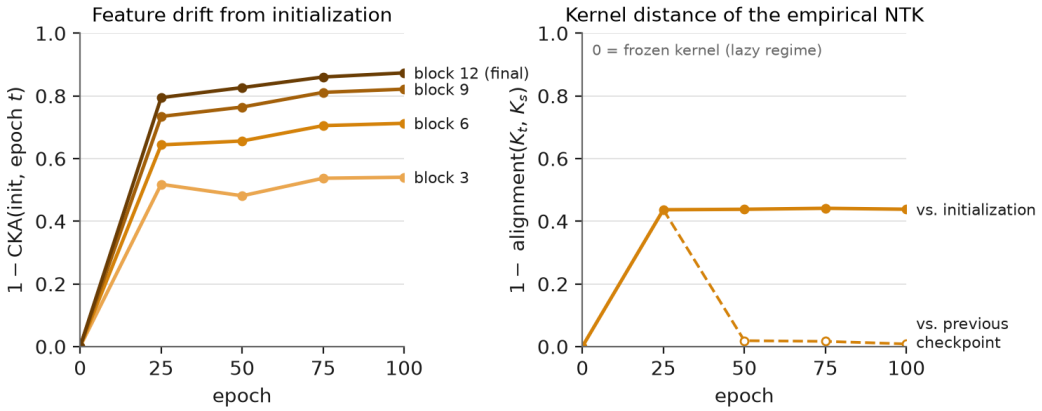

Figure 5: **Feature and kernel drift during self-supervised training.** ImageNet-100, ViT-S/16 trained for 100 epochs with SACReg applied at the backbone. **Left:** feature drift from initialization, measured by $1 - \mathrm{CKA}$ at the CLS token after transformer blocks 3, 6, 9, and 12. **Right:** empirical NTK distance from initialization (solid) and from the previous checkpoint (dashed). Representations drift increasingly with depth, while the NTK moves substantially early in training and subsequently stabilizes, consistent with rich rather than lazy learning dynamics.

dissimilarity using one minus linear CKA (Kornblith et al., 2019),

$$1 - \mathrm{CKA}(\mathbf{X}_0, \mathbf{X}_t), \qquad \mathrm{CKA}(\mathbf{X}, \mathbf{Y}) = \frac{\|\mathbf{X}_c^\top \mathbf{Y}_c\|_F^2}{\|\mathbf{X}_c^\top \mathbf{X}_c\|_F \, \|\mathbf{Y}_c^\top \mathbf{Y}_c\|_F},$$

where $\mathbf{X}_c$ and $\mathbf{Y}_c$ are column-centered. We report this quantity after blocks 3, 6, 9, and 12. Linear CKA is invariant to orthogonal transformations and isotropic rescaling, so the measure captures changes in representation geometry rather than changes in feature coordinates or overall scale.

Kernel drift. We estimate the empirical NTK of the backbone output with respect to the model parameters on a fixed subset of 256 validation images. To avoid constructing the full vector-valued NTK, we use eight fixed unit-norm random directions $\mathbf{v}_k \in \mathbb{R}^{384}$ in output space and compute

$$\mathbf{g}_{ik}^{(t)} = \nabla_\theta \big(\mathbf{v}_k^\top \mathbf{h}_\theta(\mathbf{x}_i)\big), \qquad \mathbf{K}_t[i,j] = \frac{1}{8} \sum_{k=1}^{8} \left\langle \mathbf{g}_{ik}^{(t)}, \mathbf{g}_{jk}^{(t)} \right\rangle.$$

The same images and random directions are used at every checkpoint. Up to a constant factor, $\mathbf{K}_t$ is an unbiased estimate of the trace NTK; this factor cancels in the normalized comparison. We measure kernel drift as

$$d(\mathbf{K}_t, \mathbf{K}_s) = 1 - \frac{\langle \mathbf{K}_t, \mathbf{K}_s \rangle_F}{\|\mathbf{K}_t\|_F \|\mathbf{K}_s\|_F},$$

and report distance both from initialization ($s = 0$) and from the preceding checkpoint. In the lazy-training limit, the NTK remains fixed, so its distance from initialization stays at zero.

Figure 5 shows substantial drift in both representations and the empirical NTK. Feature drift increases with depth, reaching its largest value at the final transformer block. The NTK moves away from its initialization during the first 25 epochs and then changes only slightly between subsequent checkpoints. Together, these measurements are consistent with rich feature learning.

B Implementation of the regularizer

Invariance objective. For image i with V augmented views, we use the mean squared distance between all pairs of projected representations,

$$\mathcal{L}_{\mathrm{SSL}} = \frac{1}{B} \sum_{i=1}^{B} \frac{2}{V(V-1)} \sum_{j<k} \frac{1}{d_z} \left\| \mathbf{z}_i^{(j)} - \mathbf{z}_i^{(k)} \right\|_2^2,$$

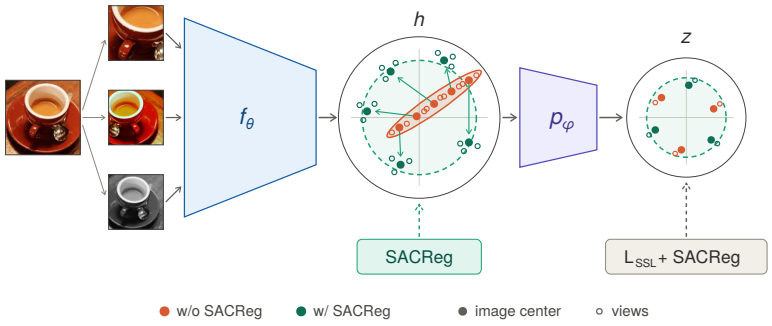

Figure 6: **Schematic overview of λ-JEPA.** Augmented views are mapped by the encoder (f_θ) to hidden representations (h), then by (p_φ) to embeddings (z). SACReg applies regularization directly in the hidden space to prevent per-view collapse, encouraging a more dispersed representation distribution (green) compared with training without SACReg (orange). Filled circles denote image centers; open circles denote individual views.

where B is the number of images in the batch and d_z is the projector dimension. This objective contains neither negatives nor an anti-collapse term. Using $\bar{\mathbf{z}}_i = \frac{1}{V}\sum_{j=1}^{V}\mathbf{z}_i^{(j)}$, it is equivalently $\frac{2V}{V-1}$ times the average squared distance from each view to its image center. All runs use global views of a single size: $V = 4$ on ImageNet-100 and $V = 6$ on ImageNet-1k and video. The invariance and regularization weights are set jointly as described in Appendix C.

Slicing. For B view centers $\mathbf{R} = \{\mathbf{r}_i\}_{i=1}^{B} \subset \mathbb{R}^d$, the centered empirical covariance has rank at most $B-1$. Thus, when $d \geq B$, the full log-determinant is degenerate. At each step, we draw a random orthonormal projection $\mathbf{U} \in \mathbb{R}^{d\times d'}$ using a thin QR factorization of a Gaussian matrix and evaluate SACReg on the projected centers $\{\mathbf{U}^\top\mathbf{r}_i\}_{i=1}^{B}$. The same projection is shared across GPUs and resampled every step. Within the slice, we compute the projected mean and covariance and use $\varepsilon = 10^{-4}$. We divide the regularizer by d', making its scale approximately independent of the slice width.

We use $d' = 128$ after the projector ($d_z = 256$). At the backbone, we use one third of the feature width: $d' = 128$ for ViT-S ($d = 384$) and $d' = 256$ for ViT-B ($d = 768$). The backbone and projector regularizers use independent random slices.

B.1 Relation to the full-covariance regularizer

For clarity, consider the idealized case $\varepsilon = 0$. Let $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma} \succ 0$ denote the mean and covariance of the view centers. The full regularizer per dimension is

$$\mathcal{R}_{\text{full}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2d}\left(\|\boldsymbol{\mu}\|_2^2 + \operatorname{tr}\boldsymbol{\Sigma} - d - \log\det\boldsymbol{\Sigma}\right).$$

For an orthonormal frame $\mathbf{U} \in \mathbb{R}^{d\times d'}$, the projected centers have mean $\mathbf{U}^\top\boldsymbol{\mu}$ and covariance $\mathbf{U}^\top\boldsymbol{\Sigma}\mathbf{U}$, giving the sliced regularizer

$$\mathcal{R}_{\mathbf{U}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2d'}\left(\|\mathbf{U}^\top\boldsymbol{\mu}\|_2^2 + \operatorname{tr}(\mathbf{U}^\top\boldsymbol{\Sigma}\mathbf{U}) - d' - \log\det(\mathbf{U}^\top\boldsymbol{\Sigma}\mathbf{U})\right).$$

Since a fresh Haar-random frame is sampled at every step, the corresponding expected slice objective is

$$\bar{\mathcal{R}}_{d'}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathbb{E}_{\mathbf{U}}[\mathcal{R}_{\mathbf{U}}(\boldsymbol{\mu}, \boldsymbol{\Sigma})].$$

Proposition B.1 (Random slicing)**.** *For every $1 \leq d' \leq d$:*

(i) $\bar{\mathcal{R}}_{d'} \geq 0$, with equality if and only if $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \mathbf{I}_d$.

(ii) The normalized mean and total variance are preserved in expectation:

$$\mathbb{E}_{\mathbf{U}}\frac{\|\mathbf{U}^\top\boldsymbol{\mu}\|_2^2}{d'} = \frac{\|\boldsymbol{\mu}\|_2^2}{d}, \qquad \mathbb{E}_{\mathbf{U}}\frac{\operatorname{tr}(\mathbf{U}^\top\boldsymbol{\Sigma}\mathbf{U})}{d'} = \frac{\operatorname{tr}\boldsymbol{\Sigma}}{d}.$$

(iii) The expected sliced objective approaches the full one monotonically:

$$\bar{\mathcal{R}}_{d'} \le \bar{\mathcal{R}}_{d'+1} \le \mathcal{R}_{\text{full}}, \qquad \bar{\mathcal{R}}_d = \mathcal{R}_{\text{full}}.$$

Thus, slicing does not change the preferred representation: averaging over random subspaces still uniquely favors zero mean and isotropic unit covariance. The mean and total variance are preserved exactly in expectation, while the log-determinant observes only the spectrum inside the sampled subspace. Wider slices therefore capture progressively more of the full spectral penalty, recovering the original regularizer when $d' = d$. In practice, we choose $d' < d$ so that the projected covariance can be estimated from substantially more centers than dimensions.

Proof. Each $\mathcal{R}_{\mathbf{U}}$ is nonnegative and is minimized when $\mathbf{U}^\top \boldsymbol{\mu} = \mathbf{0}$ and $\mathbf{U}^\top \boldsymbol{\Sigma} \mathbf{U} = \mathbf{I}_{d'}$. If this holds for all random subspaces, then every one-dimensional direction has zero projected mean and unit variance, implying $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \mathbf{I}_d$.

Haar invariance gives

$$\mathbb{E}_{\mathbf{U}}[\mathbf{U}\mathbf{U}^\top] = \frac{d'}{d}\mathbf{I}_d,$$

which directly yields (ii). Finally, concavity of the logarithm gives

$$\frac{1}{d'}\mathbb{E}_{\mathbf{U}} \log\det(\mathbf{U}^\top \boldsymbol{\Sigma} \mathbf{U}) \ge \frac{1}{d} \log\det \boldsymbol{\Sigma}.$$

Together with (ii), this gives $\bar{\mathcal{R}}_{d'} \le \mathcal{R}_{\text{full}}$. Applying the same argument to a random d'-dimensional subspace inside a random $(d'+1)$-dimensional subspace gives $\bar{\mathcal{R}}_{d'} \le \bar{\mathcal{R}}_{d'+1}$. For $d' = d$, $\mathbf{U}$ is orthogonal and the two objectives are identical.

Ring buffer. Reliable covariance estimation requires more centers than the slice dimension. For each regularized space, we therefore maintain a FIFO buffer containing detached view centers from the previous q steps and concatenate them with the current batch before computing the regularizer. This gives

$$B_{\text{eff}} = (q+1)B,$$

where only the current B centers carry gradients. Under data parallelism, the buffer contains centers from the full global batch.

We keep $B_{\text{eff}}/d' = 4$ throughout image training. Thus, $q = 3$ after the projector and at the ViT-S backbone ($B_{\text{eff}} = 512$, $d' = 128$), while $q = 7$ at the ViT-B backbone ($B_{\text{eff}} = 1024$, $d' = 256$). On video, the larger batch sizes reduce the need for buffering: no buffer is used for ViT-S, while ViT-B uses $q = 1$ at the backbone. We observed no measurable effect from the short parameter lag introduced by the buffered centers.

Averaged invariance target. For ImageNet-1k and video, we follow LeJEPA and compute the invariance target using an exponential moving average of the backbone and projector. The target $\tilde{\mathbf{z}}_i$ is the mean projected representation of the V views under the averaged network, and we use

$$\mathcal{L}_{\text{SSL}} = \frac{2V}{V-1} \frac{1}{BVd_z} \sum_{i=1}^{B} \sum_{j=1}^{V} \left\| \mathbf{z}_i^{(j)} - \tilde{\mathbf{z}}_i \right\|_2^2.$$

ImageNet-100 uses the pairwise objective directly. In all cases, SACReg is computed from view centers of the current network.

C Experimental Details

C.1 Datasets

ImageNet-1k (Russakovsky et al., 2015) is the standard split with 1.28M training and 50k validation images. ImageNet-100 is the 100-class subset introduced by CMC (Tian et al., 2020a), with 126,689 training and 5,000 validation images. For video we follow LeVJEPA (Kuhn et al., 2026): we take the union of the Kinetics-400, -600 and -700 (Li et al., 2023) training sets, remove duplicates and

every clip that appears in a validation set, and keep 20% of the clips of each class (120,100 clips after decoding). LeVJEPA does not release its subset, so ours is a different draw of the same size. Transfer (Ericsson et al., 2021) uses DTD (Cimpoi et al., 2014), Aircraft (Maji et al., 2013), Cars (Krause et al., 2013), CIFAR10, CIFAR100 (Krizhevsky, 2009), Flowers (Nilsback & Zisserman, 2008), Food (Bossard et al., 2014) and Pets (Parkhi et al., 2012) with the splits of the VISReg protocol (Wu et al., 2026); video encoders are evaluated on ImageNet-1k, Something-Something-v2 (Goyal et al., 2017) and Kinetics-400 (Kay et al., 2017).

C.2 Models and training

Large-scale image runs. We train ViT-S/16 and ViT-B/16 at 224^2 with a two-layer projector (hidden width 2048, output 256, BatchNorm), batch size 128 split over two GPUs, for 100 and 400 epochs. Each image gives $V = 6$ global views from the LeJEPA augmentation family: random resized crops covering 30–100% of the image, horizontal flips, color jitter, random grayscale, Gaussian blur, and solarization on one of the views. We do not use local crops. The slices and buffers are as described in Appendix B. Optimization uses AdamW with learning rate 10^{-3}, weight decay 0.05, 10 warm-up epochs followed by a cosine schedule, gradient clipping at 1.0, and bf16 precision. The loss weights are $(31.07,\ 225.50,\ 1.671)$ for ViT-S and $(22.81,\ 222.26,\ 4.054)$ for ViT-B, for the invariance term, β_z and β_h respectively; they are set as described below and shared by the 100- and 400-epoch runs.

Controlled experiments on ImageNet-100. All controlled experiments use ViT-S/16 at 224^2, are trained for 100 epochs with batch size 128, and are repeated over three independent random seeds. We re-implement LeJEPA, VICReg, SimCLR, DINO, BYOL, and VISReg within a common training codebase while preserving each method's released loss, projector, and augmentation settings. For each method, we train a baseline and a matched variant with $\beta_h \,\mathrm{SACReg}(\bar{\mathbf{H}})$ added; all other settings are kept fixed, so the two variants differ only by the backbone regularization term. We report the mean across the three seeds. The regularizer is applied to the representation provided as input to each method's projector: the CLS token for SimCLR, BYOL, VICReg, and DINO, and the 512-dimensional embedding immediately preceding the projector for LeJEPA and VISReg. Our own objective uses $V = 4$ views and the pairwise invariance term defined in Appendix B, with weights 32.8 for invariance, 157.8 for β_z, and 1.894 for β_h. The corresponding baseline sets $\beta_h = 0$, with all other settings unchanged.

Video. We use LeVJEPA's training pipeline and released configuration unchanged except for the loss: their video ViT-S/16 and ViT-B/16 (16 frames at stride 2, tubelet size 1, 95% token drop), their projector, and their optimizer (AdamW, learning rate 4×10^{-4}, weight decay 0.04, warm-up over 12% of training followed by a constant learning rate, bf16). Our loss takes $V = 6$ global 224^2 views of the same clip. The batch size is 768 for ViT-S and 512 for ViT-B, the loss weights are those of the image run of the same model size, and we evaluate the exponential-moving-average encoder, as LeVJEPA does.

C.3 Setting the loss weights

The loss terms have different forms and scales, so we do not balance them by their values but by their realized pull on the backbone. For a term with weight w, let g be the norm of the gradient of the unweighted term with respect to the backbone parameters, measured on a fixed batch at a given checkpoint. The product $w\,g$ is the pull of the term, and its share of the total pull is $w\,g / \sum_k w_k g_k$.

Adding the backbone term to an existing method. When adding $\beta_h \,\mathrm{SACReg}(\bar{\mathbf{H}})$ to an existing SSL method, we choose β_h so that the backbone regularizer contributes a small fraction $T = 0.06$ of the sum of weighted backbone-gradient norms. Let

$$G_{\mathrm{SSL}} = \sum_k w_k g_k$$

denote the contribution of the method's original loss terms, measured at its epoch-25 checkpoint. We set

$$\beta_h = \frac{T}{1-T} \frac{G_{\mathrm{SSL}}}{g_R},$$

using a fixed reference $g_R = 0.30$ for the gradient norm of SACReg. We use a fixed reference because this gradient can be unusually large at a collapsed backbone, which would otherwise yield an artificially small weight. The target share $T = 0.06$ was chosen from an earlier dose study on LeJEPA. Thus, T and g_R are fixed across methods, and β_h is determined from the scale of each method's existing objective rather than tuned separately.

This gives $\beta_h = 0.098$ (SimCLR), 0.0205 (BYOL), 1.548 (VICReg), 0.258 (DINO), 0.0146 (LeJEPA), and 0.0123 (VISReg; measured from a one-epoch pilot). For DINO, this weight reduced both linear and kNN accuracy, so we tested $\{0.115, 0.02, 0.01\}$ and use $\beta_h = 0.01$, selected by the online probe.

Our standalone objective. For our standalone objective, we set the three loss weights jointly. After a one-epoch pilot, we measure the backbone gradient norm g_k of each term and choose

$$w_k = 57\,\frac{s_k}{g_k},$$

with target shares $(s_{\mathrm{inv}}, s_{\mathrm{proj}}, s_{\mathrm{back}}) = (0.44, 0.54, 0.02)$. These shares and the total magnitude 57 come from an earlier ViT-S run, and we use the same calibration rule for ViT-S and ViT-B without further tuning.

C.4 Evaluation

ImageNet-1k linear probe and kNN. We follow the linear-evaluation recipe of the Lightly benchmark (the MAE linear-probe recipe): a BatchNorm layer without affine parameters followed by a linear classifier on the frozen CLS token, trained for 90 epochs with random resized crops and flips, SGD with momentum 0.9 (equivalent to LARS at zero weight decay), learning rate $0.1 \times \mathrm{batch}/256$ with a cosine schedule and 10 warm-up epochs; we report the best validation top-1 over the 90 epochs. The kNN classifier uses $k = 200$ neighbors with cosine similarity weighted at temperature 0.07, on the CLS features of the training set.

Transfer. We use the VISReg transfer protocol: the CLS tokens of the last four blocks are concatenated, a linear classifier is trained for 10 epochs over a grid of 13 learning rates, and the best test accuracy is reported for each dataset; the table reports the average over the eight datasets. Our implementation reproduces the published DINO ViT-B/16 row within 0.4 points on every dataset.

ImageNet-100. The linear probe is a linear classifier on the raw CLS features, trained with AdamW (learning rate 10^{-3}, weight decay 10^{-7}) until validation accuracy stops improving; the kNN classifier uses $k = 200$ and temperature 0.1. Both are fit on a fixed subset of 500 training images per class and evaluated on the 5,000 validation images.

Video. We follow the frozen-encoder protocols of V-JEPA as used by LeVJEPA. ImageNet-1k and Something-Something-v2 use an attentive probe: one cross-attention block with a learnable query over all output tokens followed by a linear classifier, trained for 20 epochs with AdamW (learning rate 10^{-3}, cosine schedule). For ImageNet-1k each image is repeated over the 16 input frames; for Something-Something-v2 a video is sampled as 2 clips of 16 frames with 3 spatial crops. Kinetics-400 uses a linear classifier. LeVJEPA does not release the settings of this linear classifier. So we report the results with L-BFGS (multinomial logistic regression on standardized features, the best of three ridge strengths $\{10^{-5}, 10^{-4}, 10^{-3}\}$) using one center crop per clip and the full training and validation sets.

Representation statistics. All statistics are computed on frozen features. RankMe (Garrido et al., 2023) is the exponential of the entropy of the normalized singular values reported as a fraction of the feature dimension. Positive and negative-pair cosines are the mean cosine similarity between two views of the same image and between views of two different images.

Augmentation thickness. For every model we store $V = 8$ augmented views of 10,000 training images (100 per class), drawn from the model's own training augmentations. For the per-class analysis of Fig. 8, we split the 10,000 images once at random into two equal halves. On the fitting

half, we estimate $\mathbf{A}_h$ as the average within-image covariance of the views and $\mathbf{B}_h$ as the covariance of the view means minus $\mathbf{A}_h/V$, which corrects for estimating each image center from finitely many views. We use $\mathbf{B}_h$ to define the whitening map and compute $\mathbf{\Theta}_h = \mathbf{B}_h^{\dagger/2}\mathbf{A}_h\mathbf{B}_h^{\dagger/2}$.

For each class c, we define the target as the centered indicator scaled to unit variance,

$$y_c = \frac{\mathbf{1}[y = c] - p_c}{\sqrt{p_c(1 - p_c)}},$$

where p_c is the class frequency in the fitting split. Using the fitting half, we fit the least-squares coefficient $\mathbf{a}_c$ for predicting y_c from the whitened image centers. We then define $d_h(c)^2$ as the mean squared residual on the held-out half. Since this is a held-out error, it can exceed 1, so the class-center separability $1 - d_h(c)^2$ can be negative for some classes. The view-sensitivity term $\mathbf{a}_c^\top \mathbf{\Theta}_h(\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1}\mathbf{a}_c$ uses the same $\mathbf{a}_c$ and the $\mathbf{\Theta}_h$ estimated on the fitting half.

C.5 Baselines

OK-AI releases LeJEPA, DINO and iBOT trained under one modernized code base at ViT-S and ViT-B for 100 and 300 epochs; we evaluate these checkpoints under the protocols above. Their model card lists 1.45M training images, which is more than the standard ImageNet-1k training split; we did not correct for this. We also trained VISReg ViT-B/16 for 100 epochs with the authors' code and their released view configuration (four global and six local crops), and evaluated the public MoCo v3, DINO, iBOT and VISReg checkpoints (300–400 epochs) the same way. Transfer numbers marked ‡ are taken from the VISReg paper. Video baselines are taken from the LeVJEPA paper: the 240-epoch numbers from its Fig. 2 and the remaining rows from its equal-compute table.

C.6 Detailed results

Table 4: Per-dataset linear-probe transfer accuracy (%) under the VISReg protocol; the last column is the average reported in Table 1a and 1b. ‡Numbers reported by the VISReg paper; all other rows are evaluated by us under the matched protocol.

Method	Backbone	Ep.	DTD	Aircraft	Cars	CIFAR10	CIFAR100	Flowers	Food	Pets	Avg.
LeJEPA	ViT-S/16	100	69.4	43.8	43.4	89.3	71.1	84.0	73.6	75.3	68.7
DINO	ViT-S/16	100	69.3	55.2	57.0	93.8	78.6	89.3	76.7	85.9	75.7
iBOT	ViT-S/16	100	69.3	55.3	56.1	93.3	77.4	90.3	77.9	87.3	75.9
λ-JEPA	ViT-S/16	100	71.1	55.5	62.5	95.3	81.6	90.4	75.3	89.1	77.6
LeJEPA	ViT-S/16	300	70.2	47.6	50.3	92.1	74.4	85.8	75.7	79.1	71.9
DINO	ViT-S/16	300	71.5	59.8	66.2	95.0	81.1	92.0	79.6	89.6	79.4
iBOT	ViT-S/16	300	71.6	59.7	63.3	96.0	82.4	91.2	80.9	90.6	79.5
λ-JEPA	ViT-S/16	400	71.8	57.9	67.9	95.5	81.8	90.3	78.1	90.7	79.3
LeJEPA	ViT-B/16	100	72.3	51.5	54.4	93.0	76.4	86.9	78.9	80.1	74.2
DINO	ViT-B/16	100	70.0	58.1	61.9	95.8	82.0	89.9	79.7	88.7	78.3
iBOT	ViT-B/16	100	71.9	61.1	66.8	97.0	84.2	91.5	82.8	90.1	80.7
VISReg	ViT-B/16	100	73.3	53.1	58.0	94.3	79.6	87.9	79.1	85.5	76.4
λ-JEPA	ViT-B/16	100	71.5	59.8	71.1	96.7	84.9	92.2	79.4	91.7	80.9
LeJEPA	ViT-B/16	300	73.7	52.0	55.0	93.5	77.6	87.7	80.5	81.8	75.2
DINO	ViT-B/16	300	71.8	59.1	63.8	96.0	83.4	90.0	80.5	89.9	79.3
iBOT	ViT-B/16	300	74.5	62.0	68.8	97.4	86.2	93.0	84.2	92.2	82.3
MoCo v3‡	ViT-B/16	300	73.7	57.9	67.5	96.9	85.2	91.5	81.8	89.8	80.5
DINO‡	ViT-B/16	400	74.3	63.6	73.9	96.5	85.0	94.6	83.1	93.6	83.1
iBOT‡	ViT-B/16	400	74.1	63.5	73.8	97.1	85.9	93.7	84.2	93.6	83.2
VISReg‡	ViT-B/16	400	75.7	57.1	64.8	94.6	78.8	90.4	82.9	88.3	79.1
λ-JEPA	ViT-B/16	400	73.2	61.1	74.0	96.9	85.4	92.0	81.0	92.2	82.0

D A closer look at LeJEPA

Our results suggest that anti-collapse should act on the representation retained for downstream tasks. A natural alternative explanation is that the particular regularizer is unimportant: perhaps simply

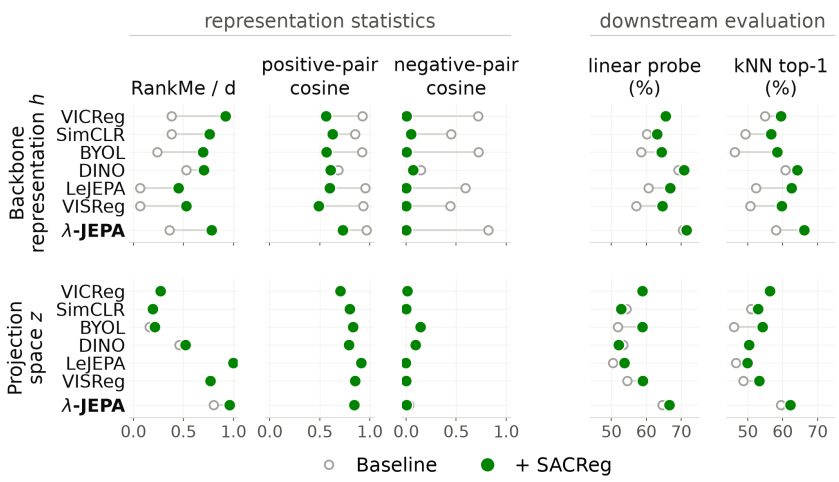

Figure 7: Effect of applying SACReg to the backbones across SSL objectives on ImageNet-100. We report representation statistics at the backbone $\mathbf{h}$ and projection space $\mathbf{z}$, together with downstream linear probe and kNN accuracy. Open circles denote the baseline method and green circles the same method with SACReg added at the backbone. Each point is the mean over three seeds.

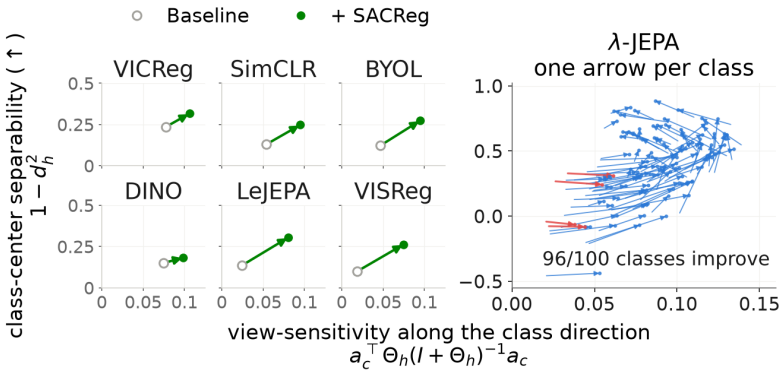

Figure 8: Effect of SACReg on view sensitivity and class-center separability. Left: mean class-level changes for six SSL methods on ImageNet-100, with arrows from the baseline representation to the same method with SACReg added at the backbone. Right: class-wise changes for λ-JEPA; 96 of 100 classes improve in class-center separability.

applying an existing anti-collapse objective at the backbone is sufficient. We test this with LeJEPA, one of the explicitly regularized methods in our study.

LeJEPA encourages the projected representation to match an isotropic Gaussian using SIGReg, a sketched Gaussianity test over random one-dimensional projections. However, its retained backbone representation is far from this target (Fig. 1). Starting from vanilla LeJEPA, with the original SIGReg and invariance losses after the projector unchanged, we therefore add a second SIGReg term directly to the 512-dimensional encoder output, using the weighting heuristic of Appendix C.3, and train under the ImageNet-100 setup of Section 4.4.

Table 5 shows that the added SIGReg substantially improves several Gaussianity statistics at the backbone. Negative-pair cosine drops from 0.58 to 0.01, the diagonal Gaussian KL from 0.96 to 0.07, and the Epps–Pulley statistic also decreases. However, the covariance spectrum remains concentrated: RankMe$/d$ changes only from 0.07 to 0.09, while the full-covariance KL remains high ($3.89 \rightarrow 3.03$). At the same time, positive-pair cosine decreases and downstream performance worsens by 1.4 linear-probe points and 4.3 kNN points. Thus, the gains from SACReg are not explained simply by moving LeJEPA's existing anti-collapse constraint to the retained representation. In this

comparison, directly controlling the covariance spectrum is important: SIGReg substantially improves its Gaussianity diagnostics but leaves the backbone spectrum concentrated, whereas $\mathcal{R}_{\mathrm{SAC}}$ acts explicitly on that spectrum through its log-determinant term.

Readout at encoder output ($\mathbf{h}$)	LeJEPA	+ SIGReg at $\mathbf{h}$
Positive-pair cosine	0.96	0.73
Negative-pair cosine	0.58	0.01
KL to $\mathcal{N}(0,1)$ (diagonal)	0.96	0.07
KL to $\mathcal{N}(0,I)$ (full covariance)	3.89	3.03
Epps–Pulley	652	504
RankMe$/d$	0.07	0.09
Linear probe (%)	60.4	58.9
kNN, $k = 200$ (%)	52.4	48.1

Table 5: Adding SIGReg directly to the LeJEPA encoder output on ImageNet-100. The original projector losses are unchanged. All representation statistics are measured at the 512-dimensional encoder output.

E Augmentation thickness: definition and interpretation

This section gives the formal motivation for the augmentation-thickness quantities used in Section 4.4. In particular, we show that augmentation thickness has two opposing effects: too little thickness can suppress meaningful augmentation-dependent variation, whereas too much allows within-image variation to dominate the separation between images.

E.1 Definition

Let Q denote a source image and let $\mathbf{h}$ be its representation under a randomly sampled augmented view. Define the image center

$$\mathbf{m}_h(Q) = \mathbb{E}[\mathbf{h} \mid Q] \tag{39}$$

and the residual view-dependent component

$$\boldsymbol{\xi}_h = \mathbf{h} - \mathbf{m}_h(Q). \tag{40}$$

By the law of total covariance,

$$\mathbf{C}_h = \mathrm{Cov}(\mathbf{h}) = \mathbf{B}_h + \mathbf{A}_h, \qquad \mathbf{B}_h = \mathrm{Cov}(\mathbf{m}_h(Q)), \qquad \mathbf{A}_h = \mathbb{E}[\mathrm{Cov}(\mathbf{h} \mid Q)]. \tag{41}$$

Thus, $\mathbf{B}_h$ measures variation between image centers with the rank $r_h = \mathrm{rank}(\mathbf{B}_h)$, while $\mathbf{A}_h$ measures variation across augmented views of the same image.

The same amount of within-image variation can have very different consequences depending on the scale of the between-image variation. We therefore normalize $\mathbf{A}_h$ by $\mathbf{B}_h$ and define

$$\boldsymbol{\Theta}_h = \mathbf{B}_h^{\dagger/2} \mathbf{A}_h \mathbf{B}_h^{\dagger/2}. \tag{42}$$

We call $\boldsymbol{\Theta}_h$ the *augmentation thickness* of the representation. Along any direction $\mathbf{a}$ with nonzero between-image variance,

$$\theta_h(\mathbf{a}) = \frac{\mathbf{a}^\top \mathbf{A}_h \mathbf{a}}{\mathbf{a}^\top \mathbf{B}_h \mathbf{a}} \tag{43}$$

is the ratio between within-image and between-image variation in that direction.

We next formalize why augmentation thickness should not simply be minimized. Let $y = y(Q)$ be a centered image-level target with $\|y\|_{L_2} \le 1$, and define

$$E_{\mathcal{A}}(y) = \min_f \mathbb{E}\Big[\big(y(Q) - f(\mathbf{v})\big)^2\Big], \tag{44}$$

where $\mathbf{v}$ is a single augmented view of Q. This is the irreducible error of predicting y from one augmented view.

Let $\mathcal{M}_h$ denote the set of affine functions of the image centers and define

$$d_h(y) = \min_{g \in \mathcal{M}_h} \|y - g\|_{L_2}. \tag{45}$$

Thus, $d_h(y)^2$ is the error of the best affine prediction of y from the image centers.

Theorem E.1 (Two sides of augmentation thickness)**.** *Restrict to the support of $\mathbf{B}_h$ and work in centered, whitened coordinates, so that*

$$\mathbb{E}[\mathbf{m}_h(Q)] = \mathbf{0}, \qquad \mathrm{Cov}(\mathbf{m}_h(Q)) = \mathbf{I}_{r_h}, \qquad \mathrm{Cov}(\boldsymbol{\xi}_h) = \boldsymbol{\Theta}_h. \tag{46}$$

Then

$$d_h(y) \geq \left(\sqrt{E_{\mathcal{A}}(y)} - \sqrt{\|\boldsymbol{\Theta}_h\|_{\mathrm{op}}} \right)_+. \tag{47}$$

Moreover, let

$$g(Q) = \mathbf{a}^\top \mathbf{m}_h(Q) \tag{48}$$

be the best linear prediction of y from the image centers. Then the best linear prediction error from a single-view representation $\mathbf{h}$ is

$$\inf_{\mathbf{u}} \mathbb{E}\big[(y - \mathbf{u}^\top \mathbf{h})^2\big] = d_h(y)^2 + \mathbf{a}^\top \boldsymbol{\Theta}_h (\mathbf{I}_{r_h} + \boldsymbol{\Theta}_h)^{-1} \mathbf{a}. \tag{49}$$

For fixed image-center organization and fixed task direction $\mathbf{a}$, the second term is monotone increasing in $\boldsymbol{\Theta}_h$ in the positive-semidefinite order.

The two statements capture complementary failure modes. Equation 47 says that if a target cannot be recovered perfectly from an augmented view, then a representation with very little augmentation thickness cannot nevertheless organize its image centers arbitrarily well for that target. Some augmentation-dependent variation may therefore need to remain in the representation. In the opposite direction, Eq. 49 shows that, once the image-center organization is fixed, additional within-image variation makes prediction from a single view harder. Augmentation thickness therefore describes a trade-off rather than a quantity that should be uniformly minimized.

The quantity

$$\mathbf{a}^\top \boldsymbol{\Theta}_h (\mathbf{I}_{r_h} + \boldsymbol{\Theta}_h)^{-1} \mathbf{a} \tag{50}$$

is the *view-sensitivity* term reported in Fig. 8. We use this quantity in Fig. 8 rather than the raw directional thickness $\theta_h(\mathbf{a})$ because it captures the contribution of augmentation variation after optimizing the linear prediction. It isolates the part of the optimal single-view linear prediction error attributable to variation across augmentations, separately from the image-center organization term $d_h(y)^2$.

E.2 Proof of Theorem E.1

Throughout the proof we work on the support of $\mathbf{B}_h$ in centered, whitened coordinates and reuse $\mathbf{h}$, $\mathbf{m}_h$, and $\boldsymbol{\xi}_h$ for the transformed variables. Hence

$$\mathbb{E}[\mathbf{m}_h(Q)] = \mathbf{0}, \qquad \mathrm{Cov}(\mathbf{m}_h(Q)) = \mathbf{I}_{r_h}, \qquad \mathbb{E}[\boldsymbol{\xi}_h \mid Q] = \mathbf{0}, \qquad \mathrm{Cov}(\boldsymbol{\xi}_h) = \boldsymbol{\Theta}_h. \tag{51}$$

Too little thickness. Consider any linear function of the image centers,

$$g(Q) = \mathbf{a}^\top \mathbf{m}_h(Q). \tag{52}$$

Since $\mathbf{a}^\top \mathbf{h}$ is computable from a single augmented view, it is one admissible predictor of g. Therefore,

$$E_{\mathcal{A}}(g) \leq \mathbb{E}\Big[\big(\mathbf{a}^\top \mathbf{m}_h(Q) - \mathbf{a}^\top \mathbf{h}\big)^2\Big] \tag{53}$$

$$= \mathbb{E}\big[(\mathbf{a}^\top \boldsymbol{\xi}_h)^2\big] \tag{54}$$

$$= \mathbf{a}^\top \boldsymbol{\Theta}_h \mathbf{a}. \tag{55}$$

Now let g be the best linear prediction of y from the image centers. Since the image-center covariance is whitened, we may write

$$g(Q) = \mathbf{a}_y^\top \mathbf{m}_h(Q). \tag{56}$$

Orthogonal projection in L_2 gives

$$\|\mathbf{a}_y\| = \|g\|_{L_2} \le \|y\|_{L_2} \le 1. \tag{57}$$

Using the triangle inequality,

$$\sqrt{E_{\mathcal{A}}(y)} \le \|y - g\|_{L_2} + \sqrt{E_{\mathcal{A}}(g)} \tag{58}$$

$$\le d_h(y) + \sqrt{\mathbf{a}_y^\top \mathbf{\Theta}_h \mathbf{a}_y} \tag{59}$$

$$\le d_h(y) + \sqrt{\|\mathbf{\Theta}_h\|_{\mathrm{op}}}, \tag{60}$$

where the last inequality uses $\|\mathbf{a}_y\| \le 1$. Rearranging yields

$$d_h(y) \ge \left(\sqrt{E_{\mathcal{A}}(y)} - \sqrt{\|\mathbf{\Theta}_h\|_{\mathrm{op}}} \right)_+, \tag{61}$$

which proves Eq. 47.

Too much thickness. Let

$$g(Q) = \mathbf{a}^\top \mathbf{m}_h(Q) \tag{62}$$

be the best linear prediction of y from the image centers, and write

$$y = g + r. \tag{63}$$

By the orthogonality property of linear regression,

$$\mathbb{E}[r\, \mathbf{m}_h(Q)] = \mathbf{0}, \qquad \mathbb{E}[r^2] = d_h(y)^2. \tag{64}$$

Moreover, since $\mathbb{E}[\boldsymbol{\xi}_h \mid Q] = \mathbf{0}$ and r is a function of Q,

$$\mathbb{E}[r\, \boldsymbol{\xi}_h] = \mathbb{E}[r\, \mathbb{E}[\boldsymbol{\xi}_h \mid Q]] = \mathbf{0}. \tag{65}$$

Hence r is uncorrelated with

$$\mathbf{h} = \mathbf{m}_h(Q) + \boldsymbol{\xi}_h \tag{66}$$

and contributes exactly $d_h(y)^2$ to the optimal affine prediction error from $\mathbf{h}$.

For the component $g = \mathbf{a}^\top \mathbf{m}_h(Q)$,

$$\mathrm{Cov}(\mathbf{h}) = \mathbf{I}_{r_h} + \mathbf{\Theta}_h, \qquad \mathrm{Cov}(\mathbf{h}, \mathbf{m}_h(Q)) = \mathbf{I}_{r_h}. \tag{67}$$

The best linear prediction error of g from $\mathbf{h}$ is therefore

$$\mathbb{E}[g^2] - \mathrm{Cov}(g, \mathbf{h})\, \mathrm{Cov}(\mathbf{h})^{-1}\, \mathrm{Cov}(\mathbf{h}, g) \tag{68}$$

$$= \mathbf{a}^\top \left[\mathbf{I}_{r_h} - (\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1} \right] \mathbf{a} \tag{69}$$

$$= \mathbf{a}^\top \mathbf{\Theta}_h (\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1} \mathbf{a}. \tag{70}$$

Adding the residual contribution gives

$$\inf_{\mathbf{u}} \mathbb{E}\left[(y - \mathbf{u}^\top \mathbf{h})^2\right] = d_h(y)^2 + \mathbf{a}^\top \mathbf{\Theta}_h (\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1} \mathbf{a}, \tag{71}$$

which proves Eq. 49.

Finally,

$$\mathbf{\Theta}_h (\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1} = \mathbf{I}_{r_h} - (\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1}. \tag{72}$$

If $\mathbf{\Theta}'_h \succeq \mathbf{\Theta}_h$, then

$$(\mathbf{I}_{r_h} + \mathbf{\Theta}'_h)^{-1} \preceq (\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1}, \tag{73}$$

and therefore

$$\mathbf{\Theta}'_h (\mathbf{I}_{r_h} + \mathbf{\Theta}'_h)^{-1} \succeq \mathbf{\Theta}_h (\mathbf{I}_{r_h} + \mathbf{\Theta}_h)^{-1}. \tag{74}$$

Thus, for fixed $\mathbf{a}$, increasing augmentation thickness can only increase the view-sensitivity term. This completes the proof.