Understanding Feature Learning Dynamics in Isotropic Regularizers via BHEP Statistics

Jeongyou Lee HOO9565J@HANYANG.AC.KR
Department of Computer Science, Hanyang University, Seoul, Korea

Sungyoon Lee SUNGYOONLEE@HANYANG.AC.KR
Department of Computer Science, Hanyang University, Seoul, Korea

Abstract

Self-supervised training of LeJEPA is governed by the SIGReg regularizer to ensure an isotropic Gaussian embedding distribution and prevent dimensional collapse. However, directly analyzing its learning dynamics remains mathematically challenging, due to its random projection and lack of closed-form expressions. In this work, we bridge this gap by reformulating the SIGReg objective into a tractable BHEP-type statistic that shares the identical target distribution. Leveraging this theoretical proxy, we derive exact closed-form ODEs governing the macroscopic eigenvalue learning dynamics. Based on this formulation, we characterize the feature learning dynamics through two distinct phase transitions: a rapid initial *Explosion Phase* followed by a self-stabilizing *Shrinking Phase*. Crucially, we decouple the global driving force into a collapsing *engine* and a stiffening *brake*. Our exact analytic analysis reveals an extreme asymmetric decay between these forces, mathematically proving how the early saturation of top features profoundly depletes the driving force for subsequent features. This mechanism explains the severe, non-linear stepwise delay observed during training, while ultimately guaranteeing stable convergence toward isotropy.

1. Introduction

Self-supervised learning (SSL) has revolutionized representation learning, but a fundamental challenge is avoiding dimensional collapse—where representations degenerate into a low-dimensional manifold. To mitigate this, prior works employ off-diagonal correlation suppression (Barlow Twins [16]), variance/covariance regularization (VICReg [3]), negative pairs (SimCLR [7]), or stop-gradient with EMA (JEPA [1], DINO [6]). Recently, LeJEPA [2] introduced the *SIGReg* regularizer, which encourages the embedding distribution to match an isotropic Gaussian. However, while such approaches constrain the overall distribution, it remains unclear how individual features are actually learned during training.

Although prior analyses have characterized per-feature dynamics for Barlow Twins [11, 13], these results do not transfer to LeJEPA, whose dynamics align more closely with VICReg. Moreover, directly analyzing SIGReg is challenging because of its random projections. Recent work [17] provided a closed-form MMD-Gauss approximation, but its reliance on Kummer's function (can be approximated as an inverse-root form in practical setting of dimension $d > 20$) yields a highly non-linear structure that remains intractable for explicit ODE derivation.

In this work, we bridge this gap by applying a Gamma-integral transform, which reveals that the MMD-Gauss objective can be reformulated as a Gamma-weighted extension of BHEP-type statistics [4]. While the resulting integral term still poses analytical hurdles, this structural equivalence

© J. Lee & S. Lee.

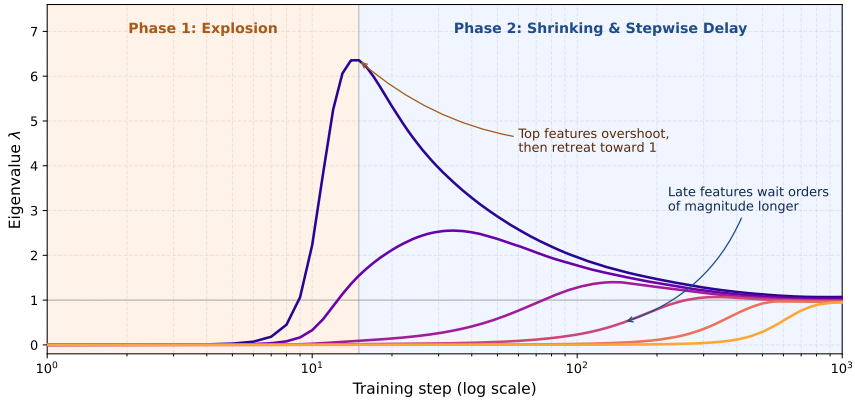

Figure 1: **Learning dynamics of the tractable BHEP proxy.** Eigenvalue trajectories exhibit two distinct phases. In *Phase 1 (Explosion)*, top features rapidly grow and overshoot the target. In *Phase 2 (Shrinking & Stepwise Delay)*, they slowly retreat, triggering lower-ranked features to sequentially emerge after exponentially long delays.

identifies the BHEP statistic as the fundamental energy landscape of Gaussian regularization. Consequently, we adopt BHEP as a tractable proxy to derive exact ODEs and map the macroscopic learning trajectories. As illustrated in Figure 1, our analysis reveals a striking coupled dynamic: a small subset of top eigenvalues rapidly overshoots the ideal target 1, then slowly decays as lower-ranked eigenvalues emerge. This overshoot-and-join phenomenon has also been observed empirically in SimCLR/VICReg [13] and theoretically in Barlow Twins [11].

Our contributions: **(i)** we derive a complete closed-form ODE $\dot{\lambda}_k \propto \lambda_k [\alpha - \beta\lambda_k]$ for each eigenvalue, exposing an interpretable engine–brake structure; **(ii)** we cleanly separate the trajectory into an *Explosion* phase, where the unsuppressed engine α drives top features to overshoot, and a *Shrinking* phase, where the asymmetric decay of α relative to a quickly-plateauing brake β pulls $T = \alpha/\beta$ toward $\lambda^* = 1$; **(iii)** we show that this engine–brake dichotomy persists under the exact MMD-Gauss objective via a Gamma-integral representation, and verify that the same dynamics govern SIGReg in practice.

2. SIGReg to BHEP Proxy

LeJEPA's SIGReg regularizer evaluates the Gaussianity of the embedding space using the empirical characteristic function based on the Epps-Pulley (EP) test [9].

Definition 1 (SIGReg / Epps-Pulley Loss) *The deviation between the empirical characteristic function $\hat{\phi}_X(t)$ and Gaussian $\phi(t)$, weighted by $w(t)$:*

$$EP = n \int_{-\infty}^{\infty} |\hat{\phi}_X(t) - \phi(t)|^2 w(t)\, dt. \tag{1}$$

In practice, we divide by n and apply a coefficient $\frac{1}{\sqrt{2\pi}}$ to align magnitudes (Appendix C).

This only checks scalar quantities, but SIGReg uses random projections to map high-dimensional embeddings to scalar distributions; with sufficient projections, it enforces isotropic Gaussianity [8].

Bridging SIGReg to BHEP via the Gamma integration trick. Tracking exact dynamics of SIGReg is intractable due to random projections. Recently, Zimmermann et al. [17] showed that the expected sliced EP test equals the Maximum Mean Discrepancy (MMD) with a Kummer-related kernel, which can be approximated into inverse root form in practical setting (dimension $d > 20$). However, its inverse polynomial structure remains non-linear and intractable for closed-form ODEs. To bridge this, we apply the Gamma identity $\frac{1}{\sqrt{Z}} = \frac{1}{\sqrt{\pi}} \int_0^\infty t^{-1/2} e^{-Zt} dt$ (Appendix J.1), which represents the inverse polynomial as a continuous mixture of exponentials. This directly motivates the Baringhaus-Henze-Epps-Pulley (BHEP) statistic [4] as our proxy: it preserves the identical target distribution and exponential dynamics while admitting a strictly closed-form ODE.

Definition 2 (BHEP Loss)

$$\begin{aligned} BHEP_{n,\beta} = \frac{1}{n} \sum_{j,k=1}^{n} \exp\left(-\frac{\beta^2 \|Y_{n,j} - Y_{n,k}\|^2}{2}\right) \\ - \frac{2}{(1+\beta^2)^{d/2}} \sum_{j=1}^{n} \exp\left(-\frac{\beta^2 \|Y_{n,j}\|^2}{2(1+\beta^2)}\right) + \frac{n}{(1+2\beta^2)^{d/2}}. \end{aligned} \tag{2}$$

For calculation, we divide the term by n to match the average.

We adopt BHEP with bandwidth $\beta^2 = \frac{1}{d}$ as our *tractable surrogate proxy*. The closed-form ODEs derived from BHEP provide critical insights into the phase transitions that SIGReg inherently experiences.

Assumptions. We assume sufficient dimension d and sample size n ($n > d$). Following Simon et al. [13], we align the weight matrix $W = V^{(\leq d)} S Q^T$ with the top-d components of C_X, yielding $\lambda_k = \sigma_k^2 s_k^2$ (Appendix I). Top-eigenvector projections of the input are assumed multivariate Gaussian (Appendix E), and the invariant term $\|z - z'\|_2^2$ is treated as already saturated and negligible [5].

3. Learning Dynamics of BHEP

3.1. Closed-form Derivation

Using continuous-time gradient descent, the BHEP objective yields:

Theorem 3 (Eigenvalue ODE) *The time derivative of the k-th embedding eigenvalue λ_k is governed by the interplay of Local and Global forces:*

$$\dot{\lambda}_k = \underbrace{2\sigma_k^2}_{\textit{Local force}} \cdot \lambda_k \cdot \underbrace{u_k \cdot [\alpha - \beta \lambda_k]}_{\textit{Global force}}, \tag{3}$$

with denominator u_k, and α, β macroscopic loss-dependent coefficients (Appendix A, Appendix B).

The *Local force* depends only on the input variances σ_k^2, so features with larger input variance learn faster, same as Simon et al. [13]. The *Global force* evolves dynamically: unlearned eigenvalues ($\lambda_k \approx 0$) are powered solely by the **driving engine** α, while active eigenvalues are bounded by the **regularizing brake** β, with target $T = \alpha/\beta$:

$$T = \frac{c_1 P_1 - \gamma c_2 P_2}{c_1 c_2 (\gamma P_2 - P_1)} \approx \frac{2P_1 - \gamma P_2}{2(\gamma P_2 - P_1)} d, \tag{4}$$

where P_1, P_2 are the global Push/Pull potentials, with $c_1 = \frac{2}{d}, c_2 = \frac{1}{d+1}, \gamma = \frac{2}{\sqrt{e}}$.

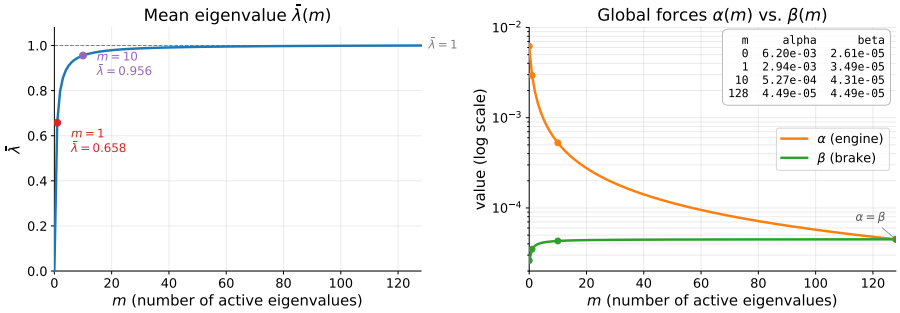

Figure 2: **Dynamics vs. number of active eigenvalues m for 128 dimension theoretical setting.** Via Eq. (5), $\bar{\lambda}$ jumps toward 1, α abruptly collapses, and β plateaus after a few features. (Used precise calculation from Appendix G.1)

3.2. Two Phases of Learning

Because α and β are dynamically determined by the macroscopic state of all eigenvalues, features do not learn in isolation: the rapid growth of early dominant features alters the loss landscape by activating the systemic brake.

Phase 1 — Initial explosion. At initialization ($\lambda_k \approx 0$), $\alpha \approx 0.787/d$ dominates the much weaker $\beta \approx 0.426/d^2$, driving top features rapidly. Isolating λ_1 via a first-order Taylor expansion (Appendix F) mirrors the stepwise ODE of Simon et al. [13], with both the peak and critical time scaling as $\mathcal{O}(d)$. Yet a single activated feature instantly halves α ($\to 0.374/d$) while β stiffens to $0.568/d^2$: the engine is critically crippled even at small $\bar{\lambda}$.

Phase 2 — Asymmetric decay and stepwise delay. Suppose m eigenvalues have saturated into T, and the others remain small, $\lambda_i \approx 0$. Steady-state analysis gives (Proof in Appendix G):

$$\bar{\lambda} \approx \frac{e^{1/(m+2)} - 1}{2 - e^{1/(m+2)}}\, m, \tag{5}$$

which steadily rises from ≈ 0.654 (at $m = 1$) toward 1. A profound asymmetry emerges: $\beta \propto \frac{1}{d^2}(\gamma P_2 - P_1)$ relies on the slower-decaying P_2 and rapidly plateaus—at $m = 10$, β already reaches over 95% of its asymptotic maximum. In stark contrast, α collapses by term cancellation ($> 90\%$ loss at $m = 10$), ultimately matching β at $\mathcal{O}(1/d^2)$. With β acting as a near-constant wall, $T = \alpha/\beta$ becomes proportional to the collapsing engine, forcing subsequent features to learn under exponentially depleted driving forces. As T drifts downward, the early-exploded eigenvalues retreat, and the process concludes at $\alpha = \beta$ with all active eigenvalues at 1 (refer to Figure 2 for the corresponding curves).

Universality and asymptotic target. The same engine–brake structure persists under the MMD-Gauss via the Gamma-integral representation (Appendix J). The BHEP fixed-point equation $\dot{\lambda}_k = 0$ admits the exact equilibrium $\lambda^* = 1$ (Appendix G); the MMD-Gauss expansion yields $\lambda^* = 1 - 1/d$ as a finite-dimensional residual (Appendix J.5), whose $\mathcal{O}(1/d)$ deviation vanishes at large d — and indeed the unapproximated SIGReg objective recovers exact convergence to $\lambda = 1$.

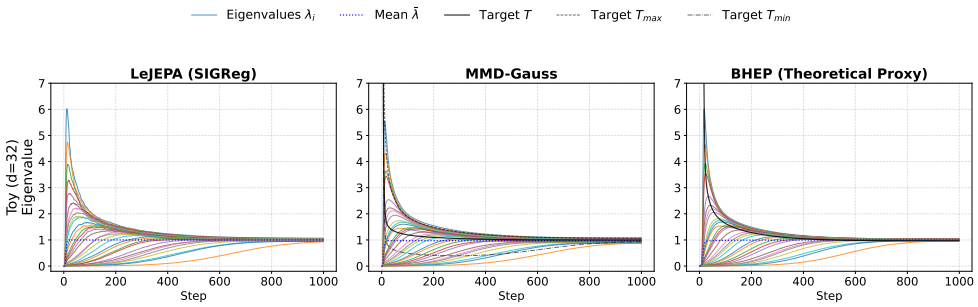

Figure 3: **Eigenvalue learning dynamics across regularizers on a 32-dimensional toy setting.** Panels: SIGReg, MMD-Gauss (Gauss-Laguerre, 10 knots), and BHEP. (For MMD-Gauss, the per-eigenvalue target from Appendix J.3 gives three lines by $\bar{\lambda}$, $\lambda_{\max}$, $\lambda_{\min}$)

4. Experimental Results

Phase transitions and stepwise delay. On a 32-dim toy setting (Detailed setting is provided in Appendix N.1), Figure 3 reproduces the full lifecycle predicted by our theory across SIGReg, MMD-Gauss, and BHEP. Several embeddings rapidly peak in the *Explosion Phase*; the dynamic target $T = \alpha/\beta$ (black solid line) then decreases as $\bar{\lambda}$ rises—triggering the *Shrinking Phase*, where exploded eigenvalues retreat toward $T \to 1$. Empirically, while initial input variances (σ_k^2) differ by only $\sim 4\times$, convergence times span orders of magnitude: the engine α depletes by a factor far exceeding the variance gap, leaving inactive dimensions with a profoundly weakened force. Comparing the theoretical predictions for the activated eigenvalue m derived in Section 3.2 with empirical observations, we find that while simultaneous multi-feature activation causes the transient trajectories to deviate, both ultimately gravitate toward the same convergence target (Appendix H).

Practical validation on CIFAR-10. The same qualitative phenomena hold on a ResNet-18 ($d = 128$) trained on CIFAR-10 with all three regularizers (Appendix L). Under infinitesimal initialization, top eigenvalues rise sharply and then retreat as lower-ranked features emerge, reproducing the explosion-and-shrink lifecycle predicted by our analysis. Crucially, the predicted engine–brake asymmetry holds outside the controlled toy setting: across all three regularizers, α drops by orders of magnitude while β stays nearly constant, mirroring the steady-state analysis in Section 3.2.

5. Conclusions, Limitations, and Future Work

We presented a closed-form analysis of LeJEPA's feature learning dynamics via the BHEP proxy, decomposing the trajectory into explosion and shrinking phases through an interpretable engine (α)–brake (β) structure, with all eigenvalues theoretically converging to the isotropic target $\bar{\lambda} = 1$.

Limitations. Our analysis assumes sufficient n, d, Gaussian inputs, and weight alignment; the BHEP–SIGReg gap blocks fully closed-form derivations, and $\mathcal{O}(n^2 d)$ pairwise cost limits large-scale tracking. Full discussion in Appendix M.

Future work. We hope to extend the analysis to VICReg [3] and SimCLR [7], which share similar dynamics [13] and embedding distributions [5, 15].

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15619–15629, 2023.

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

[3] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

[4] Ludwig Baringhaus and Norbert Henze. A consistent test for multivariate normality based on the empirical characteristic function. *Metrika*, 35(1):339–348, 1988.

[5] Roy Betser, Eyal Gofer, Meir Yossef Levi, and Guy Gilboa. InfoNCE induces gaussian distribution. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=BlSH7gNQSq.

[6] 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*, pages 9650–9660, 2021.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

[8] Harald Cramér and Herman O. A. Wold. Some theorems on distribution functions. *Journal of The London Mathematical Society-second Series*, pages 290–294, 1936. URL https://api.semanticscholar.org/CorpusID:122761325.

[9] T. W. EPPS and LAWRENCE B. PULLEY. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 12 1983. ISSN 0006-3444. doi: 10.1093/biomet/70.3.723. URL https://doi.org/10.1093/biomet/70.3.723.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] Juhwan Kim and Sungyoon Lee. Stepwise feature learning in self-supervised learning, 2026. URL https://openreview.net/forum?id=P23YpnH3kq.

[12] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

[13] James B Simon, Maksis Knutins, Liu Ziyin, Daniel Geisz, Abraham J Fetterman, and Joshua Albrecht. On the stepwise nature of self-supervised learning. In *International Conference on Machine Learning*, pages 31852–31876. PMLR, 2023.

[14] Basile Terver, Randall Balestriero, Megi Dervishi, David Fan, Quentin Garrido, Tushar Nagarajan, Koustuv Sinha, Wancong Zhang, Mike Rabbat, Yann LeCun, et al. A lightweight library for energy-based joint-embedding predictive architectures. *arXiv preprint arXiv:2602.03604*, 2026.

[15] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR, 2020.

[16] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.

[17] Eric Zimmermann, Harley Wiltzer, Justin Szeto, David Alvarez-Melis, and Lester Mackey. Kerjepa: Kernel discrepancies for euclidean self-supervised learning. *arXiv preprint arXiv:2512.19605*, 2025.

Appendix A. Transformation of the BHEP objective into a product of eigenvalues

Using $\beta^2 = \frac{1}{d}$ from BHEP with average, then the overall formula becomes:

$$L = \frac{1}{n^2}\sum_{j,k=1}^{n}\exp\left(-\frac{\|z_j - z_k\|^2}{2d}\right) - \frac{2}{n(1+\frac{1}{d})^{d/2}}\sum_{j=1}^{n}\exp\left(-\frac{\|z_j\|^2}{2(d+1)}\right) + \frac{1}{(1+\frac{2}{d})^{d/2}} \tag{6}$$

As we assume the projections of the input data onto the top eigenvectors (the principal components) follow a multivariate Gaussian distribution (See Appendix E for empirical validation), we can use the formula:

$$\mathbb{E}_x\left[\exp\left(-\frac{1}{2}x^T M x\right)\right] = \det(I + \Sigma M)^{-1/2} \tag{7}$$

Applying this identity with $\Sigma = 2C_z, M = \frac{1}{d}I_d$ for the first term (since $z_i - z_j \sim \mathcal{N}(0, 2C_z)$) and $\Sigma = C_z, M = \frac{1}{d+1}I_d$ for the second term, and the overall loss formula elegantly simplifies to a function of determinants:

Assuming that d is sufficiently large,

$$\begin{aligned} L &= \det\left(I + \frac{2}{d}C_Z\right)^{-1/2} - \frac{2}{(1+\frac{1}{d})^{d/2}}\det\left(I + \frac{1}{d+1}C_Z\right)^{-1/2} + \frac{1}{(1+\frac{2}{d})^{d/2}} \\ &\approx \det\left(I + \frac{2}{d}C_Z\right)^{-1/2} - \frac{2}{\sqrt{e}}\det\left(I + \frac{1}{d+1}C_Z\right)^{-1/2} + \frac{1}{e} \end{aligned} \tag{8}$$

Using $\lambda_k = \sigma_k^2 s_k^2$, we can derive loss as:

$$\begin{aligned} L &= \prod_{i=1}^{d}(1 + c_1\lambda_i)^{-1/2} - \gamma\prod_{i=1}^{d}(1 + c_2\lambda_i)^{-1/2} + \frac{1}{e} \\ &= \underbrace{\prod_{i=1}^{d}(1 + c_1\sigma_i^2 s_i^2)^{-1/2}}_{\text{Push: } P_1} - \gamma\underbrace{\prod_{i=1}^{d}(1 + c_2\sigma_i^2 s_i^2)^{-1/2}}_{\text{Pull: } P_2} + \frac{1}{e} \end{aligned} \tag{9}$$

where $\gamma = \frac{2}{\sqrt{e}}, c_1 = \frac{2}{d}, c_2 = \frac{1}{d+1}$.

Appendix B. Deriving the closed-form learning dynamics

Using the chain rule, $\frac{dL}{ds_k} = \frac{dL}{d\lambda_k}\cdot\frac{d\lambda_k}{ds_k}$,

$$\frac{dL}{d\lambda_k} = P_1\cdot\left(-\frac{1}{2}\right)\cdot\left(\frac{c_1}{1 + c_1\lambda_k}\right) - \gamma\cdot P_2\cdot\left(-\frac{1}{2}\right)\cdot\left(\frac{c_2}{1 + c_2\lambda_k}\right) \tag{10}$$

Furthermore, since $\frac{d\lambda_k}{ds_k} = 2\sigma_k^2 s_k$ and $\dot{s}_k = -\frac{dL}{ds_k}$,

$$
\begin{aligned}
\dot{s}_k &= \sigma_k^2 s_k \cdot \left[P_1 \cdot \frac{c_1}{1 + c_1\lambda_k} - \gamma \cdot P_2 \cdot \frac{c_2}{1 + c_2\lambda_k} \right] \\
&= \frac{\sigma_k^2 s_k}{(1 + c_1\lambda_k)(1 + c_2\lambda_k)} [(c_1P_1 - \gamma c_2P_2) + c_1c_2(P_1 - \gamma P_2)\lambda_k] \\
&= \sigma_k^2 s_k \cdot u_k \cdot [(c_1P_1 - \gamma c_2P_2) - c_1c_2(\gamma P_2 - P_1)\sigma_k^2 s_k^2]
\end{aligned} \tag{11}
$$

Where $u_k = \frac{1}{(1+c_1\lambda_k)(1+c_2\lambda_k)}$. This yields a formulation quite similar to that of Simon et al. [13].

If we use $\dot{\lambda}_k = 2\sigma_k^2 s_k \cdot \dot{s}_k$ to track eigenvalue dynamics, then

$$
\begin{aligned}
\dot{\lambda}_k &= 2\sigma_k^2 s_k \cdot \sigma_k^2 s_k u_k [(c_1P_1 - \gamma c_2P_2) + c_1c_2(P_1 - \gamma P_2)\lambda_k] \\
&= \underbrace{2\sigma_k^2}_{\text{Local force}} \cdot \lambda_k \cdot \underbrace{u_k[\alpha - \beta\lambda_k]}_{\text{Global force}}
\end{aligned} \tag{12}
$$

Appendix C. SIGReg - loss coefficient

In LeJEPA [2] paper pseudocode and official repository, they share $e^{-\frac{1}{2}t^2}$ for both $w(t)$ and target CF $\phi(t)$, instead of $\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}t^2}$ for window $w(t)$. So it is recommended to multiply $\sqrt{2\pi} \approx 2.5066$ to regularizer coefficient, if you want to match magnitude of other regularizers with SIGReg.

$$
\begin{aligned}
EP &= \int_{-\infty}^{\infty} |\hat{\phi}_X(t) - \phi(t)|^2 w(t) dt \\
&= \int_{-\infty}^{\infty} (\hat{\phi}_X(t) - \phi(t)) \cdot (\overline{\hat{\phi}_X(t)} - \phi(t)) \cdot e^{-\frac{1}{2}t^2} dt \\
&= \int_{-\infty}^{\infty} (\hat{\phi}_X(t) \cdot \overline{\hat{\phi}_X(t)} - \phi(t) \cdot (\overline{\hat{\phi}_X(t)} + \hat{\phi}_X(t)) + \phi(t)^2) \cdot e^{-\frac{1}{2}t^2} dt \\
&= \int_{-\infty}^{\infty} \frac{1}{n^2} \sum_{j,k=1}^{n} e^{-\frac{1}{2}t^2 + it(z_j - z_k)} - \frac{1}{n} \sum_{j=1}^{n} \left(e^{-t^2 - itz_j} + e^{-t^2 + itz_j} \right) + e^{-\frac{3}{2}t^2} dt \\
&= \frac{\sqrt{2\pi}}{n^2} \sum_{j,k=1}^{n} e^{-\frac{1}{2}(z_j - z_k)^2} - \frac{\sqrt{2\pi}}{n} \sum_{j=1}^{n} \left(\frac{2}{\sqrt{2}} \cdot e^{-\frac{1}{4}(z_j)^2} \right) + \sqrt{\frac{2\pi}{3}} \\
&= \frac{\sqrt{2\pi}}{n^2} \sum_{j,k=1}^{n} e^{-\frac{1}{2}(z_j - z_k)^2} - \frac{\sqrt{2\pi} \cdot \sqrt{2}}{n} \sum_{j=1}^{n} \left(e^{-\frac{1}{4}(z_j)^2} \right) + \sqrt{\frac{2\pi}{3}} \\
&= \sqrt{2\pi} \cdot \left(\frac{1}{n^2} \sum_{j,k=1}^{n} e^{-\frac{1}{2}(z_j - z_k)^2} - \frac{\sqrt{2}}{n} \sum_{j=1}^{n} \left(e^{-\frac{1}{4}(z_j)^2} \right) + \frac{1}{\sqrt{3}} \right)
\end{aligned} \tag{13}
$$

We consider the ratio with the first term, that becomes 1 at initialization. This loss becomes identical if used with scaled MMD-Gauss, differ from that of BHEP.

Appendix D. Small λ assumption

If all $\lambda_i \ll d$, and $u_k \approx 1$, and we can formulate the derivation as:

$$\begin{aligned} P_1 &= \prod_{i=1}^{d} \left(1 + \frac{2}{d}\lambda_i\right)^{-1/2} \\ &\approx \exp\left(-\frac{1}{2}\sum_{i=1}^{d} \ln\left(1 + \frac{2}{d}\lambda_i\right)\right) \\ &\approx \exp\left(-\frac{1}{2}\sum_{i=1}^{d} \frac{2}{d}\lambda_i\right) \\ &\approx \exp(-\bar{\lambda}) \end{aligned} \tag{14}$$

$$\begin{aligned} P_2 &= \prod_{i=1}^{d} \left(1 + \frac{1}{d+1}\lambda_i\right)^{-1/2} \\ &\approx \exp\left(-\frac{1}{2}\sum_{i=1}^{d} \ln\left(1 + \frac{1}{d+1}\lambda_i\right)\right) \\ &\approx \exp\left(-\frac{1}{2}\sum_{i=1}^{d} \frac{1}{d+1}\lambda_i\right) \\ &\approx \exp\left(-0.5\bar{\lambda}\right) \end{aligned} \tag{15}$$

Where $\bar{\lambda} = \frac{1}{d}\sum_{i=1}^{d} \lambda_i$, mean of eigenvalues.

This formulation can be applied to the early starting phase ($\lambda \approx 0$), or later shrinking phase ($\lambda \approx k \ll d$), or convergence ($\lambda = 1$).

So, the overall formulation would be:

$$\begin{aligned} \dot{\lambda}_k &= 2\sigma_k^2 \cdot \lambda_k[(c_1 P_1 - \gamma c_2 P_2) - c_1 c_2(\gamma P_2 - P_1)\lambda_k] \\ &\approx 2\sigma_k^2 \cdot \lambda_k \left[\left(\frac{2}{d}e^{-\bar{\lambda}} - \frac{2}{\sqrt{e}(d+1)}e^{-0.5\bar{\lambda}}\right) - \frac{2}{d(d+1)}\left(\frac{2}{\sqrt{e}}e^{-0.5\bar{\lambda}} - e^{-\bar{\lambda}}\right)\lambda_k\right] \end{aligned} \tag{16}$$

This also proves that when all $\lambda_i = 1$, the overall global force becomes 0, corresponding to the global minimum of isotropic Gaussian.

Appendix E. Empirical validation of the Gaussian input assumption

As discussed in the main text, our dynamic analysis relies on the assumption that the top-d principal components of the inputs follow a multivariate Gaussian distribution. We assume that for any eigenvector $v_i \in V^{(\leq d)}$, the projected input feature $v_i^T x$ is approximately normally distributed.

To empirically validate this assumption, we visualize the feature distributions using the CIFAR-10 dataset. Figure 4 illustrates the empirical distribution of the top 32 projected component. While the actual distribution exhibits a higher peak (i.e., it is slightly leptokurtic) compared to a perfect theoretical Gaussian, it remains strictly unimodal, zero-centered, and symmetric. In the context of analyzing macroscopic learning dynamics and phase transitions, this structural similarity provides a sufficiently accurate proxy, justifying our Gaussian approximation framework.

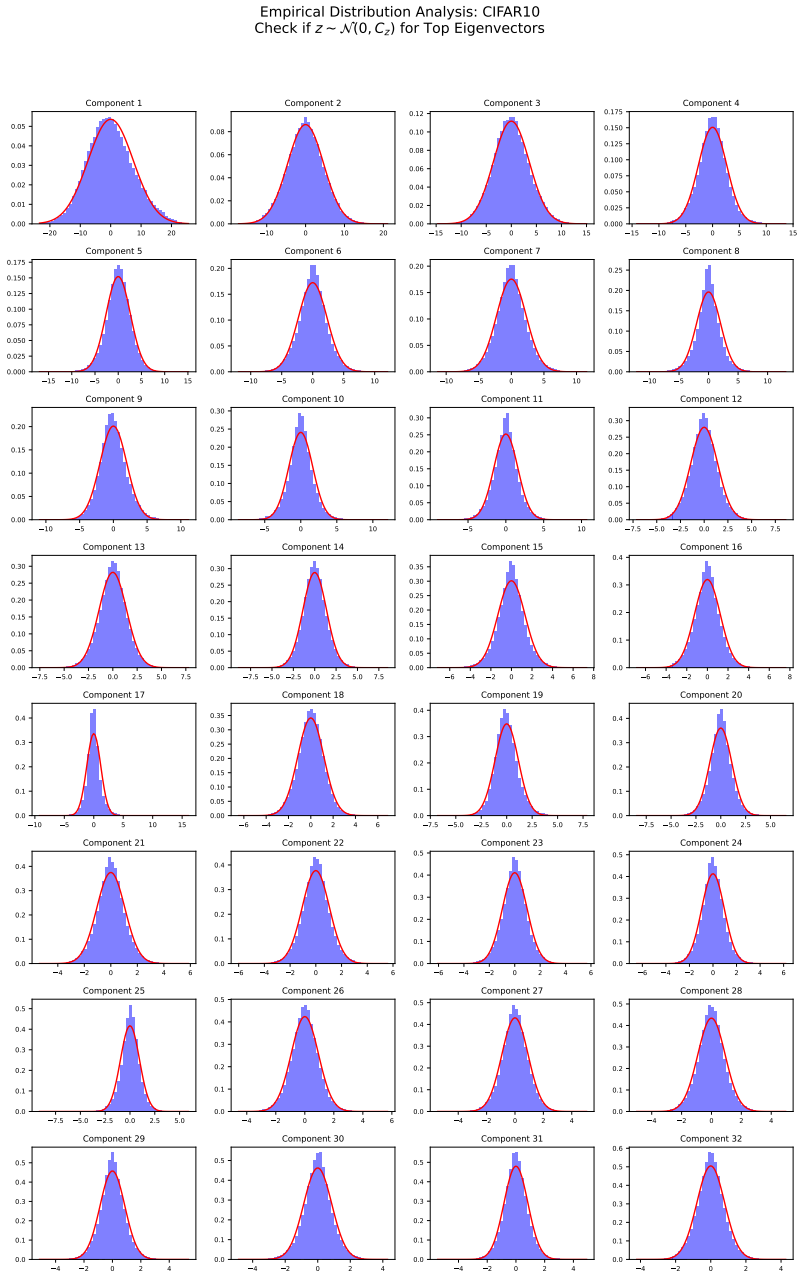

Figure 4: Empirical distribution of the top principal component of the CIFAR-10 dataset. The distribution is strongly unimodal and symmetric, supporting the macroscopic Gaussian approximation used in our theoretical derivations.

Appendix F. Learning dynamics at start

We start from eq (11), without replacing denominator into u_k, and use $\dot{\lambda}_k = 2\sigma_k^2 s_k \cdot \dot{s}_k$, then the form would be:

$$\dot{\lambda}_k = 2\sigma_k^2 \lambda_k \cdot \left[\frac{2}{d} \cdot \frac{\prod_{i=1}^{d}(1+c_1\lambda_i)^{-1/2}}{1+c_1\lambda_k} - \frac{2}{\sqrt{e}(d+1)} \cdot \frac{\prod_{i=1}^{d}(1+c_2\lambda_i)^{-1/2}}{1+c_2\lambda_k} \right]. \tag{17}$$

Assuming that the biggest eigenvalue λ_1 is the only eigenvalue being learned, then $\lambda_{i\neq 1} \approx 0$. Then, we can rewrite formula for λ_1 as:

$$\begin{aligned}
\dot{\lambda}_1 &= 2\sigma_1^2 \lambda_1 \cdot \left[\frac{2}{d} \cdot (1+c_1\lambda_1)^{-3/2} - \frac{2}{\sqrt{e}(d+1)} \cdot (1+c_2\lambda_1)^{-3/2} \right] \\
&\approx 2\sigma_1^2 \lambda_1 \cdot \left[\frac{2}{d} \cdot (1+\frac{2}{d}\lambda_1)^{-3/2} - \frac{2}{\sqrt{e}d} \cdot (1+\frac{1}{d}\lambda_1)^{-3/2} \right] \\
&\approx \frac{2\sigma_1^2}{d} \lambda_1 \cdot \left[2(1+\frac{2}{d}\lambda_1)^{-3/2} - \frac{2}{\sqrt{e}} \cdot (1+\frac{1}{d}\lambda_1)^{-3/2} \right]
\end{aligned} \tag{18}$$

where the dimension d is big enough (to approximate $\frac{1}{d} \approx \frac{1}{d+1}$).

Using first-order Taylor series to approximate, and retrieve $\dot{s}_1$ by using $\dot{\lambda}_k = 2\sigma_k^2 s_k \cdot \dot{s}_k$, then the formula becomes:

$$\begin{aligned}
\dot{s}_1 &\approx \frac{1}{d}\sigma_1^2 \cdot s_1 \cdot \left[\left(2 - \frac{2}{\sqrt{e}}\right) - \left(6 - \frac{3}{\sqrt{e}}\right) \cdot \frac{1}{d}\sigma_1^2 s_1^2 \right] \\
&\approx \frac{1}{d}\sigma_1^2 \cdot s_1 \cdot \left[A - B \cdot \frac{1}{d}\sigma_1^2 s_1^2 \right]
\end{aligned} \tag{19}$$

This resembles the ODE in Simon et al. [13], and we can derive critical time with same method:

$$\tau_1 = d \cdot \frac{-\log\left(\frac{\sigma_1^2}{d} s_1^2(0) \cdot \frac{(6-\frac{3}{\sqrt{e}})}{(2-\frac{2}{\sqrt{e}})} \right)}{2\sigma_1^2(2-\frac{2}{\sqrt{e}})}. \tag{20}$$

Although the analytical peak $\lambda_1^* \approx 0.654d$ deviates from the linear-approximation root ($\approx 0.188d$), τ_1 remains robust because exponential growth keeps the dynamics in the near-zero regime; both also scale linearly with $\mathcal{O}(d)$. See Figure 5 for empirical result.

Appendix G. Eigenvalue mean movement

Let's generalize from Appendix F. Let's assume there are m eigenvalues that are already trained $\lambda_i = \lambda_k$, and the other $(d-m)$ untrained $\lambda_i = 0$. Then we can expand formula as:

$$\begin{aligned}
\dot{\lambda}_k &= 2\sigma_k^2 \lambda_k \cdot \left[\frac{2}{d} \cdot \frac{\prod_{i=1}^{d}(1+c_1\lambda_i)^{-1/2}}{1+c_1\lambda_k} - \frac{2}{\sqrt{e}(d+1)} \cdot \frac{\prod_{i=1}^{d}(1+c_2\lambda_i)^{-1/2}}{1+c_2\lambda_k} \right] \\
&\approx \frac{4\sigma_k^2}{d} \lambda_k \cdot \left[(1+\frac{2}{d}\lambda_k)^{-\frac{m+2}{2}} - \frac{1}{\sqrt{e}} \cdot (1+\frac{1}{d}\lambda_k)^{-\frac{m+2}{2}} \right]
\end{aligned} \tag{21}$$

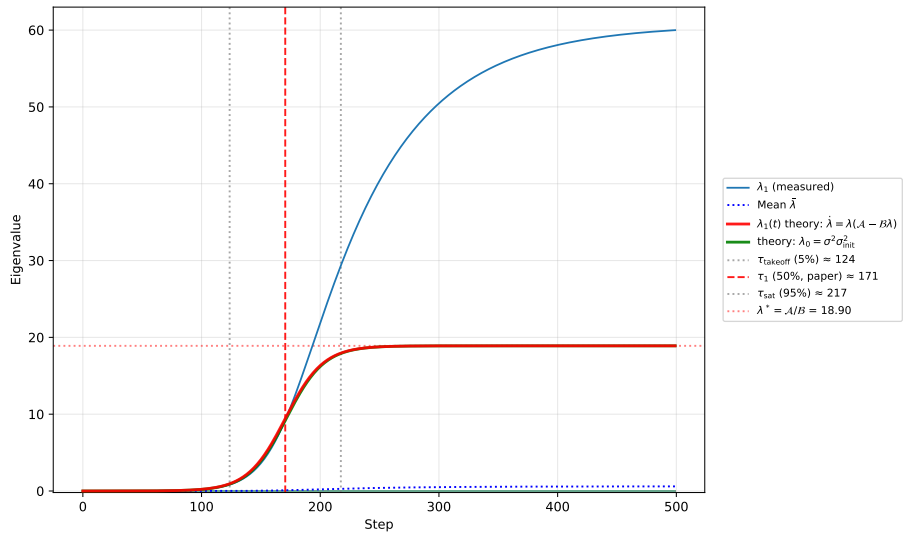

Figure 5: **Theoretical and empirical trajectory over 1st eigenvalue λ_1.** With our calculated trajectory, empirical trajectory of 1st eigenvalue λ_1 resembles our theoretical ODE and critical time at emergence, which λ_1 stays low in initialization, reducing error in Taylor expansion. *Red* curve used $\lambda_1(0)$ at $t = 0$ with initial calculation with covariance matrix C_Z. *Green* curve used $\lambda_1(0)$ at $t = 0$ calculated with initialization value over weight: $\lambda_1(0) = \sigma^2 s_1^2(0) = \sigma^2 \cdot (0.001)^2$.

As all those eigenvalues are assumed converged, so corresponding $\dot{\lambda}_k = 0$. Then,

$$
\begin{aligned}
(1 + \frac{2}{d}\lambda_k)^{-\frac{m+2}{2}} &= e^{-\frac{1}{2}} \cdot (1 + \frac{1}{d}\lambda_k)^{-\frac{m+2}{2}} \\
(1 + \frac{2}{d}\lambda_k) &= e^{\frac{1}{m+2}} \cdot (1 + \frac{1}{d}\lambda_k) \\
\therefore \lambda_k^* &= \frac{e^{\frac{1}{m+2}} - 1}{2 - e^{\frac{1}{m+2}}} d
\end{aligned} \tag{22}
$$

The overall eigenvalue mean, $\bar{\lambda} = \frac{1}{d}\sum_{i=1}^{d} \lambda_i = \frac{e^{\frac{1}{m+2}}-1}{2-e^{\frac{1}{m+2}}} m$. This eigenvalue mean starts from $\bar{\lambda} \approx 0.654$, and gradually increases with number of learned eigenvalues, and becomes 1 at convergence.

G.1. Exact mean eigenvalue without approximation

Upon approximation, in $m = d$, this does not exactly converge at 1, but slightly lower as a scale of $\mathcal{O}(\frac{1}{d})$. Without using any approximation, including the $\gamma = \frac{2}{(1+1/d)^{d/2}} \approx \frac{2}{\sqrt{e}}$ from Eq. (8) with

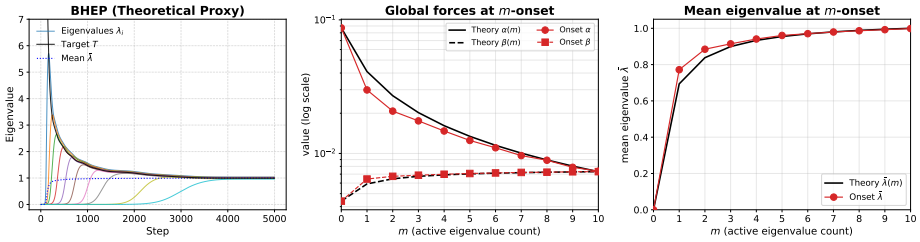

Figure 6: **Dynamics vs. number of active eigenvalues m for 10 dimension theoretical and empirical setting.** Via Eq. (5), $\bar{\lambda}$ jumps toward 1, α collapses, and β plateaus after a few features (Used 0.01x learning rate warmup for first 1000 epochs to observe discrete eigenvalue emergence).

$m = d$, then the ODE becomes exactly $\bar{\lambda}^* = 1$:

$$\begin{aligned}\dot{\lambda}_k &= 2\sigma_k^2\lambda_k \cdot \left[\frac{2}{d} \cdot \frac{\prod_{i=1}^d (1+c_1\lambda_i)^{-1/2}}{1+c_1\lambda_k} - \frac{2}{(1+1/d)^{d/2}(d+1)} \cdot \frac{\prod_{i=1}^d (1+c_2\lambda_i)^{-1/2}}{1+c_2\lambda_k}\right] \\ &= \frac{4\sigma_k^2}{d}\lambda_k \cdot \left[\frac{1}{d} \cdot (1+\frac{2}{d}\lambda_k)^{-\frac{m+2}{2}} - \frac{1}{(1+1/d)^{d/2}(d+1)} \cdot (1+\frac{1}{d}\lambda_k)^{-\frac{m+2}{2}}\right] \end{aligned} \tag{23}$$

This should be 0, then

$$\frac{1}{d} \cdot (1+\frac{2}{d}\lambda_k)^{-\frac{m+2}{2}} = \frac{1}{(1+1/d)^{d/2}(d+1)} \cdot (1+\frac{1}{d+1}\lambda_k)^{-\frac{m+2}{2}} \tag{24}$$

Let's assume $K[m,d] = \left(\frac{(1+1/d)^{d/2}(d+1)}{d}\right)^{2/(m+2)}$, then

$$(1+\frac{2}{d}\lambda_k) = K[m,d] \cdot (1+\frac{1}{d+1}\lambda_k) \tag{25}$$

$$\therefore \lambda^* = \frac{d(d+1)(K[m,d]-1)}{2(d+1)-dK[m,d]} \tag{26}$$

And the corresponding eigenvalue mean would be: $\bar{\lambda}^* = \frac{m(d+1)(K[m,d]-1)}{2(d+1)-dK[m,d]}$

Appendix H. Theory vs practice on the toy setting

To verify the steady-state predictions of Section 3.2 quantitatively, we directly compare the closed-form $\alpha(m)$, $\beta(m)$, and $\bar{\lambda}(m)$ against measurements from BHEP training on the 10-dim toy setting.

Figure 6 shows close quantitative agreement between the analytical predictions (with the exact $\bar{\lambda}^*$ from Appendix G.1) and the empirical trajectory. The acceleration of the empirical curves reflects simultaneous multi-feature activation in practice: when several eigenvalues grow concurrently, α-collapse and β-plateau both proceed faster than the one-by-one schedule assumed in the derivation.

Appendix I. Derivation of weight alignment and eigenvalue decoupling

In this section, we provide the detailed derivation for the eigenvalue decoupling, adopting the framework utilized to analyze stepwise learning in self-supervised learning architectures [13].

We assume that the effective weight matrix $W \in \mathbb{R}^{k\times d}$ dynamically aligns with the principal components of the input data $X \in \mathbb{R}^{n\times k}$ during training. Let the Singular Value Decomposition (SVD) of the input be $X = U\Sigma V^T$. The input covariance matrix is thus given by $C_X = \frac{1}{n}X^TX = \frac{1}{n}V\Sigma^2V^T$, where $\Sigma^2 = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_k^2)$ contains the input variances for each principal component.

We assume the weight matrix W structurally aligns with the top d eigenvectors of C_X:

$$W = V^{(\leq d)}SQ^T \tag{27}$$

where $V^{(\leq d)} \in \mathbb{R}^{k\times d}$ represents the top d eigenvectors, $S = \text{diag}(s_1, s_2, \ldots, s_d)$ contains the continuous, learnable weight coefficient for each feature, and Q is an orthogonal matrix.

Consequently, the embedding $Z = XW \in \mathbb{R}^{n\times d}$ exhibits the following covariance structure:

$$\begin{aligned} C_Z &= \frac{1}{n}Z^TZ = W^TC_XW \\ &= \frac{1}{n}QS(V^{(\leq d)})^T(V\Sigma^2V^T)V^{(\leq d)}SQ^T \\ &= \frac{1}{n}Q(S^2\Sigma^2_{\leq d})Q^T = \frac{1}{n}Q\Lambda Q^T \end{aligned} \tag{28}$$

Here, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$ represents the eigenvalues of the embedding covariance matrix. Crucially, this exact decoupling demonstrates that each macroscopic embedding eigenvalue λ_k is strictly determined by the constant input variance σ_k^2 and its corresponding weight eigenvalue s_k^2:

$$\lambda_k = \sigma_k^2 s_k^2 \tag{29}$$

Appendix J. Align MMD-Gauss with BHEP using gamma function

As noted above, the SIGReg regularizer term is more similar to MMD-Gauss than BHEP. So we also derive MMD-Gauss using the same method as BHEP.

J.1. Derive MMD-Gauss into gamma function

To advance the analysis of the MMD-Gaussian loss provided in Zimmermann et al. [17], we rewrite the empirical formula using the approximation proposed in the original paper. Also, as we made the assumption that n is big enough, we can neglect the approximation difference proposed in KerJEPA paper.

$$|\mathbb{E}[\widehat{\text{MMD-Gauss}}^2_{\kappa_d}(\hat{P}_n) - \widehat{\text{SIGReg}}(\hat{P}_n)]| \leq O(1/n) \approx 0 \tag{30}$$

Also, as we made an assumption that the dimension d is sufficiently big, we can use approximation ${}_1F_1(1/2, d/2, \gamma\|x-y\|_2^2) \approx k_{\text{imq}}^{\alpha,\beta}(x,y) \quad \text{for} \quad \alpha = 4\gamma(2d-3)^{-1}, \beta = 1/2$, which can be used for practical situations at $d > 20$.

$$\widehat{\text{MMD-Gauss}}^2_{\kappa_d}(\hat{P}_n) \approx \frac{1}{n(n-1)}\sum_{j\neq i}\left(1 + \frac{2}{2d-3}\|z_i - z_j\|_2^2\right)^{-1/2} - \frac{\sqrt{2}}{n}\sum_{i=1}^{n}\left(1 + \frac{1}{2d-3}\|z_i\|_2^2\right)^{-1/2} + C \tag{31}$$

To facilitate further analytical derivation, we leverage the Gamma function identity:

$$\frac{1}{\sqrt{Z}} = \frac{1}{\sqrt{\pi}} \int_0^\infty t^{-1/2} e^{-Zt} dt \tag{32}$$

which allows us to convert the inverse square-root terms into exponential forms. Assuming the number of samples n is sufficiently large such that $\frac{1}{n(n-1)} \approx \frac{1}{n^2}$, we can rewrite the empirical loss as:

$$\widehat{\text{MMD-Gauss}}^2(\hat{P}_n) \approx \frac{1}{\sqrt{\pi}} \int_0^\infty t^{-1/2} e^{-t} \left[\frac{1}{n^2} \sum_{i,j=1}^{n} e^{-\frac{2t}{2d-3}\|z_i - z_j\|_2^2} - \frac{\sqrt{2}}{n} \sum_{i=1}^{n} e^{-\frac{t}{2d-3}\|z_i\|_2^2} \right] dt + C \tag{33}$$

where the first term of $i = j$, always $e^0 = 1$, is pulled from constant term C.

Using the same method as BHEP, we can proceed as:

$$\begin{aligned}
\widehat{\text{MMD-Gauss}}^2(\hat{P}_n) &\approx \frac{1}{\sqrt{\pi}} \int_0^\infty t^{-1/2} e^{-t} \Bigg[\det\left(I + \frac{8t}{2d-3} C_z\right)^{-1/2} \qquad (34) \\
&\qquad - \sqrt{2} \det\left(I + \frac{2t}{2d-3} C_z\right)^{-1/2} \Bigg] dt + C \\
&= \frac{1}{\sqrt{\pi}} \int_0^\infty t^{-1/2} e^{-t} \Bigg[\underbrace{\prod_{i=1}^{d} (1 + c_1(t)\lambda_i)^{-1/2}}_{P_1(t)} - \sqrt{2} \underbrace{\prod_{i=1}^{d} (1 + c_2(t)\lambda_i)^{-1/2}}_{P_2(t)} \Bigg] dt + C \\
&= \frac{1}{\sqrt{\pi}} \int_0^\infty t^{-1/2} e^{-t} \Big[P_1(t) - \sqrt{2} P_2(t) \Big] dt + C \qquad (35)
\end{aligned}$$

where we define $s = \frac{2}{2d-3}$, $c_1(t) = 4ts$, and $c_2(t) = ts$ for brevity.

J.2. Deriving the dynamics with MMD-Gauss

Setting aside t, the overall derivation is same:

$$\dot{s}_k = \frac{\sigma_k^2 s_k}{\sqrt{\pi}} \int_0^\infty t^{-1/2} e^{-t} \left[P_1(t) \frac{c_1(t)}{1 + c_1(t)\lambda_k} - \sqrt{2} P_2(t) \frac{c_2(t)}{1 + c_2(t)\lambda_k} \right] dt \tag{36}$$

Using $\dot{\lambda}_k = 2\sigma_k^2 s_k \dot{s}_k$,

$$\begin{aligned}
\dot{\lambda}_k &= 2\sigma_k^2 \lambda_k \cdot \frac{1}{\sqrt{\pi}} \int_0^\infty t^{-1/2} e^{-t} \left[P_1(t) \frac{4ts}{1 + c_1(t)\lambda_k} - \sqrt{2} P_2(t) \frac{ts}{1 + c_2(t)\lambda_k} \right] dt \\
&= 2\sigma_k^2 \lambda_k \cdot \frac{1}{\sqrt{\pi}} \int_0^\infty t^{1/2} e^{-t} \left[\frac{4P_1(t)}{1 + c_1(t)\lambda_k} - \frac{\sqrt{2} P_2(t)}{1 + c_2(t)\lambda_k} \right] \cdot s\, dt \qquad (37)
\end{aligned}$$

$$\dot{\lambda}_k = \frac{2\sigma_k^2 \lambda_k}{\sqrt{\pi}} \int_0^\infty \frac{t^{1/2} e^{-t}}{(1 + c_1(t)\lambda_k)(1 + c_2(t)\lambda_k)} \Big[4P_1(t)(1 + c_2(t)\lambda_k) - \sqrt{2} P_2(t)(1 + c_1(t)\lambda_k) \Big] \cdot s\, dt \tag{38}$$

Unlike BHEP, the denominator u_k is dependent on t, as we contain it inside of each coefficient $c_1(t), c_2(t)$. So, instead we use $u_{t,k}$:

$$\begin{aligned}\dot{\lambda}_k &= \frac{2\sigma_k^2\lambda_k}{\sqrt{\pi}}\int_0^\infty u_{t,k}\cdot t^{1/2}e^{-t}\cdot\Big[P_1(t)(4+4c_2(t)\lambda_k)-\sqrt{2}P_2(t)(1+c_1(t)\lambda_k)\Big]\cdot s\,dt\\ &= \frac{2\sigma_k^2\lambda_k}{\sqrt{\pi}}\int_0^\infty u_{t,k}\cdot t^{1/2}e^{-t}\cdot\Big[\big(4P_1(t)-\sqrt{2}P_2(t)\big)-\big(4\sqrt{2}P_2(t)-4P_1(t)\big)ts\lambda_k\Big]\cdot s\,dt\\ &= \frac{2\sigma_k^2\lambda_k}{\sqrt{\pi}}\int_0^\infty u_{t,k}\cdot t^{1/2}e^{-t}\cdot\left[\alpha_t-\beta_t\cdot\lambda_k\right]\,dt \qquad (39)\end{aligned}$$

where we define the t-dependent coefficients $\alpha_t = s\cdot(4P_1(t)-\sqrt{2}P_2(t))$ and $\beta_t = s^2t\cdot(4\sqrt{2}P_2(t)-4P_1(t))$.

Then, the final derivation would be like:

$$\dot{\lambda}_k = \underbrace{\frac{2\sigma_k^2}{\sqrt{\pi}}}_{\text{Local force}}\cdot\lambda_k\cdot\underbrace{\int_0^\infty u_{t,k}\cdot t^{1/2}e^{-t}\big[\alpha_t-\beta_t\lambda_k\big]\,dt}_{\text{Global force}} \qquad (40)$$

Which is quite similar to the BHEP-method.

J.3. Target eigenvalue T_k

Unlike the BHEP-method, using Target eigenvalue $T = \frac{\alpha}{\beta}$, the target eigenvalue for MMD-Gauss is more complicated as they are t-dependent and cannot be simplified:

$$T_k = \frac{\int_0^\infty u_{t,k}\cdot t^{1/2}e^{-t}\cdot\alpha_t dt}{\int_0^\infty u_{t,k}\cdot t^{1/2}e^{-t}\cdot\beta_t dt} \qquad (41)$$

Not only this is hard to compute, but they are even k-dependent, which means the target eigenvalue differs by their own eigenvalues. If λ_k gets bigger, denominator becomes smaller, while the numerator becomes bigger, resulting in the bigger target eigenvalue T_k, and vice versa.

J.4. Small λ assumption

Thanks to approximation, we can convert each value into exponential form, and offset the gamma integration. We can formulate the derivation as:

$$\begin{aligned}\alpha_t &= (4P_1(t)-\sqrt{2}P_2(t))\cdot s\\ &= \left(4\prod_{i=1}^d\left(1+\frac{8t}{2d-3}\cdot\lambda_i\right)^{-1/2}-\sqrt{2}\prod_{i=1}^d\left(1+\frac{2t}{2d-3}\cdot\lambda_i\right)^{-1/2}\right)\cdot s\\ &\approx (4e^{-2t\bar{\lambda}}-\sqrt{2}e^{-\frac{1}{2}t\bar{\lambda}})\cdot s\end{aligned} \qquad (42)$$

$$\begin{aligned}\beta_t &= (4\sqrt{2}P_2(t)-4P_1(t))\cdot s^2t\\ &\approx \left(4\sqrt{2}e^{-\frac{1}{2}t\bar{\lambda}}-4e^{-2t\bar{\lambda}}\right)\cdot s^2t \qquad (43)\end{aligned}$$

$$
\begin{aligned}
\dot{\lambda}_k &\approx \frac{2\sigma_k^2}{\sqrt{\pi}} \cdot \lambda_k \cdot \int_0^\infty t^{1/2} e^{-t} \cdot \Bigg[\left(4e^{-2t\bar{\lambda}} - \sqrt{2}e^{-\frac{1}{2}t\bar{\lambda}}\right) \\
&\qquad - \left(4\sqrt{2}e^{-\frac{1}{2}t\bar{\lambda}} - 4e^{-2t\bar{\lambda}}\right) \cdot t \cdot s\lambda_k \Bigg] \cdot s\, dt \\
&= \frac{2\sigma_k^2}{\sqrt{\pi}} \cdot \lambda_k \cdot \int_0^\infty \Bigg[t^{1/2} \left(4e^{-(2\bar{\lambda}+1)t} - \sqrt{2}e^{-\left(\frac{1}{2}\bar{\lambda}+1\right)t}\right) \\
&\qquad - t^{3/2} \left(4\sqrt{2}e^{-\left(\frac{1}{2}\bar{\lambda}+1\right)t} - 4e^{-(2\bar{\lambda}+1)t}\right) \cdot s\lambda_k \Bigg] s\, dt \\
&= \frac{2\sigma_k^2}{\sqrt{\pi}} \cdot \lambda_k \cdot \Bigg[\left(\frac{4\sqrt{\pi}}{2(2\bar{\lambda}+1)^{3/2}} - \frac{\sqrt{2}\sqrt{\pi}}{2\left(\frac{1}{2}\bar{\lambda}+1\right)^{3/2}} \right) \\
&\qquad - \left(\frac{4\sqrt{2}\cdot 3\sqrt{\pi}}{4\left(\frac{1}{2}\bar{\lambda}+1\right)^{5/2}} - \frac{4\cdot 3\sqrt{\pi}}{4\left(2\bar{\lambda}+1\right)^{5/2}} \right) \cdot s\lambda_k \Bigg] s \\
&= 2\sigma_k^2 \cdot \lambda_k \cdot \Bigg[\left(\frac{2}{(2\bar{\lambda}+1)^{3/2}} - \frac{1}{\sqrt{2}\left(\frac{1}{2}\bar{\lambda}+1\right)^{3/2}} \right) \\
&\qquad - \left(\frac{3\sqrt{2}}{\left(\frac{1}{2}\bar{\lambda}+1\right)^{5/2}} - \frac{3}{\left(2\bar{\lambda}+1\right)^{5/2}} \right) \cdot s\lambda_k \Bigg] s
\end{aligned} \tag{44}
$$

J.5. Asymptotic target eigenvalue derivation

Unlike BHEP-method, we mathematically derive the exact equilibrium state. Let us assume a generalized uniform eigenvalue formulation $\lambda_i = \lambda_{\text{ideal}} = a + \frac{b}{d}$, where a is the leading-order scale $\mathcal{O}(1)$ and b dictates the finite-dimensional geometric residual $\mathcal{O}(1/d)$.

Starting from the combined dynamics Eq. (37), as we want to make derivative to be 0, we can neglect the prior coefficient, which is the same for both push, pull term. Also, we can merge the denominator into the push and pull terms since all eigenvalues are equal:

$$
\dot{\lambda}_{\text{ideal}} \propto \int_0^\infty t^{1/2} e^{-t} \left[4\big(1 + c_1(t)\lambda_{\text{ideal}}\big)^{-\left(\frac{d}{2}+1\right)} - \sqrt{2}\big(1 + c_2(t)\lambda_{\text{ideal}}\big)^{-\left(\frac{d}{2}+1\right)} \right] dt = 0 \tag{45}
$$

J.5.1. TWO-SCALE EXPANSION WITH ARGUMENTS

We first approximate the scaling factor $s = \frac{2}{2d-3} \approx \frac{1}{d}(1 + \frac{3}{2d})$. The generalized push argument $c_1(t)\lambda_{\text{ideal}}$ expands as:

$$
c_1(t)\lambda_{\text{ideal}} = 4st\left(a + \frac{b}{d}\right) \approx \left[\frac{4t}{d}\left(1 + \frac{3}{2d}\right)\right]\left(a + \frac{b}{d}\right) \approx \frac{4at}{d} + \frac{4bt + 6at}{d^2} \tag{46}
$$

Using the Taylor series $\log(1+x) \approx x - x^2/2$,

$$
\begin{aligned}
\log\left(1 + c_1(t)\lambda_{\text{ideal}}\right) &\approx \left(\frac{4at}{d} + \frac{4bt + 6at}{d^2}\right) - \frac{1}{2}\left(\frac{4at}{d}\right)^2 \\
&= \frac{4at}{d} + \frac{4bt + 6at - 8a^2t^2}{d^2}
\end{aligned} \tag{47}
$$

Multiplying by the exponent $-(d/2+1)$:

$$-\left(\frac{d}{2}+1\right)\log\left(1+c_1(t)\lambda_{\text{ideal}}\right) \approx -\frac{d}{2}\left(\frac{4at}{d}+\frac{4bt+6at-8a^2t^2}{d^2}\right)-1\left(\frac{4at}{d}\right)$$
$$= -2at+\frac{4a^2t^2-(2b+7a)t}{d} \tag{48}$$

Exponentiating back using $e^x \approx 1+x$ yields the push term:

$$\left(1+c_1(t)\lambda_{\text{ideal}}\right)^{-\left(\frac{d}{2}+1\right)} \approx e^{-2at}\left[1+\frac{4a^2t^2-(2b+7a)t}{d}\right] \tag{49}$$

By applying the exact same systematic expansion to the pull term $c_2(t)\lambda_{\text{ideal}} = st(a+b/d)$, we obtain:

$$\left(1+c_2(t)\lambda_{\text{ideal}}\right)^{-\left(\frac{d}{2}+1\right)} \approx e^{-\frac{1}{2}at}\left[1+\frac{\frac{1}{4}a^2t^2-(\frac{1}{2}b+\frac{7}{4}a)t}{d}\right] \tag{50}$$

J.5.2. LEADING-ORDER $\mathcal{O}(1)$ ANALYSIS: DERIVING $a=1$

To solve for the macroscopic equilibrium state, we take the infinite dimensional limit ($d \to \infty$), which leaves only the $\mathcal{O}(1)$ leading terms inside the integral:

$$\int_0^\infty t^{1/2}e^{-t}\left(4e^{-2at}-\sqrt{2}e^{-\frac{1}{2}at}\right)dt = 0 \tag{51}$$

Rearranging the exponents, we evaluate the Gamma integral $\int_0^\infty t^{1/2}e^{-ct}dt = \frac{\sqrt{\pi}}{2c^{3/2}}$:

$$4\int_0^\infty t^{1/2}e^{-(2a+1)t}dt = \sqrt{2}\int_0^\infty t^{1/2}e^{-(\frac{1}{2}a+1)t}dt$$
$$\frac{4}{(2a+1)^{3/2}} = \frac{\sqrt{2}}{(\frac{1}{2}a+1)^{3/2}} \tag{52}$$

Squaring both sides to eliminate the fractional exponents:

$$\frac{16}{(2a+1)^3} = \frac{2}{(0.5a+1)^3}$$
$$8(\frac{1}{2}a+1)^3 = (2a+1)^3 \tag{53}$$

Since $8(\frac{1}{2}a+1)^3 = 8\left(\frac{a+2}{2}\right)^3 = (a+2)^3$, the equation elegantly simplifies to:

$$(a+2)^3 = (2a+1)^3$$
$$a+2 = 2a+1$$
$$\mathbf{a = 1} \tag{54}$$

This result mathematically proves that without any prior assumptions, the self-supervised Euclidean dynamics inherently gravitate toward the isotropic Gaussian state ($\lambda_i \approx 1$).

J.5.3. RESIDUAL-ORDER $\mathcal{O}(1/d)$ ANALYSIS: DERIVING $b = -1$

Having established $a = 1$, we substitute it back into the full integral to resolve the finite-dimensional geometric residual parameter b. The $\mathcal{O}(1)$ terms perfectly cancel out, leaving the $\mathcal{O}(1/d)$ residual integral equating to zero:

$$\int_0^\infty t^{1/2} \left(4e^{-3t} \left[4t^2 - (2b+7)t \right] - \sqrt{2} e^{-\frac{3}{2}t} \left[\frac{1}{4}t^2 - (\frac{1}{2}b + \frac{7}{4})t \right] \right) dt = 0 \tag{55}$$

Using the properties of the Gamma function $\Gamma(p)$: For the push residual:

$$4 \left(\frac{4\Gamma(\frac{7}{2})}{3^{\frac{7}{2}}} - \frac{(2b+7)\Gamma(\frac{5}{2})}{3^{\frac{5}{2}}} \right) = \frac{\sqrt{\pi}}{3\sqrt{3}} \left[\frac{10}{3} - (2b+7) \right] \tag{56}$$

For the pull residual:

$$\sqrt{2} \left(\frac{\frac{1}{4}\Gamma(\frac{7}{2})}{(\frac{3}{2})^{\frac{7}{2}}} - \frac{(\frac{1}{2}b + \frac{7}{4})\Gamma(\frac{5}{2})}{(\frac{3}{2})^{\frac{5}{2}}} \right) = \frac{\sqrt{\pi}}{3\sqrt{3}} \left[\frac{5}{6} - (b + \frac{7}{2}) \right] \tag{57}$$

Equating the residual terms:

$$\begin{aligned} \left[\frac{10}{3} - (2b+7) \right] - \left[\frac{5}{6} - (b + \frac{7}{2}) \right] &= 0 \\ -2b - \frac{11}{3} + b + \frac{8}{3} &= 0 \\ -b - 1 &= 0 \\ \mathbf{b} &= \mathbf{-1} \end{aligned} \tag{58}$$

By combining the leading and residual orders, we conclude that the precise equilibrium eigenvalue in a finite-dimensional embedding space is inherently bounded by:

$$\lambda^* = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{d}} \tag{59}$$

Appendix K. Effect of augmentation on dynamics

While deriving formulations, we assumed that invariance term does not affect on learning dynamics: already saturated. However, in practical setting, it is highly unlikely, as augmentations are applied to promote model to learn invariant features. So, in this section, we are adding gradient over invariant term.

We have additional assumption that:

- Augmentation shares same SVD over input, that augmentation $A = U\Sigma_A V^T$ (Appendix I).
- Augmentation is also distributed over multivariate Gaussian.

Then, the original input would be modified as:

$$X' = X + A \qquad [A \sim C_A] \tag{60}$$

Where $C_A = \frac{1}{n}A^TA = \frac{1}{n}V\Sigma_A^2V^T$. So, covariance matrix C_Z would be modified as

$$\begin{aligned} C_{Z'} &= \frac{1}{n}Z'^TZ' = W^T(C_X + C_A)W \\ &= \frac{1}{n}QS(V^{(\leq d)})^T(V(\Sigma^2 + \Sigma_A^2)V^T)V^{(\leq d)}SQ^T \\ &= \frac{1}{n}Q(S^2(\Sigma_{\leq d}^2 + \Sigma_{A:\leq d}^2))Q^T = \frac{1}{n}Q\Lambda'Q^T \end{aligned} \tag{61}$$

Here, $\Lambda' = \text{diag}(\lambda_1', \lambda_2', \ldots, \lambda_d') = \text{diag}((\sigma_1^2 + a_1^2)s_1^2, (\sigma_2^2 + a_2^2)s_2^2, \ldots, (\sigma_n^2 + a_n^2)s_n^2)$.
Then for the original derivation for BHEP (Theorem 3), would be:

$$\dot{\lambda}_k = \underbrace{2(\sigma_k^2 + a_k^2)}_{\text{Local force}} \cdot \lambda_k \cdot \underbrace{u_k \cdot [\alpha - \beta\lambda_k]}_{\text{Global force}}. \tag{62}$$

K.1. Gradient from invariant term

Using dimension-agnostic invariant term with 2 views, it would be:

$$L_{\text{inv}} = \frac{1}{nd}\sum_{i=1}^{n}\|z_i - z_i'\|_2^2 \tag{63}$$

$$= \frac{1}{nd}\|Z - Z'\|_F^2 \tag{64}$$

$$= \frac{1}{nd}\|(X - X')W\|_F^2 \tag{65}$$

$$= \frac{1}{nd}\text{tr}\left(W^T(X - X')^T(X - X')W\right) \tag{66}$$

$$= \frac{1}{d}\text{tr}\left(W^T(2C_A)W\right) \tag{67}$$

$$= \frac{2}{d}\sum_{i=1}^{d}a_i^2s_i^2 \tag{68}$$

So, the following derivative would be like (using $\lambda_i = a_i^2s_i^2$):

$$\dot{\lambda}_i = 2a_i^2s_i \cdot \dot{s}_i \tag{69}$$

$$= 2a_i^2s_i \cdot (-\frac{dL_{\text{inv}}}{ds_i}) \tag{70}$$

$$= 2a_i^2s_i \cdot (-\frac{4a_i^2s_i}{d}) \tag{71}$$

$$= -\frac{8}{d}a_i^2 \cdot \lambda_i \tag{72}$$

Then the unified loss term would be derived as:

$$\dot{\lambda}_k = 2(\sigma_k^2 + a_k^2) \cdot \lambda_k \cdot u_k \cdot [\alpha - \beta\lambda_k] - \frac{8}{d}a_k^2 \cdot \lambda_k \tag{73}$$

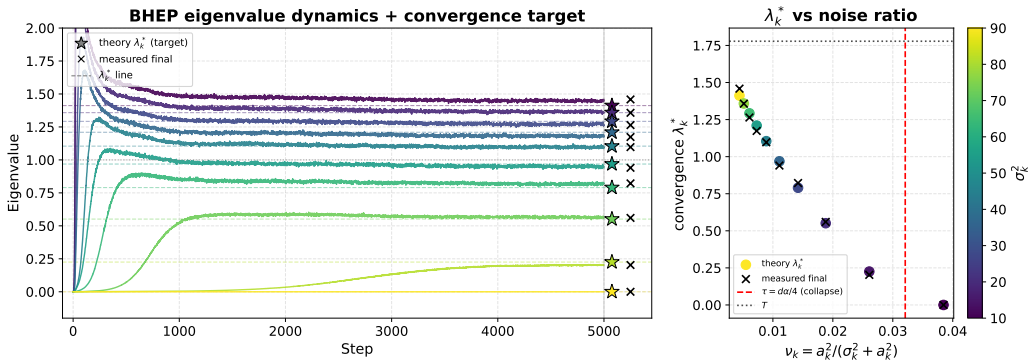

Figure 7: **Learning dynamics with augmentation.** Depending on input variance, features differ not only in their emergence times but also in the eigenvalues they converge to.

K.2. Augmentation: removal of features

By linearizing the dynamics around $\lambda_k \approx 0$ (with $\lambda_k > 0$), we obtain the effective rate at which an eigenvalue grows as it leaves the origin. If this rate is negative, the eigenvalue cannot emerge at all, and the corresponding feature is effectively removed. This gives a simple criterion for predicting whether a feature survives or is pruned.

At $\lambda_k = 0$ we have $u_k \approx 1$, so the linearized rate reads

$$\dot{\lambda}_{k,0} = \left(2(\sigma_k^2 + a_k^2)\,\alpha - \frac{8}{d}\,a_k^2\right)\lambda_k. \tag{74}$$

The sign of the bracket is governed by the ratio of augmentation variance to total variance. Defining the augmentation ratio ν_k and the threshold τ as

$$\nu_k = \frac{a_k^2}{\sigma_k^2 + a_k^2}, \qquad \tau = \frac{d\alpha}{4}, \tag{75}$$

whether the collapse of the feature is determined by comparing ν_k with τ:

- if $\nu_k > \tau$, the rate is negative and the correlated feature collapses;
- if $\nu_k < \tau$, the rate is positive and the correlated feature survives.

Two caveats complicate a direct application of this criterion. First, consistent with the role of the invariance term, the model can substitute a more augmentation-invariant feature (one with smaller ν_k) for a collapsing one, so removal is not determined for each feature in isolation. Second, the criterion presupposes a saturated α, so it can only be evaluated once learning has fully saturated.

K.3. Adaptation to the Model

We test the theory on a 10-dimensional toy model analogous to the one in the main section. The input variance is spread over $\sigma_k^2 \in [1.0^2, 3.0^2]$ while the augmentation variance is held fixed at

$a_k^2 = 0.2^2$. As a result, as seen in figure 7, ν_k varies across features, and features whose ν_k exceeds the threshold collapse.

The same analysis also predicts the value each surviving eigenvalue converges to, which is no longer the unaugmented target of 1. Starting from Eq. K.1 and setting $\dot{\lambda}_k = 0$, the fixed point satisfies

$$u_k \left(\alpha - \beta\lambda_k\right) = \frac{4}{d}\,\nu_k. \tag{76}$$

Writing $R_k = \frac{4}{d}\nu_k$ and substituting $u_k = \frac{1}{(1+c_1\lambda_k)(1+c_2\lambda_k)}$,

$$\alpha - \beta\lambda_k = (1 + c_1\lambda_k)(1 + c_2\lambda_k)\, R_k. \tag{77}$$

Expanding and collecting terms yields the quadratic

$$(R_k c_1 c_2)\,\lambda_k^2 + \big(\beta + R_k(c_1 + c_2)\big)\,\lambda_k + (R_k - \alpha) = 0, \tag{78}$$

whose physically relevant (positive) root is

$$\lambda_k^* = \frac{-\big(\beta + R_k(c_1 + c_2)\big) + \sqrt{\big(\beta + R_k(c_1 + c_2)\big)^2 - 4R_k c_1 c_2 (R_k - \alpha)}}{2R_k c_1 c_2}. \tag{79}$$

This mechanism accounts for two effects observed in practice (Appendix L):

- A target T that drifts across eigenvalues rather than sitting at a fixed value, and
- "Stacked" eigenvalues that settle at different levels instead of collapsing onto a single line.

Appendix L. Practical Experiments with CIFAR-10

We further validate our theoretical predictions on a realistic setting: SSL pre-training on CIFAR-10 [12] with a ResNet-18 [10] backbone ($d = 128$), built on the EB-JEPA library [14]. We compare all three regularizers (SIGReg, MMD-Gauss, and BHEP) under two initialization regimes — *normal* (the default LeJEPA setting) and *infinitesimal* (reduced LR by 100×, no BatchNorm in projector, small weight init; see Appendix N for details).

L.1. Stepwise learning is hidden under normal training

Why two initialization regimes? Our theoretical analysis predicts that the explosion-and-shrink lifecycle unfolds over $\mathcal{O}(d)$ time steps. In practice, with standard learning rates and standard initialization, this entire lifecycle is compressed into the first few hundred iterations — well within the first epoch — and is therefore invisible in typical training curves (Figure 8, top row). To expose the predicted dynamics, we slow time down via infinitesimal initialization (bottom row), which dilates the explosion phase across many epochs.

Observed phases. In the infinitesimal regime (Figure 8, bottom row), all three regularizers reproduce the lifecycle predicted by our BHEP analysis: (i) the tail of the explosion phase is clearly captured — top-1 eigenvalue is caught mid-rise and reaches its peak within the first few epochs; (ii) lower-ranked eigenvalues emerge sequentially after exponentially long delays — exactly as predicted by the asymmetric decay of α versus the plateaued β (Section 3); and (iii) all active eigenvalues eventually settle toward $\bar{\lambda} \to 1$. In the normal regime (top row), the same dynamics occur but are compressed into the first epoch, making the explosion barely visible.

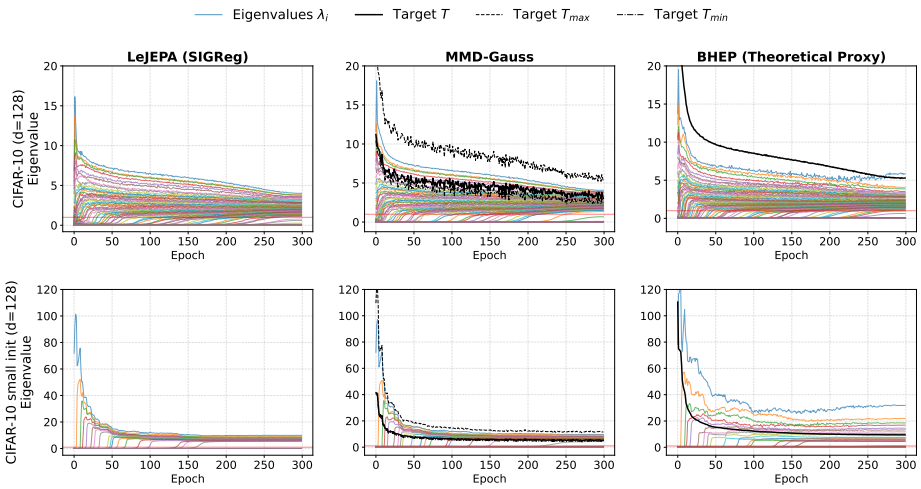

Figure 8: **Eigenvalue dynamics on CIFAR-10 (ResNet-18, $d = 128$).** Columns: SIGReg, MMD-Gauss, BHEP. *Top row (normal init):* the explosion phase is compressed into the first epoch and barely visible. *Bottom row (infinitesimal init):* the tail of the explosion phase is clearly resolved — the top-1 eigenvalue is caught mid-rise — followed by sequential emergence of lower-ranked eigenvalues. The black line indicates the target T (as in Figure 3). Quantitative deviations are discussed in Appendix L.4.

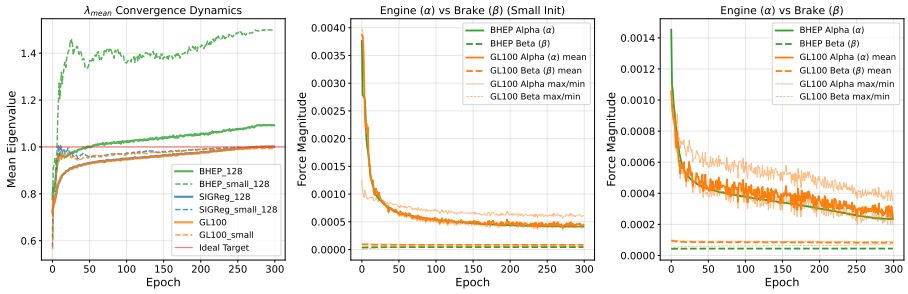

Figure 9: $\bar{\lambda}, \alpha, \beta$ **dynamics with CIFAR-10.** While BHEP proxy overshoots, both SIGReg and MMD-Gauss converge toward 1.

L.2. Asymptotic convergence: SIGReg and MMD-Gauss reach $\lambda^* = 1$

In our CIFAR-10 setup, both SIGReg and MMD-Gauss converge strictly to $\bar{\lambda} = 1$ (Figure 9), matching the exact equilibrium of the underlying SIGReg objective. The BHEP proxy systematically deviates from this target, settling at $\bar{\lambda} > 1.4$ — a kernel-specific artifact analyzed in Appendix L.4 (Artifact 1). The $\lambda^* = 1 - 1/d$ residual derived in Appendix J.5 arises from the MMD-Gauss kernel approximation and vanishes as d grows ($1 - 1/128 \approx 0.992$).

L.3. Engine α collapses, brake β plateaus — even in practice

Figure 9 also tracks the global forces α and β over training. Despite the deep nonlinear network and real data, the qualitative asymmetry predicted by our BHEP analysis holds remarkably well: β remains essentially constant after the first few features saturate, while α drops by orders of magnitude over the course of training. This is the empirical signature of the engine–brake mechanism (Section 3.2) operating outside the controlled toy setting.

The driver of α-suppression: discrete saturation count, not mean magnitude. A particularly striking observation concerns the relationship between the engine's strength and the mean eigenvalue. Under infinitesimal initialization, $\bar{\lambda}$ is actually slightly *higher* than under normal initialization at a comparable training stage, yet the engine α is also significantly *stronger*. This is counterintuitive if one assumes α is monotonically suppressed by $\bar{\lambda}$, but is exactly consistent with our steady-state analysis: α is governed by the discrete number m of saturated features, not their bulk mean (Section 3.2, with Eq. (5)). This empirical finding provides direct support for the discrete, stepwise nature of the engine collapse predicted by our ODE.

L.4. Observed deviations from theory: two artifacts

While the qualitative engine–brake dynamics hold robustly across all three regularizers, two quantitative deviations from our theoretical predictions emerge in the CIFAR-10 setting. We document both transparently and discuss their likely origins, noting that neither affects our main theoretical claims.

Artifact 1 — BHEP overshoot under infinitesimal initialization. Under infinitesimal initialization, the BHEP proxy converges to $\bar{\lambda} \approx 1.4$ rather than the theoretical target $\lambda^* = 1$ (Figure 9). The overshoot is most pronounced in this regime, where the absolute number of saturated features remains small. We attribute this to a structural difference between the kernel families: the inverse-root form of SIGReg and MMD-Gauss is long-tailed and continues to exert a corrective gradient on far-from-target eigenvalues, whereas BHEP's exponential kernel decays rapidly and fails to penalize eigenvalues that have travelled far beyond the target. The effect is therefore expected to be specific to the exponential proxy and does not affect the SIGReg/MMD-Gauss convergence behaviour that our theory directly targets.

Artifact 2 — Inflated target T under normal initialization. Under normal initialization, the empirically measured target $T = \alpha/\beta$ sits noticeably above the theoretical prediction across all regularizers. Inspecting Figure 8 (top row) reveals the underlying cause: rather than all saturated eigenvalues converging to a common T, they settle at distinct 'tiers' determined by their respective explosion peaks—a residue of the rapid Phase 1 dynamics that the normal setting compresses into the first epoch. Because α depends multiplicatively on the entire spectrum (Appendix A), this stratification inflates α above its theoretical value, lifting $T = \alpha/\beta$ accordingly. The infinitesimal setting, by stretching out Phase 1, gives features enough time to merge into a common target and recovers the predicted behaviour more cleanly.

Scope of these artifacts. Both deviations concern *quantitative* fits of specific numerical targets under specific training regimes. They do not affect the qualitative claims that drive our analysis: (i) the existence of distinct explosion and shrinking phases, (ii) the asymmetric decay of α versus β, and (iii) the stepwise emergence of lower-ranked features. All three are clearly observed across all three regularizers and both initialization regimes.

Appendix M. Limitations

While our framework provides the first closed-form analysis of LeJEPA's feature learning dynamics, several theoretical and practical limitations should be acknowledged.

Theoretical Assumptions. To ensure mathematical tractability and derive the macroscopic ODEs, our analysis relies on several idealized assumptions:

- We assume high dimensionality d and a sufficiently large sample size n ($n > d$) to support the asymptotic expansions.
- Following the linear network framework, we assume that the weight matrix W structurally aligns with the top-d principal components of the input covariance matrix during training.
- The projected input features are assumed to follow a multivariate Gaussian distribution, which provides a necessary proxy for deriving the energy landscape.
- We assume that invariant terms (e.g., $\|z - z'\|_2^2$) have already saturated, allowing the model to focus purely on the regularization dynamics of non-trivial features. The effect of augmentation is described in Appendix K.

Computational Complexity. A practical limitation arises from the computational cost of the theoretical proxies. While the original SIGReg regularizer is highly efficient at $\mathcal{O}(ndp)$ through stochastic projections, the BHEP and MMD-Gauss formulations used for our analysis require pairwise comparisons, resulting in $\mathcal{O}(n^2 d)$ complexity. Furthermore, evaluating the MMD-Gauss integral via Gauss-Laguerre quadrature introduces a knot-dependent factor k, leading to a total complexity of $\mathcal{O}(n^2 dk)$. This quadratic scaling makes real-time tracking of these dynamics computationally prohibitive for large-scale architectures and massive datasets beyond the ResNet-18/CIFAR-10 setting used in this work.

Structural Gap and Feature Entanglement. Lastly, our proxy-based approach introduces minor structural discrepancies. The fundamental difference between BHEP and SIGReg prevents the derivation of certain closed-form statistics. Moreover, unlike previous work on Barlow Twins, the inherent entanglement of features during the density-matching process in LeJEPA makes it difficult to extract clean, per-feature saturation times, as features evolve through complex global interaction potentials.

Appendix N. Experimental details

To ensure the full reproducibility of our empirical findings, we provide a comprehensive description of our experimental setups, dataset generation procedures, and optimization details.

N.1. Toy model setting

We design a synthetic toy environment to empirically validate the theoretical learning dynamics, particularly the stepwise feature acquisition and the initial explosion phase.

Dataset Generation and Dimensionality. We synthesize a dataset $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$ comprising $n = 1000$ samples. The input dimension is set to $m = 100$, and the target embedding dimension is $d = 32$. To encourage the model to learn features sequentially based on their magnitude, we divide the m-dimensional input space into 50 distinct blocks. We assign a specific variance σ_k^2 to

each block k, which linearly increases from 1.0 to 4.0 (i.e., $\sigma_k^2 \in [1.0, 4.0]$ for $k = 1, \ldots, 50$). This variance scaling explicitly establishes a hierarchy of principal components. For each base signal x, we generate two augmented positive views, x_1 and x_2, by injecting independent Gaussian noise. The noise-injected base signal: $x_v = x + \epsilon$, where $\epsilon_v \sim \mathcal{N}(0, (0.01)^2 I)$, enforcing a noise level of 0.01 regardless of the signal variance.

Network Architecture and Initialization. To strictly isolate the effect of the regularizers and align with our theoretical linear framework, we employ a single bias-free linear layer. The model is parameterized by a weight matrix $W \in \mathbb{R}^{m \times d}$ (where $m = 100, d = 32$), computing the embedding as $z = W^\top x$. Crucially, to clearly observe the initial Explosion Phase described in our theoretical analysis, the weight matrix is initialized with near-zero values. Specifically, each element W_{ij} is drawn from an isotropic Gaussian distribution $\mathcal{N}(0, \sigma_{\text{init}}^2)$ with a very small standard deviation $\sigma_{\text{init}} = 0.001$.

Optimization Scheme. The network is optimized using standard Stochastic Gradient Descent (SGD) without momentum. We train the model for $1,000$ iterations using a batch size of $1,000$ (whole batch). The learning rate η is carefully tuned for each regularizer to match learning dynamics and convergence: we set $\eta = 1.2$ for SIGReg and MMD-Gauss (10 knots), and $\eta = 0.8$ for the BHEP proxy.

Loss Functions and Regularizers. The overall objective function is formulated as a weighted sum of the mean squared error (MSE) invariance loss and a regularizer L_{reg}:

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \left\| z_1^{(i)} - z_2^{(i)} \right\|_2^2 + \lambda \cdot \mathcal{L}_{\text{reg}}(Z) \tag{80}$$

where n is the batch size, and λ controls the regularization strength. We set the base $\lambda = 1.0$. However, to ensure fair comparison across different regularization scales, we apply a scaling factor of $\sqrt{2\pi} \approx 2.5$ to the regularization coefficient for BHEP and MMD-Gauss, matching the theoretical magnitude of SIGReg (detailed derivations are provided in Appendix C).

The specific implementations of the regularizers $\mathcal{L}_{\text{reg}}$ are as follows:

- **SIGReg (LeJEPA):** same as the original paper, we used 256 projections and 17 knots for integration.
- **BHEP (Theoretical Proxy):** We used $\beta = 1/\sqrt{d}$.
- **MMD-Gauss:** We applied the Gauss-Laguerre quadrature method, utilizing $K = 10$ nodes.

Compute resources

Single GPU (NVIDIA T4, 16GB), approximately 1 minute total across three regularizers.

N.2. Practical setting with CIFAR-10

Overall settings are identical with the SOTA setting for LeJEPA, provided on EB-JEPA library [14].

The specific implementations of the regularizers $\mathcal{L}_{\text{reg}}$ are as follows:

- **SIGReg (LeJEPA):** To ensure the accuracy over high dimensionality, we used **4096** projections and 17 knots for integration.
- **BHEP (Theoretical Proxy):** We used $\beta = 1/\sqrt{d}$ and $\lambda = 25.066$ to match coefficient (the coefficient for SIGReg was $\lambda_{\text{SIGReg}} = 10$).

- **MMD-Gauss:** To ensure the accuracy over high dimensionality, we applied the Gauss-Laguerre quadrature method, utilizing $K = 100$ nodes. Also used $\lambda = 25.066$ to match coefficient.

N.2.1. INFINITESIMAL SETTINGS FOR EARLY DYNAMICS OBSERVATION

To observe the starting dynamics (which is less than 1 epoch in normal setting), we modified some of the files to observe the starting dynamics of the regularizers. The specific implementations of the files are as follows:

- **Learning Rate Scaling:** To effectively simulate continuous-time dynamics and slow down the rapid temporal evolution, we significantly reduce the base learning rate by a factor of 100 (from the default $\eta = 0.3$ to $\eta = 0.003$). This scaling is uniformly applied across all tested regularizers.
- **Projector Modification (Removal of BatchNorm):** We remove the final Batch Normalization layer from the projector network. Since BatchNorm artificially rescales the embeddings, removing it is crucial to observe the natural, unconstrained growth of the embedding variance driven solely by the gradient of the regularizers.
- **Near-Zero Weight Initialization:** To capture the pure initial state before any dominant principal components are learned, we scale down the initial magnitude of the model weights. Specifically, the standard deviation for the weight initialization distribution is reduced from the default 0.02 to 0.005 using init_module_weights function provided in nn_utils.py.

Compute resources

Single GPU (NVIDIA A40, 48GB), approximately 1 hour and 34 minutes per training run. We conducted experiments across 3 regularizer settings (SIGReg, MMD-Gauss, BHEP) and 2 initialization settings (normal and small init), total of 6 runs and approximately 9.5 GPU-hours.

Appendix O. Some Notations

$d \in \mathbb{N}$	(embedding dimension)
$n \in \mathbb{N}$	(batch size / sample size)
$z_i \in \mathbb{R}^d$	(i-th embedding)
$x_i \in \mathbb{R}^m$	(i-th input)
λ_k	(k-th eigenvalue of embedding covariance C_Z)
σ_k^2	(k-th eigenvalue of input covariance C_X)
s_k	(learnable weight scaling factor)
$W \in \mathbb{R}^{m \times d}$	(weight matrix)
$C_Z \in \mathbb{R}^{d \times d}$	(embedding covariance matrix)
α, β	(global engine / brake forces)
$T = \alpha/\beta$	(dynamic target eigenvalue)
P_1, P_2	(Push / Pull interaction terms)
$\bar{\lambda}$	(mean eigenvalue $= \frac{1}{d}\sum_{i=1}^{d} \lambda_i$)
$\gamma = \frac{2}{\sqrt{e}}$	(scaling coefficient)
$c_1 = \frac{2}{d},\ c_2 = \frac{1}{d+1}$	(structural constants)

O.1. Parameters in the loss definitions

This subsection clarifies symbols that appear inside the loss formulations introduced in Section 2 and Appendix J (Epps-Pulley, MMD-Gauss, BHEP) and their counterparts in the dynamics analysis. Several symbols are reused with different meanings, so we group them here for reference.

Epps-Pulley test (Definition 1).

$t \in \mathbb{R}$	(frequency variable of the characteristic function)
$\hat{\phi}_X(t)$	(empirical characteristic function of the projected embedding)
$\phi(t) = e^{-t^2/2}$	(target Gaussian characteristic function)
$w(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$	(Gaussian window weighting the integral,coefficient is not used in implementation)

Note that t here is the integration variable inside the EP test and is unrelated to the Gamma-integral variable t used later in the MMD-Gauss formulation (Appendix J).

MMD-Gauss loss (Appendix J).

$\gamma = \frac{1}{2}$	(kernel exponent in MMD-Gauss; *not* the same as $\gamma = 2/\sqrt{e}$ in BHEP)
$\sigma^2 = 1$	(kernel variance of MMD-Gauss; *not* the input eigenvalue σ_k^2)
$s = \frac{2}{2d-3}$	(scaling factor inside the kernel argument; App. J.1)
$t \in [0, \infty)$	(Gamma-integral variable; App. J.1)
$c_1(t) = 4ts$	(t-dependent counterpart of c_1; App. J.1)
$c_2(t) = ts$	(t-dependent counterpart of c_2; App. J.1)
$P_1(t), P_2(t)$	(t-dependent counterparts of the Push / Pull terms)
α_t, β_t	(t-dependent counterparts of the engine / brake; App. J.2)
$u_{t,k}$	(t-dependent counterpart of the local denominator u_k)
T_k	(per-feature target eigenvalue in MMD-Gauss; App. J.3)

Each scalar quantity in the BHEP analysis (top-level table) has a structurally identical t-dependent counterpart here. The only difference is that BHEP gives them as scalars directly, whereas MMD-Gauss expresses them as integrands weighted by $t^{1/2}e^{-t}/\sqrt{\pi}$.

BHEP loss (Definition 2).

$\beta^2 = 1/d$ (BHEP *kernel bandwidth*; a fixed hyperparameter, *not* the brake force β in Theorem 3)

The reuse of β for two unrelated quantities (kernel bandwidth in Definition 2 vs. brake force in Theorem 3) is the most common point of confusion. Throughout Sections 3 onward, β refers exclusively to the brake force; the bandwidth appears only inside the original BHEP definition and Appendix A.

Appendix P. LLM usage

Large language models were used to assist in drafting and revising portions of the manuscript for clarity and readability. All scientific arguments, experimental design, analysis, and conclusions were developed, verified, and approved by the authors.