SELF-SUPERVISED COLLAPSE PREVENTION WITH SCORE-BASED ENTROPY MAXIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have demonstrated the effectiveness of score-based objectives for generative modeling, yet their potential for self-supervised representation learning remains underexplored. We introduce Entropy-Based Regularization (EBReg), a framework that uses latent score estimation to regularize the latent distribution during self-supervised training. EBReg jointly learns an encoder and a latent score model, and utilizes the learned score to provide a diversity-preserving signal that prevents representational collapse. Motivated by a mutual-information lower bound, we formulate the objective as view alignment with smoothed marginal entropy maximization and optimize the marginal entropy without explicit density evaluation. The same score model can additionally be used at test time to refine corrupted representations toward the training latent distribution. Empirically, EBReg learns competitive representations on ImageNet-1K, achieving 75.57% linear-probe accuracy with ViT-L. Across diverse architectures, score-guided test-time adaptation further improves robustness on ImageNet-C, yielding an average relative accuracy gain of approximately 5%.

1 INTRODUCTION

Diffusion models (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020) have established score estimation as a central component of modern generative modeling. However, representations learned through standard diffusion objectives often remain weaker than those produced by dedicated self-supervised learning (SSL) methods on recognition benchmarks (Yu et al., 2025; Chen et al., 2025). Conversely, recent work shows that strong pretrained representations obtained from self-supervised learning can substantially improve diffusion training and generation, as demonstrated by REPA (Yu et al., 2025) and Representation Autoencoders (Zheng et al., 2026). Earlier work on denoising autoencoders also demonstrated that denoising objectives can improve representation learning (Vincent et al., 2008). Together, these observations motivate the question: Can a score estimator and a self-supervised encoder be learned jointly in an end-to-end manner?

A central challenge in SSL is to enforce invariance across augmented views while maintaining sufficient diversity in the latent distribution to avoid collapse. Existing methods achieve this through different mechanisms, including negative-pair interactions (Chen et al., 2020), moment-based constraints (Bardes et al., 2022), and regularization toward prescribed distributions (Balestriero & LeCun, 2025). Despite their different forms, these mechanisms preserve representation diversity while enforcing invariance. We investigate whether an analogous anti-collapse signal can be obtained from a latent score estimated via score matching.

We propose Entropy-Based Regularization (EBReg), a self-supervised framework that jointly trains an encoder and a latent score model. EBReg uses the score of the evolving latent distribution as an adaptive entropy-promoting signal. Figure 1 illustrates this mechanism directly: samples concentrated along a low-dimensional structure are driven by the negative score toward a higher-entropy configuration that occupies both dimensions. Motivated by a variational mutual-information lower bound (Barber & Agakov, 2003b), the objective combines view alignment with maximizing the entropy of a Gaussian-smoothed latent distribution under a scale constraint. Since the latent density is unknown, EBReg estimates the score of the smoothed distribution via denoising score matching and uses it to optimize a gradient surrogate of marginal entropy without explicit density evaluation.

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.

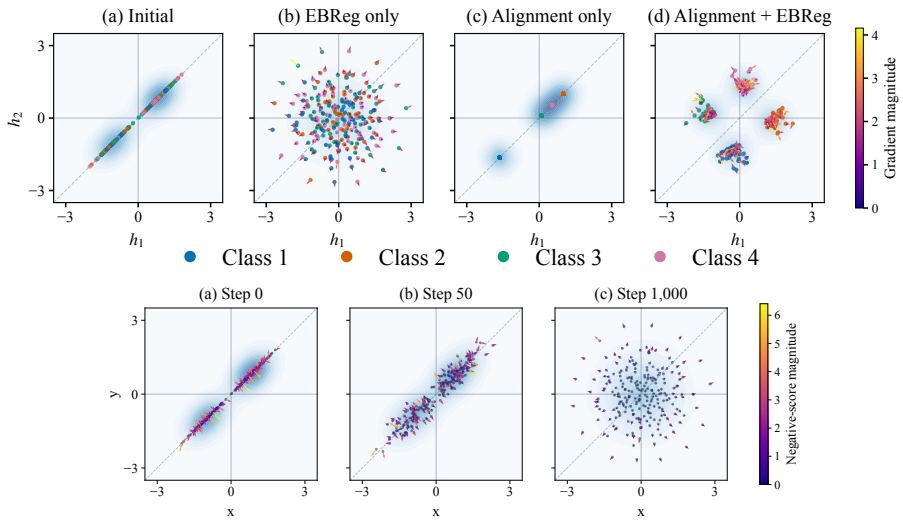


Figure 1: **Illustration of the EBReg anti-collapse mechanism. Top:** (a) Representations lie near a one-dimensional subspace of $\mathbb{R}^2$. (b) EBReg alone spreads them across both dimensions. (c) Alignment alone contracts them, resulting in dimensional collapse. (d) Alignment + EBReg yields compact, separated classes while preventing dimensional collapse. **Bottom:** Evolution of the latent distribution under EBReg. Samples are moved away from the diagonal and populate both dimensions, ultimately forming an isotropic Gaussian distribution.

The resulting encoder update promotes diversity in the latent distribution and counteracts representational collapse.

We further connect the EBReg formulation to existing anti-collapse regularizers (Balestriero & LeCun, 2025; Chen et al., 2020) while replacing explicit negative pairs or fixed distribution matching with an adaptive score field. We characterize how latent scale constraints induce maximum-entropy solutions and relate these solutions to distributional targets and moment conditions underlying existing SSL regularizers. We further analyze the exact-score dynamics near dimensional collapse, including the behavior of low-variance directions. The learned score additionally enables test-time latent refinement, moving corrupted representations toward higher-density regions of the latent distribution while adapting only normalization parameters.

Empirically, EBReg learns competitive representations on ImageNet-1K (Deng et al., 2009) across convolutional and transformer backbones, performing comparably to established self-supervised methods including VICReg (Bardes et al., 2022) and LeJEPA (Balestriero & LeCun, 2025) under the same pretraining and evaluation setup. Score-guided test-time refinement further improves robustness on ImageNet-C (Hendrycks & Dietterich, 2019) across diverse architectures.

Our main contributions are summarized as follows:

Method. We derive an alignment–entropy objective from a variational mutual-information lower bound and introduce EBReg, which uses a learned latent score as an anti-collapse regularizer by optimizing representation entropy without explicit density evaluation or negative pairs.

Theoretical analysis. We analyze the exact-score dynamics near dimensional collapse and characterize when EBReg amplifies low-variance directions. We further show how latent constraints determine maximum-entropy solutions that recover distributional targets and moment conditions underlying existing SSL regularizers.

Score-guided test-time adaptation. We reuse the learned score as a label-free adaptation signal, updating only encoder normalization parameters to guide corrupted representations toward higher-density regions of the source latent distribution and improve robustness on ImageNet-C.

2 Related Work

Latent Space Information Maximization. I-MAX (Becker & Hinton, 1992), an early precursor to joint-embedding methods, maximizes mutual information between module outputs for related inputs. Early tractable formulations use stochastic binary outputs or Gaussian assumptions for scalar continuous outputs. Predictability Maximization (Schmidhuber & Prelinger, 1993) instead combines agreement between paired representations with a discrimination loss that discourages collapse but does not in general yield a mutual-information objective. EBReg revisits the information-maximization principle through an alignment–entropy objective derived from an MI lower bound (Section 3.1). Unlike these earlier formulations, EBReg uses a learned score to optimize marginal entropy without explicit density or mutual-information estimation.

Contrastive methods. Bromley et al. (1993) introduced Siamese networks, which use shared-weight encoders to learn representations by distinguishing matching from non-matching pairs. Building on this principle, modern contrastive methods (Chen et al., 2020; He et al., 2020; Radford et al., 2021) make use of negative examples to provide a non-parametric repulsive signal that prevents representational collapse. Random negative sampling, however, can yield many uninformative pairs in high-dimensional embedding spaces (Wu et al., 2017), resulting in dimensional collapse (Jing et al., 2022). Prototype-based methods such as SwAV, DINO, and MSN (Caron et al., 2020; 2021; Assran et al., 2022) avoid instance-level negatives by balancing assignments over a finite set of learned prototypes, which imposes an explicit discretization of the latent structure. EBReg instead learns a parametric score field to provide entropy-promoting directions without explicit negative comparisons. This design draws on the empirical success of score-based generative models (Song et al., 2021; Rombach et al., 2022) and theoretical evidence that score estimation can adapt to low-dimensional subspace structure, mitigating the curse of high dimensionality (Chen et al., 2023).

Non-contrastive methods. BYOL (Grill et al., 2020) helped revive non-contrastive learning through asymmetric branches and stop-gradient operations. Subsequent teacher–student distillation methods (Assran et al., 2023; Mur-Labadia et al., 2026; Oquab et al., 2024) learn strong representations, with collapse avoidance governed by interactions among architecture, distillation, and optimization. A complementary approach explicitly regularizes representation statistics: VICReg (Bardes et al., 2022) enforces variance floors and penalizes cross-dimensional covariance, while LeJEPA (Balestriero & LeCun, 2025) targets an isotropic Gaussian, with related distributional approaches in VISReg (Wu et al., 2026) and Rectified LpJEPA (Kuang et al., 2026). EBReg instead maximizes smoothed marginal entropy subject to a scale constraint, allowing the entropy objective and constraint to determine the preferred distribution. EbReg under different scale constraints yields maximum-entropy targets that satisfy moment conditions used by existing methods (Table 1).

Reconstruction based methods. Another prominent representation-learning paradigm uses reconstruction objectives (Hinton & Salakhutdinov, 2006; Vincent et al., 2008), which encourage representations to retain information about the input. More recently, masked autoencoders have achieved strong performance by reconstructing missing input regions (He et al., 2022; Bao et al., 2022). Reconstruction naturally discourages collapse and can be interpreted as promoting high mutual information between inputs and their representations. However, input-space objectives may prioritize low-level reconstruction over downstream semantics (Balestriero & Lecun, 2024). Chen et al. (2025) connect denoising diffusion models to denoising autoencoders and propose ℓ-DAE, which corrupts low-dimensional latent features but reconstructs the original input in pixel space using a ViT-based decoder. In contrast, EBReg operates entirely in representation space, using a score function to regularize latent representations directly. This avoids pixel-level reconstruction and decouples collapse prevention from input-space fidelity.

3 Methodology

Our goal is to show that score matching itself can provide a viable objective for learning strong self-supervised representations. To this end, we first derive a mutual-information lower bound and introduce the score-based EBReg surrogate for optimizing marginal entropy (Section 3.1). We then describe alternating score fitting, encoder learning, and score-guided test-time adaptation (Section 3.2), with an overview in Figure 2 and training pseudocode in Appendix C. Finally, we theoretically characterize the exact-score dynamics in collapsed and low-variance directions (Section 3.4).

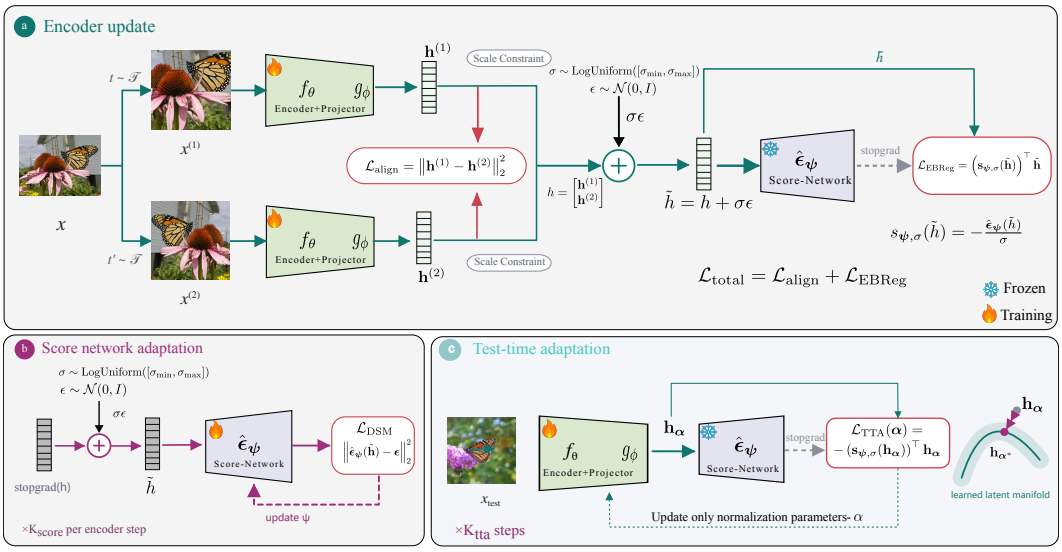


Figure 2: **Overview of EBReg: training and test-time adaptation. (a)** Two augmented views are encoded into normalized embeddings. The encoder minimizes a view-alignment loss and an entropy regularizer, $\mathcal{L}_{\mathrm{EBReg}}$, using a detached estimate of the score of the noise-smoothed embedding distribution. **(b)**. The latent score is parameterized by a noise-prediction network fitted on detached embeddings, with K_{score} updates per encoder step; it predicts the injected Gaussian noise, giving the score estimate $\boldsymbol{s}_{\boldsymbol{\psi},\sigma}(\tilde{\boldsymbol{h}}) = -\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\psi}}(\tilde{\boldsymbol{h}})/\sigma$. **(c)** At test time, the learned score is reused as a test-time adaptation signal. The score model and remaining encoder–projector parameters are frozen, while only normalization parameters $\boldsymbol{\alpha}$ are updated by minimizing $\mathcal{L}_{\mathrm{TTA}}$, guiding corrupted embeddings toward higher-density regions of the learned source distribution.

3.1 From Mutual Information to Score-Based Entropy Regularization

EBReg combines alignment between augmented views with maximization of smoothed marginal entropy. We motivate this objective through the mutual information between representations of two augmented views. Maximizing mutual information favors representations that remain predictive across views while retaining variability in their marginal distribution. This leads naturally to an alignment–entropy formulation, where alignment promotes cross-view agreement and marginal entropy preserves representation diversity and prevents collapse. EBReg then uses a learned score to optimize the smoothed marginal entropy without explicit density evaluation.

An alignment–entropy lower bound. Given an input $\mathbf{x}$, we draw two transformations t, t' independently from an augmentation distribution $\mathcal{T}$. A shared encoder $f_{\boldsymbol{\theta}}$ and projector $g_{\boldsymbol{\phi}}$ produce embeddings $\mathbf{h}^{(1)} = g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(t(\mathbf{x})))$ and $\mathbf{h}^{(2)} = g_{\boldsymbol{\phi}}(f_{\boldsymbol{\theta}}(t'(\mathbf{x})))$ in $\mathbb{R}^d$. Both views share the marginal distribution $p_{\boldsymbol{\theta},\boldsymbol{\phi}}$. Embeddings may occupy a lower-dimensional support and lack a density in the full embedding space, making their differential entropy ill-defined. Gaussian smoothing yields a full-dimensional density, allowing us to work with a well-defined smoothed entropy. For a fixed noise scale $\sigma > 0$, we define

$$\tilde{\mathbf{h}}^{(1)} = \mathbf{h}^{(1)} + \sigma\epsilon, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d), \tag{1}$$

where ϵ is independent of both views. We write $H_\sigma(\mathbf{h}) = H(\mathbf{h} + \sigma\epsilon)$, where H denotes differential entropy, and $p_\sigma = p_{\boldsymbol{\theta},\boldsymbol{\phi}} * \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}_d)$ for the smoothed density.

To relate the smoothed representation to the mutual-information objective between the two clean views, we consider $I(\tilde{\mathbf{h}}^{(1)}; \mathbf{h}^{(2)})$. Since $\tilde{\mathbf{h}}^{(1)}$ is obtained from $\mathbf{h}^{(1)}$ by adding independent Gaussian noise, $\mathbf{h}^{(2)} \to \mathbf{h}^{(1)} \to \tilde{\mathbf{h}}^{(1)}$ forms a Markov chain. The data processing inequality therefore gives $I(\mathbf{h}^{(1)}; \mathbf{h}^{(2)}) \geq I(\tilde{\mathbf{h}}^{(1)}; \mathbf{h}^{(2)})$. Combining this inequality with the variational bound of Barber & Agakov (2003a) applied to the smoothed pair yields

$$I(\mathbf{h}^{(1)}; \mathbf{h}^{(2)}) \geq H_\sigma(\mathbf{h}^{(1)}) + \mathbb{E} \log q(\tilde{\mathbf{h}}^{(1)} \mid \mathbf{h}^{(2)}), \tag{2}$$

for any variational conditional density q. We choose $q_\tau(\cdot \mid \mathbf{h}^{(2)}) = \mathcal{N}(\mathbf{h}^{(2)}, \tau^2 \boldsymbol{I}_d)$ with fixed $\tau > 0$, yielding alignment loss. Choosing a Gaussian variational density does not require either the embedding distribution or the true conditional distribution to be Gaussian. Its cross-entropy is

$$-\mathbb{E} \log q_\tau(\tilde{\mathbf{h}}^{(1)} \mid \mathbf{h}^{(2)}) = \frac{d}{2\tau^2} \mathcal{L}_{\text{align}} + C_{\sigma,\tau}, \tag{3}$$

where $\mathcal{L}_{\text{align}} = d^{-1}\mathbb{E}\|\mathbf{h}^{(1)} - \mathbf{h}^{(2)}\|_2^2$ is the normalized alignment loss (Wang & Isola, 2020) and $C_{\sigma,\tau} = \frac{d}{2}\log(2\pi\tau^2) + \frac{d\sigma^2}{2\tau^2}$ is independent of $(\boldsymbol{\theta}, \boldsymbol{\phi})$. See Appendix A for the full derivation. Substituting this cross-entropy expression into Equation 2 yields the alignment–entropy lower bound

$$\boxed{I(\mathbf{h}^{(1)}; \mathbf{h}^{(2)}) \geq H_\sigma(\mathbf{h}) - \frac{d}{2\tau^2} \mathcal{L}_{\text{align}} - C_{\sigma,\tau}.} \tag{4}$$

Here, $\mathbf{h}$ denotes an embedding drawn from the shared marginal distribution of the two views. For fixed σ and τ, maximizing this lower bound is equivalent to minimizing $\mathcal{L}_{\text{align}} - \gamma H_\sigma(\mathbf{h})$, where $\gamma = 2\tau^2/d$. The objective therefore balances agreement between paired views with entropy of their shared marginal distribution. Increasing latent scale can increase entropy without enriching the representation. We therefore constrain latent scale to prevent entropy maximization through rescaling alone, with the choice of constraint discussed in Section 3.1.

We now introduce a tractable gradient surrogate for optimizing the lower bound in Equation 4. The alignment term is computed directly from paired embeddings, but the smoothed entropy depends on the unknown density p_σ, which is generally intractable to evaluate. Rather than evaluating p_σ directly, EBReg estimates its score, which points toward locally higher-density regions. We therefore replace the negative entropy term with the EBReg gradient surrogate and train the encoder on

$$\boxed{\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{align}} + \gamma \underbrace{\mathbb{E}\left[\text{stopgrad}\left(\boldsymbol{s}_{\boldsymbol{\psi},\sigma}(\tilde{\mathbf{h}})\right)^\top \tilde{\mathbf{h}}\right]}_{\mathcal{L}_{\text{EBReg}}}.} \tag{5}$$

Here, $\tilde{\mathbf{h}} = \mathbf{h} + \sigma\epsilon$ and $\boldsymbol{s}_{\boldsymbol{\psi},\sigma}$ denotes the score estimate. The stop-gradient treats the score as fixed when differentiating the surrogate with respect to the encoder and projector. Section 3.2 describes the alternating score-fitting and encoder updates.

We now show that, with an exact score, this surrogate implements entropy maximization at the gradient level. The entropy-gradient identity for implicit distributions (Li & Turner, 2018; Shi et al., 2018) gives, under standard regularity conditions,

$$\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} H_\sigma(\mathbf{h}) = -\mathbb{E}\left[\boldsymbol{J}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{h})^\top \boldsymbol{s}_\sigma(\tilde{\mathbf{h}})\right], \tag{6}$$

where $\boldsymbol{J}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{h}) = \partial\mathbf{h}/\partial(\boldsymbol{\theta}, \boldsymbol{\phi})$ is the embedding Jacobian. Detaching the entire score vector in Equation 5 gives $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}_{\text{EBReg}} = \mathbb{E}[\boldsymbol{J}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{h})^\top \boldsymbol{s}_{\boldsymbol{\psi},\sigma}(\tilde{\mathbf{h}})]$. Thus, with an exact score, the EBReg update implements descent on $-H_\sigma$ without evaluating the density or entropy. For $\gamma = 2\tau^2/d$, the resulting representation-parameter gradient equals that of the negative MI lower bound up to the positive factor γ. With a learned score, this equivalence is approximate and the surrogate's value is neither an entropy estimate nor an MI bound.

Intuitively, alignment pulls paired embeddings together, while EBReg encourages the embedding distribution to spread out. The learned score $\boldsymbol{s}_{\boldsymbol{\psi},\sigma}(\tilde{\mathbf{h}})$ approximates the local direction of increasing density in the noise-smoothed embedding distribution. Minimizing the surrogate encourages embeddings to move against the score, toward locally lower-density regions, thereby producing an entropy-promoting encoder update.

Connections to existing regularizers. EBReg's scale constraint determines its entropy-maximizing target. Under zero mean and unit marginal variances, the maximizer is the isotropic Gaussian targeted by SIGReg (Balestriero & LeCun, 2025), whose moments coincide with the variance–covariance conditions encouraged by VICReg (Bardes et al., 2022). More generally, different scale constraints induce different maximum-entropy targets while leaving the EBReg formulation unchanged. Table 1 relates these targets to existing SSL regularizers. However, sharing the same target does not imply equivalence of the corresponding objectives. Appendix E provides the derivations.

Table 1: **Maximum-entropy targets under different latent constraints.** Each constraint induces a corresponding maximum-entropy target linked to existing SSL regularizers. First three targets maximize the smoothed entropy for any $\sigma > 0$. The last holds as $\sigma \to 0$ (Appendix E).

Latent constraint	Typical realization	Max.-entropy target	SSL connection
$\mathbb{E}[\mathbf{h}] = \mathbf{0}$, $\operatorname{tr}\operatorname{Cov}(\mathbf{h}) = d$	Batch normalization without affine parameters	Isotropic Gaussian	SIGReg: Gaussian target; VICReg: corresponding first- and second-order moments
$\|\mathbf{h}\|_2 = r$	ℓ_2, RMS, or layer normalization	Uniform on the sphere	Contrastive learning
$\mathbf{h} \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K\}$	Discrete prototype / code assignment	Uniform code usage	SwAV and MSN: balanced prototype usage;
$\mathbb{E}\|\mathbf{h}\|_p^p = M,\ \mathbf{h} \in S$	p-th moment scaling with support constraint	Generalized Gaussian (truncated to S)	Rectified LpJEPA

3.2 Alternating score fitting and encoder learning

Since the score in Equation 5 depends on the encoder's evolving embedding distribution, EBReg alternates between fitting the score model to the current embeddings and updating the encoder with the total objective. Figure 2 summarizes these two phases. Algorithm 1 in Appendix C provides the complete training procedure. Embedding normalization enforces the chosen scale constraint before Gaussian noise is added.

Fitting the latent score. We parameterize a noise-prediction network $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\psi}}$ and recover the latent score from its output via Tweedie's formula. For each minibatch, we pool embeddings from both augmented views and stop gradients through them, $\bar{\mathbf{h}} = \operatorname{stopgrad}(\mathbf{h})$. With the encoder and projector held fixed, we take K_{score} updates of $\boldsymbol{\psi}$ on the denoising objective

$$\mathcal{L}_{\text{DSM}}(\boldsymbol{\psi}) = \mathbb{E}_{\bar{\mathbf{h}},\epsilon}\left[\|\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\psi}}(\bar{\mathbf{h}} + \sigma\epsilon) - \epsilon\|_2^2\right], \qquad \boldsymbol{s}_{\boldsymbol{\psi},\sigma}(\boldsymbol{z}) = -\frac{\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\psi}}(\boldsymbol{z})}{\sigma}, \tag{7}$$

using fresh $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ for each update and the same noise level σ as in Section 3.1. At the unrestricted population optimum, the predictor equals $\mathbb{E}[\epsilon \mid \mathbf{h} + \sigma\epsilon = \boldsymbol{z}]$, whose negative, scaled by $1/\sigma$ is the score of p_σ (Vincent, 2011). Finite network capacity and a finite number of updates yield an approximation to this score as the embedding distribution evolves.

Updating the encoder. We recompute embeddings for the same paired views with gradients enabled, draw fresh noise, and update $(\boldsymbol{\theta}, \boldsymbol{\phi})$ using Equation 5. The alignment loss acts on clean paired embeddings while the entropy regularizer uses the score evaluated at their noisy counterparts. We detach the entire score output so that its value supplies the update direction while gradients flow through $\tilde{\mathbf{h}}$ to the encoder and projector. The score is fixed during the encoder update, so score fitting updates only $\boldsymbol{\psi}$, and the encoder step updates only $(\boldsymbol{\theta}, \boldsymbol{\phi})$.

3.3 Score-guided test-time adaptation

Beyond its role in EBReg training, the learned score provides the direction of increasing log-density under the source latent distribution. This makes it directly useful for test-time adaptation, since corrupted inputs may produce representations that move away from high-density source regions and the score provides a direction back toward them. Let $\boldsymbol{\alpha}$ denote the encoder's normalization parameters and $\mathbf{h}_{\boldsymbol{\alpha}}$ the resulting test representation. We keep the score model, classifier, and remaining encoder parameters fixed, and adapt only $\boldsymbol{\alpha}$, following (Wang et al., 2021).

We define the score-guided adaptation objective as

$$\mathcal{L}_{\text{TTA}}(\boldsymbol{\alpha}) = -\mathbb{E}\left[\operatorname{stopgrad}\left(\boldsymbol{s}_{\boldsymbol{\psi},\sigma}(\mathbf{h}_{\boldsymbol{\alpha}})\right)^\top \mathbf{h}_{\boldsymbol{\alpha}}\right]. \tag{8}$$

The normalization parameters are updated for K_{tta} steps using gradients of $\mathcal{L}_{\text{TTA}}$, while the score model remains fixed. This objective reverses the role of the score used during EBReg training. During pretraining, the negative-score contribution promotes entropy by spreading representations

away from high-density regions. At test time, the positive-score direction instead encourages corrupted representations toward higher-density regions of the source latent distribution. Thus, the score learned for representation regularization also provides a label-free signal for adaptation under distribution shift. Section 4.3 evaluates this procedure under input corruptions.

3.4 Collapse Dynamics and Connection to SSL Regularization

We analyze EBReg near representational collapse and characterize how latent scale constraints determine maximum-entropy solutions related to existing SSL regularizers.

Instability under dimensional collapse. If the embeddings lie in a lower-dimensional subspace of $\mathbb{R}^d$, evaluating the smoothed score on the clean embeddings yields no component along collapsed directions and therefore cannot recover them. Evaluating the score at noise-perturbed embeddings introduces a nonzero component in these directions. The following theorem formalizes this behavior and shows that directions with small but nonzero variance are amplified.

Intuition. Part (a) explains why the score in Equation 5 is evaluated at $\tilde{\mathbf{h}}$ rather than at $\mathbf{h}$. At a clean embedding, the score vanishes along any collapsed direction, regardless of the distribution within the occupied subspace. It also shows that exact collapse is a stationary point of the EBReg objective, with nonzero sample-wise updates along collapsed directions but zero mean drift. Part (b) shows that this stationary point is unstable, since the right-hand side of Equation 10 defines a variance-ascent direction and descending $\mathcal{L}_{\text{total}}$ therefore amplifies any nonzero variance along $\boldsymbol{q}$. Because the update scales with the variance already present rather than introducing a fixed increment, the relative amplification in the low-variance regime is governed by γ/σ^2 (Appendix B). It is also not opposed by the alignment term, which penalizes disagreement between views while remaining invariant to variation shared across paired views. This theorem characterizes directional gradient contributions under an exact score and, in part (b), an independent Gaussian-coordinate assumption; it does not guarantee recovery from exact collapse or variance growth under the full training dynamics.

Theorem 1 (Instability of collapsed directions under EBReg)**.** *Let $\mathbf{h} \in \mathbb{R}^d$ have finite second moment and fix a unit direction $\boldsymbol{q}$. Assume an exact score $\boldsymbol{s}_\sigma$ and the Gaussian perturbation above, with noise independent of the input and augmentation, and write $\mathbf{a}_{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{h})^\top \boldsymbol{q}$ for the parameter gradient of $\boldsymbol{q}^\top \mathbf{h}$, with $\mathbb{E}\|\mathbf{a}_{\boldsymbol{q}}\|_2^2 < \infty$.*

(a) The clean score is blind to dimensional collapse. *If $\mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}) = 0$, then almost surely*

$$\boldsymbol{q}^\top \boldsymbol{s}_\sigma(\tilde{\mathbf{h}}) = -\frac{\boldsymbol{q}^\top \epsilon}{\sigma}, \qquad \boldsymbol{q}^\top \boldsymbol{s}_\sigma(\mathbf{h}) = 0, \tag{9}$$

so the per-sample gradient through $\boldsymbol{q}$, $\mathbf{g}_{\boldsymbol{q}} = -(\gamma/\sigma)(\boldsymbol{q}^\top \epsilon)\, \mathbf{a}_{\boldsymbol{q}}$, is nonzero but has zero mean.

(b) Amplification. *If $\boldsymbol{q}^\top \mathbf{h} = c + \delta \mathrm{u}$ with $\delta > 0$ and $\mathrm{u} \sim \mathcal{N}(0,1)$ independent of $(\boldsymbol{I}_d - \boldsymbol{q}\boldsymbol{q}^\top)\mathbf{h}$, then*

$$\gamma\, \mathbb{E}\big[\mathbf{a}_{\boldsymbol{q}}\, \boldsymbol{q}^\top \boldsymbol{s}_\sigma(\tilde{\mathbf{h}})\big] = -\frac{\gamma}{2(\delta^2 + \sigma^2)}\, \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\, \mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}). \tag{10}$$

4 Experiments

We compare EBReg with existing self-supervised methods, matching parameter count, data augmentations, and training epochs unless otherwise specified. We evaluate representation quality, collapse prevention, hyperparameter sensitivity, computational overhead, and test-time adaptation under input corruptions.

4.1 Linear evaluation

We evaluate the effectiveness of EBReg across multiple architectures and self-supervised objectives on ImageNet-1K (Deng et al., 2009). We consider ResNet-50 (He et al., 2016), ViT-Large, and ViT-Huge (Dosovitskiy et al., 2021) under three established pretraining protocols. For ResNet-50, we adopt the setup from Bardes et al. (2022), including its data augmentations, optimization settings, and training schedule, and replace the original variance–covariance regularization with EBReg. For ViT-Large, we follow the setup from Balestriero & LeCun (2025) and replace SIGReg with EBReg

Table 2: **Linear-probe performance and effective rank on ImageNet-1K.** Results are grouped by pretraining setup. ResNet-50, ViT-L, and ViT-H use the setups from Bardes et al. (2022), Balestriero & LeCun (2025), and Assran et al. (2023), respectively. Within each setup, EBReg and the corresponding baselines use the same backbone, data augmentations, optimization settings, training schedule, and evaluation procedure. Effective rank (Garrido et al., 2023) is computed on backbone representations after pretraining, with higher values indicating less dimensional collapse. Bold indicates higher accuracy within each setup, and underline indicates the second-best.

Backbone	Method / Regularizer	Epochs	Linear probe (%) ↑	Eff. rank ↑
ResNet-50	Alignment only ($\gamma = 0$)	100	Collapse	–
	VICReg	100	**68.02**	1368
	SIGReg	100	66.42	1256
	EBReg (ours)	100	<u>67.92</u>	**1431**
ViT-L	SIGReg	100	<u>75.32</u>	648
	ℓ-DAE	1600	75.00	–
	EBReg (ours)	100	**75.57**	**759**
ViT-H	I-JEPA	100	79.03	540
	I-JEPA + StoP	100	<u>79.33</u>	527
	I-JEPA + StoP + EBReg (ours)	100	**79.41**	**957**

while keeping the remaining pretraining and evaluation protocol unchanged. Finally, we evaluate EBReg with ViT-Huge in the setup from Assran et al. (2023), where EBReg is added to the existing objective rather than replacing it. This setting assesses whether score-based entropy regularization can provide a complementary signal to an existing self-supervised objective. Within each setting, EBReg and the corresponding baseline use the same backbone, augmentations, training schedule, and evaluation procedure. The results are summarized in Table 2.

Table 2 shows that EBReg remains competitive in linear evaluation while consistently producing higher-rank representations across architectures. On ResNet-50, EBReg matches the performance of strong regularization-based baselines while yielding the highest effective rank. On ViT-L, it improves both linear-probe accuracy and representation rank relative to SIGReg.When added to StoP on ViT-H, EBReg yields a modest improvement in linear-probe accuracy while substantially increasing effective rank, suggesting that it complements the existing objective while preserving a richer representation space.

4.2 Hyperparameter Ablations and Computational Cost

Hyperparameter ablations. We ablate hyperparameter under the setup from Bardes et al. (2022) and report ImageNet-1K linear-probe accuracy in Table 3. Among the tested values, the default configuration $\gamma = 0.34$, $\sigma = 4.5$, and $K_{\text{score}} = 16$ yields the highest accuracy. Performance is most sensitive to the noise scale and the number of score updates, while projector dimension has comparatively little effect over the tested range. Increasing score-network width beyond the default also degrades performance, indicating that greater score-model capacity or more frequent score fitting does not necessarily improve representation quality.

Computational cost. Table 4 reports the computational cost of EBReg under the setup from Balestriero & LeCun (2025) using identical hardware. For a backbone with approximately 300M parameters, the score network adds only 5.21M additional parameters, under 2% of the backbone parameter count. The primary overhead arises from repeatedly updating the score model to track the evolving latent distribution. Relative to SIGReg, this increases wall-clock training time by approximately 60 seconds per epoch, from 440 to 500 seconds, corresponding to a 13.6% increase.

The additional computation remains moderate because score fitting is performed entirely in the projected latent space. Its cost is therefore determined primarily by the latent dimension, score-network architecture, and number of score updates, rather than by the image resolution or backbone size. The strong performance obtained with a compact score network may indicate that the latent distribution

admits a sufficiently structured score field that can be approximated with limited capacity, although we do not directly characterize its intrinsic dimensionality.

Table 3: ResNet-50 sensitivity: ImageNet-1K linear-probe accuracy, varying one factor at a time. Defaults are underlined.

Factor	Values	Accuracy (%)		
Regularization weight γ	0.25 / 0.34 / 0.5	64.28	67.92	63.38
Noise level σ	4.25 / 4.5 / 5.0	62.32	67.92	65.43
Score updates K_{score}	4 / 16 / 32	63.72	67.92	58.32
Score-network width	1024 / 2048 / 4096	67.65	67.92	61.35
σ width	0.1 / 0.5 / 0.7	67.51	67.92	67.43
Projector dimension d	64 / 256 / 512	67.42	67.92	67.51

Table 4: ViT-L training costs on identical hardware. Extra parameters are relative to the backbone and projector.

Metric	SIGReg	EBReg
Extra params (M)	0	5.21
Time / epoch (s)	440	500
Peak memory (GB)	90.33	89.84
GPU-hours	195.83	224.01

Table 5: ImageNet-C accuracy (%) before and after five adaptation steps. Seven corruptions with gains for every model are selected by mean gain across models; the final row averages all 15 corruptions. Bold indicates higher accuracy within each model.

	ViT-L Setup from Balestriero & LeCun (2025)		**ViT-H** Setup from Assran et al. (2023)		**ResNet-50** Setup from Bardes et al. (2022)	
Corruption	**Before**	**After**	**Before**	**After**	**Before**	**After**
Fog	44.18	**46.34**	58.41	**61.42**	24.00	**38.60**
Impulse noise	43.41	**46.86**	50.77	**56.33**	15.52	**22.71**
Motion blur	41.12	**46.29**	58.01	**61.52**	18.86	**25.37**
Glass blur	21.72	**25.39**	44.75	**49.93**	7.78	**13.59**
Zoom blur	39.26	**43.00**	50.92	**54.70**	20.19	**27.27**
Gaussian noise	44.56	**48.21**	49.17	**55.97**	25.78	**25.99**
Contrast	57.79	**58.50**	60.44	**63.70**	40.94	**44.81**
Mean (15 corruptions)	46.55	**48.18**	57.79	**61.05**	28.67	**30.50**

4.3 Score-guided test-time adaptation

We evaluate score-guided test-time adaptation on ImageNet-C (Hendrycks & Dietterich, 2019) using the EBReg-trained models and pretraining setups from Section 4.1. In each case, EBReg training yields a corresponding latent score model, which we reuse at test time as the adaptation signal. The score model and classifier remain fixed, while only the encoder's normalization parameters are updated using the objective from Section 3.3. Table 5 reports accuracy before and after five adaptation steps.

Score-guided adaptation improves mean accuracy across all three EBReg-trained models and a broad range of corruptions. These gains indicate that the learned score remains informative under distribution shift, guiding corrupted representations toward higher-density regions of the corresponding source latent distribution. Improvements are particularly pronounced for several noise and blur corruptions, though their magnitude varies across architectures and corruption types.

5 Conclusion

In this paper, we introduce EBReg, a representation learning framework derived from a lower bound on mutual information. EBReg jointly trains the encoder with a lightweight score network and uses the learned score to counter representation collapse. Our analysis shows that dimension-collapsed solutions are unstable under the proposed objective, which induces a persistent force toward activating collapsed dimensions. Under constraints preventing unbounded feature scaling, EBReg recovers target structures related to existing self-supervised regularizers. Empirically, EBReg consistently learns informative, non-collapsed representations across a range of settings. At test time, the learned score guides corrupted representations toward the learned manifold, improving robustness to noise.

Ethics Statement

This work addresses fundamental research questions in self-supervised representation learning and entropy-based regularization. We do not identify ethical concerns specific to the proposed method or the experiments reported in this paper. As with other representation-learning methods, downstream applications should be assessed for potential bias, privacy implications, and misuse in their intended context.

Reproducibility Statement

Section 3 specifies the training objective, alternating optimization procedure, and test-time adaptation rule, with pseudocode in Appendix C. Section 4 describes the datasets, backbone architectures, baseline protocols, and evaluation settings; Tables 3 and 4 report hyperparameter ablations and computational costs. The derivation of the mutual-information bound, analysis of collapsed directions, and maximum-entropy connections are provided in Appendices A, B, and E, respectively.

References

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Michael Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *ECCV*, 2022.

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 *CVPR*, 2023.

Randall Balestriero and Yann Lecun. How learning by reconstruction produces uninformative features for perception. In *ICML*, 2024.

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

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022.

David Barber and Felix Agakov. The IM algorithm: A variational approach to information maximization. In *NeurIPS*, 2003a.

David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, 2003b.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Suzanna Becker and Geoffrey E. Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 1992.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "Siamese" time delay neural network. In *NeurIPS*, 1993.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *ICML*, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. Deconstructing denoising diffusion models for self-supervised learning. In *ICLR*, 2025.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

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.

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

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

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)*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *ICLR*, 2022.

Yilun Kuang, Yash Dagade, Tim G. J. Rudner, Randall Balestriero, and Yann LeCun. Rectified LpJEPA: Joint-embedding predictive architectures with sparse and maximum-entropy representations. *arXiv preprint arXiv:2602.01456*, 2026.

Yingzhen Li and Richard E. Turner. Gradient estimators for implicit models. In *ICLR*, 2018.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mahmoud Assran, Koustuv Sinha, Michael Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In *ICML*, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Jürgen Schmidhuber and Daniel Prelinger. Discovering predictable classifications. *Neural Computation*, 1993.

Jiaxin Shi, Shengyang Sun, and Jun Zhu. A spectral approach to gradient estimation for implicit distributions. In *ICML*, 2018.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 2011.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. In *ICCV*, 2017.

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

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *ICLR*, 2025.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *ICLR*, 2026.

A THE SMOOTHED VARIATIONAL BOUND

For completeness we record the two steps between the data processing inequality and Equation 2; both are the standard variational decomposition of mutual information (Barber & Agakov, 2003a; Poole et al., 2019), applied to the smoothed variable $\tilde{\mathbf{h}}^{(1)} = \mathbf{h}^{(1)} + \sigma\epsilon$. Writing the conditional entropy as a cross-entropy and then inserting $p = (p/q)\, q$,

$$\begin{aligned} I(\tilde{\mathbf{h}}^{(1)}; \mathbf{h}^{(2)}) &= H_\sigma(\mathbf{h}^{(1)}) + \mathbb{E}_{\tilde{\mathbf{h}}^{(1)}, \mathbf{h}^{(2)}} \log p(\tilde{\mathbf{h}}^{(1)} \mid \mathbf{h}^{(2)}) \\ &= H_\sigma(\mathbf{h}^{(1)}) + \mathbb{E}_{\tilde{\mathbf{h}}^{(1)}, \mathbf{h}^{(2)}} \log q(\tilde{\mathbf{h}}^{(1)} \mid \mathbf{h}^{(2)}) + \underbrace{\mathbb{E}_{\mathbf{h}^{(2)}} D_{\mathrm{KL}}\big(p(\cdot \mid \mathbf{h}^{(2)}) \,\|\, q(\cdot \mid \mathbf{h}^{(2)})\big)}_{\geq 0} \\ &\geq H_\sigma(\mathbf{h}^{(1)}) + \mathbb{E}_{\tilde{\mathbf{h}}^{(1)}, \mathbf{h}^{(2)}} \log q(\tilde{\mathbf{h}}^{(1)} \mid \mathbf{h}^{(2)}), \end{aligned} \tag{11}$$

where $q(\cdot \mid \boldsymbol{h}^{(2)})$ is any conditional density. Both differential entropies are finite because $\tilde{\mathbf{h}}^{(1)}$ has a density for every $\sigma > 0$ and the embeddings have finite second moments, so the decomposition is valid even when $\mathbf{h}^{(1)}$ itself is supported on a set of measure zero.

Gaussian variational conditional. For $q_\tau(\cdot \mid \boldsymbol{h}^{(2)}) = \mathcal{N}(\boldsymbol{h}^{(2)}, \tau^2 \boldsymbol{I}_d)$, independence and the moments of the Gaussian noise give

$$\begin{aligned} -\mathbb{E}_{\tilde{\mathbf{h}}^{(1)},\mathbf{h}^{(2)}} \log q_\tau(\tilde{\mathbf{h}}^{(1)} \mid \mathbf{h}^{(2)}) &= \frac{d}{2}\log(2\pi\tau^2) + \frac{1}{2\tau^2}\mathbb{E}\|\mathbf{h}^{(1)} + \sigma\epsilon - \mathbf{h}^{(2)}\|_2^2 \\ &= \frac{d}{2}\log(2\pi\tau^2) + \frac{1}{2\tau^2}\Big[\mathbb{E}\|\mathbf{h}^{(1)} - \mathbf{h}^{(2)}\|_2^2 + \underbrace{2\sigma\,\mathbb{E}\big[(\mathbf{h}^{(1)} - \mathbf{h}^{(2)})^\top \epsilon\big]}_{=0} + \underbrace{\sigma^2\,\mathbb{E}\|\epsilon\|_2^2}_{=\sigma^2 d}\Big] \\ &= \frac{1}{2\tau^2}\mathbb{E}\|\mathbf{h}^{(1)} - \mathbf{h}^{(2)}\|_2^2 + C_{\sigma,\tau}, \end{aligned} \tag{12}$$

where $C_{\sigma,\tau} = \frac{d}{2}\log(2\pi\tau^2) + \frac{d\sigma^2}{2\tau^2}$. The gap between the original MI and the lower bound is the sum of the smoothing loss $I(\mathbf{h}^{(1)};\mathbf{h}^{(2)}) - I(\tilde{\mathbf{h}}^{(1)};\mathbf{h}^{(2)})$ and the conditional KL divergence in Equation 11. The variational step is tight when q equals the true smoothed conditional density almost surely. Under finite second moments, the smoothed MI and the Gaussian conditional KL term are finite, whereas the smoothing loss can be infinite if the original MI is infinite.

B Dimensional collapse

Setup and notation. Let $\mathbf{h} \in \mathbb{R}^d$ be an embedding with mean $\boldsymbol{\mu}$, covariance $\boldsymbol{\Sigma} = \mathrm{Cov}(\mathbf{h})$, and finite second moment. Let $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ be independent of $\mathbf{h}$ and of the input and augmentation that generate it. The noisy embedding is $\tilde{\mathbf{h}} = \mathbf{h} + \sigma\epsilon$; when we want to stress that it is a random variable with a density we write $\mathbf{z}$ for it. Conditionally on $\mathbf{h}$, $\mathbf{z}$ is Gaussian with mean $\mathbf{h}$ and covariance $\sigma^2\boldsymbol{I}_d$, so averaging over $\mathbf{h}$ gives its density

$$p_\sigma(\boldsymbol{z}) = c_\sigma\,\mathbb{E}_{\mathbf{h}}\Big[\exp\Big(-\frac{\|\boldsymbol{z} - \mathbf{h}\|_2^2}{2\sigma^2}\Big)\Big], \qquad c_\sigma = (2\pi\sigma^2)^{-d/2}. \tag{13}$$

This density is smooth and strictly positive for every $\sigma > 0$, even when $\mathbf{h}$ itself is discrete or lies on a lower-dimensional set. The score is $\boldsymbol{s}_\sigma(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log p_\sigma(\boldsymbol{z})$.

B.1 Proof of Theorem 1

EBReg approximates the gradient of $-H_\sigma$. The entropy objective itself disfavors dimensional collapse, since under a fixed variance budget its unique maximizer is full rank (Appendix E, part (i)). That objective-level fact does not describe what the EBReg update does to a collapsed encoder; Theorem 1 does, and we prove it here.

Part (a): the clean score is blind

Goal. Assuming $\mathrm{Var}(\boldsymbol{q}^\top\mathbf{h}) = 0$, show that $\boldsymbol{q}^\top\boldsymbol{s}_\sigma(\mathbf{h}) = 0$ and $\boldsymbol{q}^\top\boldsymbol{s}_\sigma(\tilde{\mathbf{h}}) = -\boldsymbol{q}^\top\epsilon/\sigma$, and that the resulting gradient has zero mean. *Idea.* If the embeddings lie in a subspace, then in the directions orthogonal to it the noisy embedding is pure Gaussian noise, and the score of a Gaussian is known in closed form.

Step 1: the embeddings lie in a subspace. Let $S = \mathrm{range}(\boldsymbol{\Sigma})$ be the subspace the embeddings occupy, $r = \dim S$, $\boldsymbol{P}$ the orthogonal projector onto S, and $\boldsymbol{P}_\perp = \boldsymbol{I}_d - \boldsymbol{P}$ the projector onto $S^\perp$. The collapsed directions are exactly the vectors of $S^\perp$. For any $\boldsymbol{v} \in S^\perp$,

$$\mathrm{Var}(\boldsymbol{v}^\top\mathbf{h}) = \boldsymbol{v}^\top\boldsymbol{\Sigma}\boldsymbol{v} = 0 \quad\Longrightarrow\quad \boldsymbol{v}^\top(\mathbf{h} - \boldsymbol{\mu}) = 0 \text{ almost surely.}$$

Applying this to a basis of $S^\perp$ gives $\boldsymbol{P}_\perp(\mathbf{h} - \boldsymbol{\mu}) = \mathbf{0}$ almost surely: every embedding equals the mean plus a vector in S.

Step 2: split the distance into two orthogonal pieces. Fix a point $\boldsymbol{z}$. Insert $\boldsymbol{\mu}$ and split $\boldsymbol{z} - \boldsymbol{\mu}$ into its components in S and $S^\perp$:

$$\boldsymbol{z} - \mathbf{h} = \underbrace{\big[\boldsymbol{P}(\boldsymbol{z} - \boldsymbol{\mu}) - (\mathbf{h} - \boldsymbol{\mu})\big]}_{\in S \text{ by Step 1}} + \underbrace{\boldsymbol{P}_\perp(\boldsymbol{z} - \boldsymbol{\mu})}_{\in S^\perp}.$$

The two pieces are orthogonal, so their squared norms add:

$$\|\boldsymbol{z}-\mathbf{h}\|_2^2 = \big\|\boldsymbol{P}(\boldsymbol{z}-\boldsymbol{\mu})-(\mathbf{h}-\boldsymbol{\mu})\big\|_2^2 + \big\|\boldsymbol{P}_\perp(\boldsymbol{z}-\boldsymbol{\mu})\big\|_2^2.$$

Only the first term depends on $\mathbf{h}$.

Step 3: the density factorizes. Substitute Step 2 into Equation 13. The exponential of a sum is a product, and the factor that does not depend on $\mathbf{h}$ comes out of the expectation:

$$p_\sigma(\boldsymbol{z}) = p_S\big(\boldsymbol{P}(\boldsymbol{z}-\boldsymbol{\mu})\big) \times \underbrace{(2\pi\sigma^2)^{-(d-r)/2}\exp\Big(-\frac{\|\boldsymbol{P}_\perp(\boldsymbol{z}-\boldsymbol{\mu})\|_2^2}{2\sigma^2}\Big)}_{\text{Gaussian in } S^\perp},$$

$$p_S(\boldsymbol{y}) := (2\pi\sigma^2)^{-r/2}\,\mathbb{E}_{\mathbf{h}}\Big[\exp\Big(-\frac{\|\boldsymbol{y}-(\mathbf{h}-\boldsymbol{\mu})\|_2^2}{2\sigma^2}\Big)\Big].$$

The first factor depends on $\boldsymbol{z}$ only through its component in S, and carries everything about the embedding distribution. The second is a Gaussian density in $S^\perp$ with variance σ^2, and is the same whatever the embeddings do inside S.

Step 4: the score in the collapsed directions. Take the logarithm and differentiate:

$$\boldsymbol{s}_\sigma(\boldsymbol{z}) = \nabla_{\boldsymbol{z}} \log p_S\big(\boldsymbol{P}(\boldsymbol{z}-\boldsymbol{\mu})\big) - \frac{\boldsymbol{P}_\perp(\boldsymbol{z}-\boldsymbol{\mu})}{\sigma^2}.$$

By the chain rule the first term equals $\boldsymbol{P}\nabla \log p_S$, so it lies in S. Applying $\boldsymbol{P}_\perp$ removes it, because $\boldsymbol{P}_\perp\boldsymbol{P}=\mathbf{0}$, and leaves

$$\boldsymbol{P}_\perp \boldsymbol{s}_\sigma(\boldsymbol{z}) = -\frac{\boldsymbol{P}_\perp(\boldsymbol{z}-\boldsymbol{\mu})}{\sigma^2} \qquad \text{for every } \boldsymbol{z}\in\mathbb{R}^d. \tag{14}$$

Step 5: evaluate at clean and noisy embeddings. At a clean embedding, Step 1 gives $\boldsymbol{P}_\perp(\mathbf{h}-\boldsymbol{\mu}) = \mathbf{0}$, so $\boldsymbol{P}_\perp \boldsymbol{s}_\sigma(\mathbf{h}) = \mathbf{0}$. At a noisy embedding,

$$\boldsymbol{P}_\perp(\tilde{\mathbf{h}}-\boldsymbol{\mu}) = \underbrace{\boldsymbol{P}_\perp(\mathbf{h}-\boldsymbol{\mu})}_{=\mathbf{0}} + \sigma\boldsymbol{P}_\perp\epsilon \quad\Longrightarrow\quad \boldsymbol{P}_\perp \boldsymbol{s}_\sigma(\tilde{\mathbf{h}}) = -\frac{\boldsymbol{P}_\perp\epsilon}{\sigma}.$$

For a unit collapsed direction $\boldsymbol{q}\in S^\perp$ we have $\boldsymbol{q}^\top\boldsymbol{P}_\perp = \boldsymbol{q}^\top$, so taking the inner product with $\boldsymbol{q}$ gives Equation 9.

Step 6: the gradient through q has zero mean. The per-sample EBReg loss is $\gamma\,\mathrm{stopgrad}(\boldsymbol{s}_\sigma(\tilde{\mathbf{h}}))^\top\tilde{\mathbf{h}}$. The score is detached and the noise does not depend on the parameters, so by the chain rule its parameter gradient is $\gamma\,\boldsymbol{J}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{h})^\top\boldsymbol{s}_\sigma(\tilde{\mathbf{h}})$. The part of this gradient that passes through direction $\boldsymbol{q}$ is

$$\mathbf{g}_{\boldsymbol{q}} = \gamma\,\underbrace{\boldsymbol{J}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{h})^\top\boldsymbol{q}}_{=\mathbf{a}_{\boldsymbol{q}}}\,\underbrace{\boldsymbol{q}^\top\boldsymbol{s}_\sigma(\tilde{\mathbf{h}})}_{=-\boldsymbol{q}^\top\epsilon/\sigma} = -\frac{\gamma}{\sigma}\,(\boldsymbol{q}^\top\epsilon)\,\mathbf{a}_{\boldsymbol{q}}.$$

The vector $\mathbf{a}_{\boldsymbol{q}}$ depends only on the view, while $\boldsymbol{q}^\top\epsilon\sim\mathcal{N}(0,1)$ is independent of it. Therefore

$$\mathbb{E}[\mathbf{g}_{\boldsymbol{q}}] = -\frac{\gamma}{\sigma}\,\mathbb{E}[\boldsymbol{q}^\top\epsilon]\,\mathbb{E}[\mathbf{a}_{\boldsymbol{q}}] = \mathbf{0}, \qquad \mathbb{E}\|\mathbf{g}_{\boldsymbol{q}}\|_2^2 = \frac{\gamma^2}{\sigma^2}\,\mathbb{E}\big[(\boldsymbol{q}^\top\epsilon)^2\big]\,\mathbb{E}\|\mathbf{a}_{\boldsymbol{q}}\|_2^2 = \frac{\gamma^2}{\sigma^2}\,\mathbb{E}\|\mathbf{a}_{\boldsymbol{q}}\|_2^2.$$

So each sample contributes a nonzero push along $\boldsymbol{q}$, but the pushes average to zero: exact collapse is a stationary point of the EBReg term. □

PART (B): AMPLIFICATION

Goal. Assuming $\boldsymbol{q}^\top\mathbf{h} = c+\delta\mathrm{u}$ with $\delta>0$, where $\mathrm{u}\sim\mathcal{N}(0,1)$ is independent of the remaining coordinates $(\boldsymbol{I}_d-\boldsymbol{q}\boldsymbol{q}^\top)\mathbf{h}$, show that the expected EBReg gradient through $\boldsymbol{q}$ is a negative multiple of $\nabla\,\mathrm{Var}(\boldsymbol{q}^\top\mathbf{h})$. *Idea.* Along $\boldsymbol{q}$ the noisy embedding is an independent one-dimensional Gaussian, whose score is linear; the rest is a covariance computation. Write $\Theta=(\boldsymbol{\theta},\boldsymbol{\phi})$ for the parameters.

Step 1: the noisy coordinate along q is an independent Gaussian. Along $\boldsymbol{q}$,

$$\boldsymbol{q}^\top \mathbf{z} = c + \delta \mathrm{u} + \sigma\, \boldsymbol{q}^\top \epsilon.$$

This is a constant plus two independent Gaussians with variances δ^2 and σ^2, so $\boldsymbol{q}^\top \mathbf{z} \sim \mathcal{N}(c, \delta^2 + \sigma^2)$. It is also independent of the remaining coordinates $(\boldsymbol{I}_d - \boldsymbol{q}\boldsymbol{q}^\top)\mathbf{z} = (\boldsymbol{I}_d - \boldsymbol{q}\boldsymbol{q}^\top)\mathbf{h} + \sigma(\boldsymbol{I}_d - \boldsymbol{q}\boldsymbol{q}^\top)\epsilon$: u is independent of the first term by assumption, and for isotropic Gaussian noise the component $\boldsymbol{q}^\top \epsilon$ is independent of the components orthogonal to $\boldsymbol{q}$.

Step 2: the score along q. Independence means the density is a product of the one-dimensional Gaussian density of $\boldsymbol{q}^\top \boldsymbol{z}$ and a density of the remaining coordinates. Only the first depends on $\boldsymbol{q}^\top \boldsymbol{z}$, so

$$\boldsymbol{q}^\top \boldsymbol{s}_\sigma(\boldsymbol{z}) = \frac{\partial}{\partial(\boldsymbol{q}^\top \boldsymbol{z})}\Big[-\frac{(\boldsymbol{q}^\top \boldsymbol{z} - c)^2}{2(\delta^2 + \sigma^2)}\Big] = -\frac{\boldsymbol{q}^\top \boldsymbol{z} - c}{\delta^2 + \sigma^2}.$$

At $\boldsymbol{z} = \tilde{\mathbf{h}}$ we have $\boldsymbol{q}^\top \tilde{\mathbf{h}} - c = \delta \mathrm{u} + \sigma\, \boldsymbol{q}^\top \epsilon$, hence

$$\boldsymbol{q}^\top \boldsymbol{s}_\sigma(\tilde{\mathbf{h}}) = -\frac{\delta \mathrm{u} + \sigma\, \boldsymbol{q}^\top \epsilon}{\delta^2 + \sigma^2}.$$

Setting $\delta = 0$ recovers part (a).

Step 3: the expected gradient through q. Multiply by $\mathbf{a}_{\boldsymbol{q}}$ and take expectations:

$$\mathbb{E}\big[\mathbf{a}_{\boldsymbol{q}}\, \boldsymbol{q}^\top \boldsymbol{s}_\sigma(\tilde{\mathbf{h}})\big] = -\frac{1}{\delta^2 + \sigma^2}\Big(\delta\, \mathbb{E}[\mathbf{a}_{\boldsymbol{q}} \mathrm{u}] + \sigma \underbrace{\mathbb{E}[\mathbf{a}_{\boldsymbol{q}}]\, \mathbb{E}[\boldsymbol{q}^\top \epsilon]}_{=\mathbf{0}}\Big) = -\frac{\delta\, \mathrm{Cov}(\mathbf{a}_{\boldsymbol{q}}, \mathrm{u})}{\delta^2 + \sigma^2}.$$

The noise term vanishes because ϵ is independent of the view and has zero mean, and $\mathbb{E}[\mathbf{a}_{\boldsymbol{q}} \mathrm{u}] = \mathrm{Cov}(\mathbf{a}_{\boldsymbol{q}}, \mathrm{u})$ because $\mathbb{E}[\mathrm{u}] = 0$. The noise no longer cancels everything: the embedding's own variation $\delta \mathrm{u}$ correlates with how the encoder responds.

Step 4: the gradient of the variance. Differentiate $\mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}_\Theta) = \mathbb{E}\big[(\boldsymbol{q}^\top \mathbf{h}_\Theta - \mathbb{E}[\boldsymbol{q}^\top \mathbf{h}_\Theta])^2\big]$ under the expectation, which $\mathbb{E}\|\mathbf{a}_{\boldsymbol{q}}\|_2^2 < \infty$ and local domination of the parameter derivative permit. Using $\nabla_\Theta(\boldsymbol{q}^\top \mathbf{h}_\Theta) = \mathbf{a}_{\boldsymbol{q}}$,

$$\nabla_\Theta\, \mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}_\Theta) = 2\, \mathbb{E}\big[(\boldsymbol{q}^\top \mathbf{h} - \mathbb{E}[\boldsymbol{q}^\top \mathbf{h}])(\mathbf{a}_{\boldsymbol{q}} - \mathbb{E}[\mathbf{a}_{\boldsymbol{q}}])\big] = 2\mathrm{Cov}(\mathbf{a}_{\boldsymbol{q}}, \boldsymbol{q}^\top \mathbf{h}) = 2\delta \mathrm{Cov}(\mathbf{a}_{\boldsymbol{q}}, \mathrm{u}),$$

where the last step uses $\boldsymbol{q}^\top \mathbf{h} = c + \delta \mathrm{u}$.

Step 5: combine. Steps 3 and 4 contain the same covariance. Eliminating it,

$$\gamma\, \mathbb{E}\big[\mathbf{a}_{\boldsymbol{q}}\, \boldsymbol{q}^\top \boldsymbol{s}_\sigma(\tilde{\mathbf{h}})\big] = -\frac{\gamma\, \delta \mathrm{Cov}(\mathbf{a}_{\boldsymbol{q}}, \mathrm{u})}{\delta^2 + \sigma^2} = -\frac{\gamma}{2(\delta^2 + \sigma^2)}\, \nabla_\Theta\, \mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}),$$

which is Equation 10. □

Consequence: exponential growth. Gradient descent subtracts η times the gradient, so the contribution of this term to the update is $\Delta\Theta = \frac{\eta\gamma}{2(\delta^2+\sigma^2)} \nabla_\Theta\, \mathrm{Var}(\boldsymbol{q}^\top \mathbf{h})$, a step *up* the variance. To first order in η the variance changes by

$$\Delta\, \mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}) = \nabla_\Theta\, \mathrm{Var}^\top\, \Delta\Theta + O(\eta^2) = \frac{\eta\gamma}{2(\delta^2 + \sigma^2)} \big\|\nabla_\Theta\, \mathrm{Var}(\boldsymbol{q}^\top \mathbf{h})\big\|_2^2 + O(\eta^2).$$

Insert $\|\nabla_\Theta\, \mathrm{Var}\,\|_2^2 = 4\delta^2 \|\mathrm{Cov}(\mathbf{a}_{\boldsymbol{q}}, \mathrm{u})\|_2^2$ from Step 4 and $\mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}) = \delta^2$:

$$\Delta\, \mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}) = \kappa\, \mathrm{Var}(\boldsymbol{q}^\top \mathbf{h}) + O(\eta^2), \qquad \kappa = \frac{2\eta\gamma\, \|\mathrm{Cov}(\mathbf{a}_{\boldsymbol{q}}, \mathrm{u})\|_2^2}{\delta^2 + \sigma^2} \xrightarrow[\delta \to 0]{} \frac{2\eta\gamma\, \|\mathrm{Cov}(\mathbf{a}_{\boldsymbol{q}}, \mathrm{u})\|_2^2}{\sigma^2}.$$

The variance therefore grows by a fixed fraction of itself per step, which is exponential growth at a rate proportional to γ/σ^2. The drift is first order in η, does not depend on the batch size, and vanishes as $\delta \to 0$, consistent with part (a). Starting from a seed δ_0, reaching a fixed scale takes on the order of $\log(1/\delta_0)/\kappa$ steps, so the size of the seed matters only logarithmically.

C EBREG TRAINING PROCEDURE

Algorithms 1 and 2 detail the EBReg training procedure and score-guided test-time adaptation described in Sections 3.2 and 3.3, respectively.

Input: normalized encoder–projector $g_{\phi} \circ f_{\theta}$, noise predictor $\hat{\epsilon}_{\psi}$, augmentation distribution $\mathcal{T}$; $B, T, K_{\text{score}}, \sigma, \gamma$.
Notation: $\text{sg} = \text{stopgrad}$. Score-fitting and regularization means pool the $2B$ embeddings; alignment uses the B matched pairs as in Equation 5. Each noise draw is independent and standard Gaussian.

for $r = 1, \ldots, T$ **do**
  Sample a minibatch and paired views $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}$ using $\mathcal{T}$.
  $\bar{\boldsymbol{h}} \leftarrow \text{sg}(g_{\phi}(f_{\theta}(\boldsymbol{x}^{(1:2)})))$.
  repeat K_{score} **times**
    Draw noise $\boldsymbol{\epsilon}$; $\tilde{\boldsymbol{h}} \leftarrow \bar{\boldsymbol{h}} + \sigma\boldsymbol{\epsilon}$.
    Update only ψ on $\text{mean}\, \|\hat{\boldsymbol{\epsilon}}_{\psi}(\tilde{\boldsymbol{h}}) - \boldsymbol{\epsilon}\|_2^2$.
  end repeat
  Recompute $\boldsymbol{h} \leftarrow g_{\phi}(f_{\theta}(\boldsymbol{x}^{(1:2)}))$ on the same views with gradients; draw fresh $\boldsymbol{\epsilon}$.
  $\tilde{\boldsymbol{h}} \leftarrow \boldsymbol{h} + \sigma\boldsymbol{\epsilon}$; $\boldsymbol{s} \leftarrow \text{sg}(-\hat{\boldsymbol{\epsilon}}_{\psi}(\tilde{\boldsymbol{h}})/\sigma)$.
  Update only $\boldsymbol{\theta}, \boldsymbol{\phi}$ on $\mathcal{L}_{\text{align}} + \gamma\, \text{mean}(\boldsymbol{s}^{\top}\tilde{\boldsymbol{h}})$, using Equation 5.
end for; **return** $g_{\phi} \circ f_{\theta}$ and $\boldsymbol{s}_{\psi,\sigma}$.

Algorithm 1: EBReg training

Input: EBReg-trained encoder f_{θ}, fixed projector g_{ϕ}, score model $\boldsymbol{s}_{\psi,\sigma}$, classifier c; K_{tta} and adaptation learning rate η.
Notation: $\boldsymbol{\alpha} \subset \boldsymbol{\theta}$ denotes the encoder's normalization parameters and $\text{sg} = \text{stopgrad}$. The projector, score model, classifier, and all encoder parameters outside $\boldsymbol{\alpha}$ remain fixed.

for each test minibatch $\boldsymbol{x}$ **do**
  Reset $\boldsymbol{\alpha}$ to its pretrained source values.
  repeat K_{tta} **times**
    Compute encoder representation $\boldsymbol{r}_{\boldsymbol{\alpha}} \leftarrow f_{\theta,\boldsymbol{\alpha}}(\boldsymbol{x})$.
    Compute score-space embedding $\boldsymbol{h}_{\boldsymbol{\alpha}} \leftarrow g_{\phi}(\boldsymbol{r}_{\boldsymbol{\alpha}})$.
    $\boldsymbol{s} \leftarrow \text{sg}(\boldsymbol{s}_{\psi,\sigma}(\boldsymbol{h}_{\boldsymbol{\alpha}}))$.
    $\mathcal{L}_{\text{TTA}} \leftarrow -\,\text{mean}(\boldsymbol{s}^{\top}\boldsymbol{h}_{\boldsymbol{\alpha}})$.
    Update only $\boldsymbol{\alpha}$ on $\mathcal{L}_{\text{TTA}}$ with learning rate η.
  end repeat
  Recompute $\boldsymbol{r}_{\boldsymbol{\alpha}} \leftarrow f_{\theta,\boldsymbol{\alpha}}(\boldsymbol{x})$ and predict $\hat{\boldsymbol{y}} \leftarrow c(\boldsymbol{r}_{\boldsymbol{\alpha}})$.
end for

Algorithm 2: Score-guided test-time adaptation

D IMPLEMENTATION DETAILS

In this section, we elaborate on various hyperparameter and design choices for our experiments. We use ImageNet-1k for all our experiments, using the standard train, validation sets comprising 1.28M and 50K images, respectively. We use ResNet-50, ViT-H, and ViT-L as the major architectural choices for our study. The augmentation pipeline matches those used in the corresponding protocols. We use a simple MLP as a denoiser in all the cases. We use a simple MLP projector for both VICREG and LeJEPA cases with projector dims of 256 and 64, respectively. Other hyperparameters for each of the protocols are listed below:

VICReg protocol. The projector has layer widths 8192–8192–256. The denoiser uses a hidden width of 2048 and four blocks. The alignment-loss coefficient is 1.

LeJEPA protocol. The augmentation pipeline produces two global crops of size 224×224 with scale range $[0.3, 1.0]$ and eight local crops of size 98×98 with scale range $[0.05, 0.3]$. Images use ImageNet channel-wise normalization with mean $(0.485, 0.456, 0.406)$ and standard deviation

$(0.229, 0.224, 0.225)$. The learning-rate warmup starts at 0.01 times the configured learning rate. Training uses bfloat16 precision.

I-JEPA/StoP protocol. The input crop size is 224×224, with crop scale range $[0.3, 1.0]$ and patch size 14. We use one context mask with scale range $[0.85, 1.0]$ and four target masks with scale range $[0.15, 0.2]$, mask aspect ratios in $[0.75, 1.5]$, and a minimum of 10 retained patches. The predictor has depth 12 and embedding dimension 384. The initial and final learning rates are 2×10^{-4} and 10^{-6}, respectively; the configuration specifies constant learning-rate and weight-decay strategies from epoch 250. StoP is enabled with variance parameter 0.2.

E MAXIMUM-ENTROPY TARGETS UNDER EACH CONSTRAINT

We compute the maximum-entropy distribution under each constraint of Table 1, labelled (i)–(iv) in the order of the table. All four follow from one standard computation.

The maximum-entropy recipe. Consider distributions on a support $\mathcal{S}$ that satisfy moment constraints $\mathbb{E}[f_k(\mathbf{h})] = c_k$ for $k = 1, \dots, m$. Suppose the exponential-family density

$$p^\star(\boldsymbol{h}) = \frac{1}{Z} \exp\Big(-\sum_{k=1}^{m} \lambda_k f_k(\boldsymbol{h})\Big) \quad \text{on } \mathcal{S} \tag{15}$$

satisfies the constraints for some multipliers λ_k. Then $p^\star$ is the unique maximum-entropy distribution. To see this, take any feasible p. Because $-\log p^\star = \log Z + \sum_k \lambda_k f_k$, its average under p involves only the constrained moments:

$$-\int p \log p^\star = \log Z + \sum_k \lambda_k \, \mathbb{E}_p[f_k(\mathbf{h})] = \log Z + \sum_k \lambda_k c_k = H(p^\star).$$

Hence

$$H(p) = -\int p \log p^\star - D_{\mathrm{KL}}(p \,\|\, p^\star) = H(p^\star) - D_{\mathrm{KL}}(p \,\|\, p^\star) \le H(p^\star),$$

with equality only when $p = p^\star$. With no moment constraints, $m = 0$, the recipe gives the constant density, that is, the uniform distribution on $\mathcal{S}$.

(i) Quadratic constraint: isotropic Gaussian. Let $\mathbb{E}[\mathbf{h}] = \mathbf{0}$ and $\operatorname{tr}\operatorname{Cov}(\mathbf{h}) = d$; unit marginal variances are a special case. The noisy embedding $\mathbf{z} = \mathbf{h} + \sigma\epsilon$ then satisfies

$$\mathbb{E}[\mathbf{z}] = \mathbf{0}, \qquad \mathbb{E}\|\mathbf{z}\|_2^2 = \mathbb{E}\|\mathbf{h}\|_2^2 + \sigma^2 \mathbb{E}\|\epsilon\|_2^2 = d(1 + \sigma^2).$$

Apply the recipe to $\mathbf{z}$ on $\mathcal{S} = \mathbb{R}^d$ with $f(\boldsymbol{z}) = \|\boldsymbol{z}\|_2^2$. It gives $p^\star(\boldsymbol{z}) \propto \exp(-\lambda\|\boldsymbol{z}\|_2^2)$, a zero-mean isotropic Gaussian, and matching the second moment fixes its variance at $1 + \sigma^2$ per coordinate. Its entropy is the bound

$$H_\sigma(\mathbf{h}) = H(\mathbf{z}) \le H\big(\mathcal{N}(\mathbf{0}, (1 + \sigma^2)\boldsymbol{I}_d)\big) = \frac{d}{2} \log\big(2\pi e(1 + \sigma^2)\big), \tag{16}$$

with equality only when $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, (1 + \sigma^2)\boldsymbol{I}_d)$. This happens exactly when $\mathbf{h} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$: adding the independent noise $\sigma\epsilon$ to $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ gives that Gaussian, and conversely the characteristic function of $\mathbf{z}$ is the product $\varphi_{\mathbf{h}}(\boldsymbol{\omega}) \exp(-\frac{1}{2}\sigma^2\|\boldsymbol{\omega}\|_2^2)$, so dividing by the noise factor leaves $\varphi_{\mathbf{h}}(\boldsymbol{\omega}) = \exp(-\frac{1}{2}\|\boldsymbol{\omega}\|_2^2)$. The result holds for every $\sigma > 0$. Since the maximizer has $\operatorname{Cov}(\mathbf{h}) = \boldsymbol{I}_d$, a rank-deficient representation is never entropy-maximal under this constraint. The target $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ is SIGReg's reference (Balestriero & LeCun, 2025), its covariance is the second-moment condition that VICReg's penalties drive toward (Bardes et al., 2022), and batch normalization without affine parameters enforces the constraint up to finite-batch error.

(ii) Spherical constraint: uniform on the sphere. Let $\|\mathbf{h}\|_2 = 1$, so $\mathcal{S} = \mathbb{S}^{d-1}$ and there is no moment constraint. The recipe gives the constant density with respect to surface measure, the uniform distribution, with entropy $\log|\mathbb{S}^{d-1}|$. The same holds on a sphere of any radius, such as $\|\mathbf{h}\|_2 = \sqrt{d}$ under RMS normalization. Layer normalization additionally centers each embedding, $\mathbf{1}^\top \mathbf{h} = 0$, so $\mathcal{S}$ is the sphere inside the hyperplane orthogonal to $\mathbf{1}$ and the target is uniform on that lower-dimensional sphere. Spherical uniformity is the target that contrastive objectives approach through negatives (Wang & Isola, 2020).

(iii) Categorical constraint: uniform code usage. Let $\mathbf{h} \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_K\}$, so $\mathcal{S}$ is a set of K points with counting measure and again no moment constraint. The recipe gives equal probabilities $1/K$, with entropy $\log K$. Uniform usage is SwAV's equipartition constraint and the marginal-entropy regularizer of MSN (Caron et al., 2020; Assran et al., 2022); DINO's centering and sharpening act as a related mechanism rather than a prescribed density (Caron et al., 2021). Marginal balance across inputs differs from uninformative uniform predictions for every input.

(iv) Moment constraint: generalized Gaussian. Let $\mathbb{E}\|\mathbf{h}\|_p^p = M$ on a support S. The recipe with $f(\boldsymbol{h}) = \|\boldsymbol{h}\|_p^p$ gives

$$p^\star(\boldsymbol{h}) = Z^{-1} \exp\bigl(-\alpha \|\boldsymbol{h}\|_p^p\bigr) \mathbb{1}_S(\boldsymbol{h}), \tag{17}$$

with α chosen to satisfy the constraint, whenever this density is normalizable. On $\mathbb{R}^d$ it is a product generalized Gaussian, which reduces to the Gaussian of part (i) at $p = 2$, and restricting to the positive orthant gives its truncation. This is the principle behind Rectified LpJEPA's target design (Kuang et al., 2026), whose additional rectification introduces atoms at zero and requires mixed-measure entropy.

Effect of the smoothing. Part (i) is computed directly for the smoothed entropy H_σ and is exact at every noise level. Parts (ii)–(iv) are computed for the entropy of $\mathbf{h}$ itself, which H_σ approaches from above as $\sigma \to 0$: adding independent noise can only add entropy, $H_\sigma(\mathbf{h}) \geq H(\mathbf{h})$, and the excess vanishes with σ. For the sphere and the codes the targets remain maximizers of H_σ at every $\sigma > 0$ by symmetry. Rotating a distribution on the sphere, or permuting the codes, leaves H_σ unchanged because the noise is isotropic; the uniform distribution is the average of all such rotated or permuted copies; and the entropy of a mixture is at least the average entropy of its components. For the moment constraint with $p \neq 2$ the maximizer of H_σ at fixed $\sigma > 0$ is a smoothed analogue of Equation 17 that we do not characterize.

F Use of Large Language Models

LLMs were used to assist with writing, revising, organizing, and polishing the manuscript; coding tasks; table and bibliography formatting; and LaTeX troubleshooting. They also assisted in deriving proofs presented in the appendix, reviewing mathematical arguments, identifying gaps between stated assumptions and claimed conclusions, and drafting revisions to narrow the interpretation of the directional gradient results. All experimental values used in preparing tables and discussing results were supplied by the authors. The authors take full responsibility for the manuscript's correctness, originality, and proper attribution, including all LLM-assisted text, code, proofs, and mathematical reasoning.