Self-Supervised Collapse Prevention with Score-Based Entropy Maximization*

Adarsh Kappiyath, Silpa Vadakkeeveetil Sreelatha, Xiatian Zhu

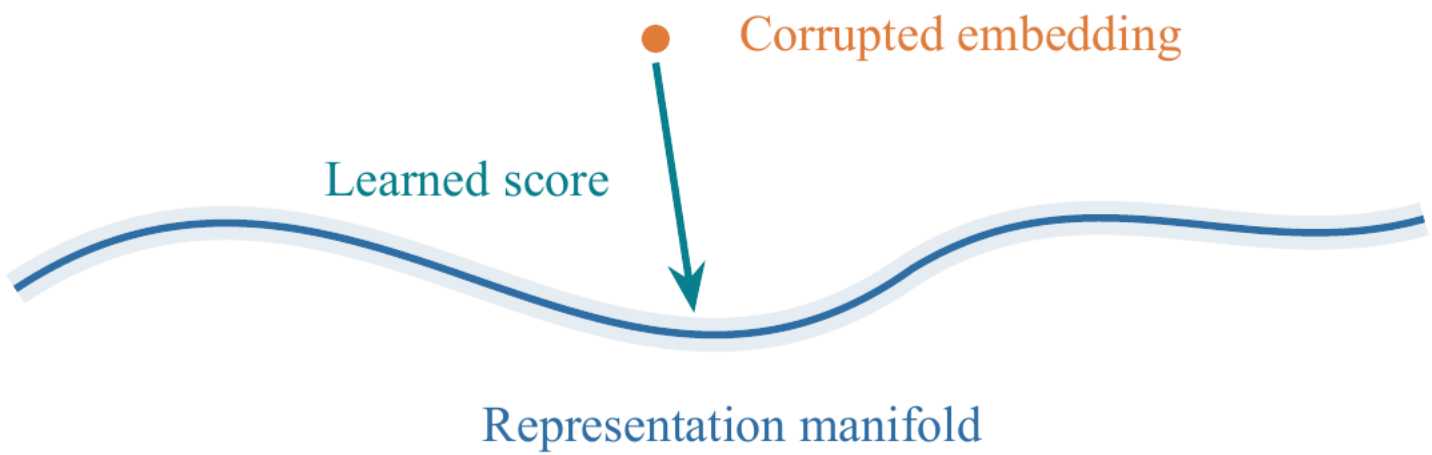
Yann LeCun, Lu Yin, Amir Bar

Under review

October 20, 2026

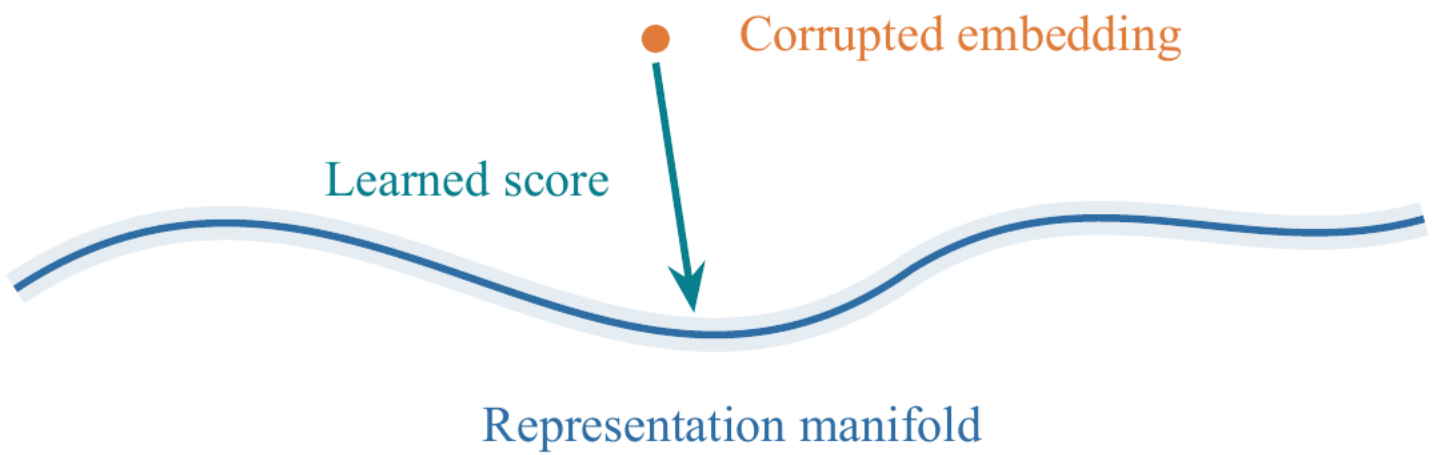
A score captures representation geometry

A learned score captures the local geometry of the representation distribution, pointing toward higher density.



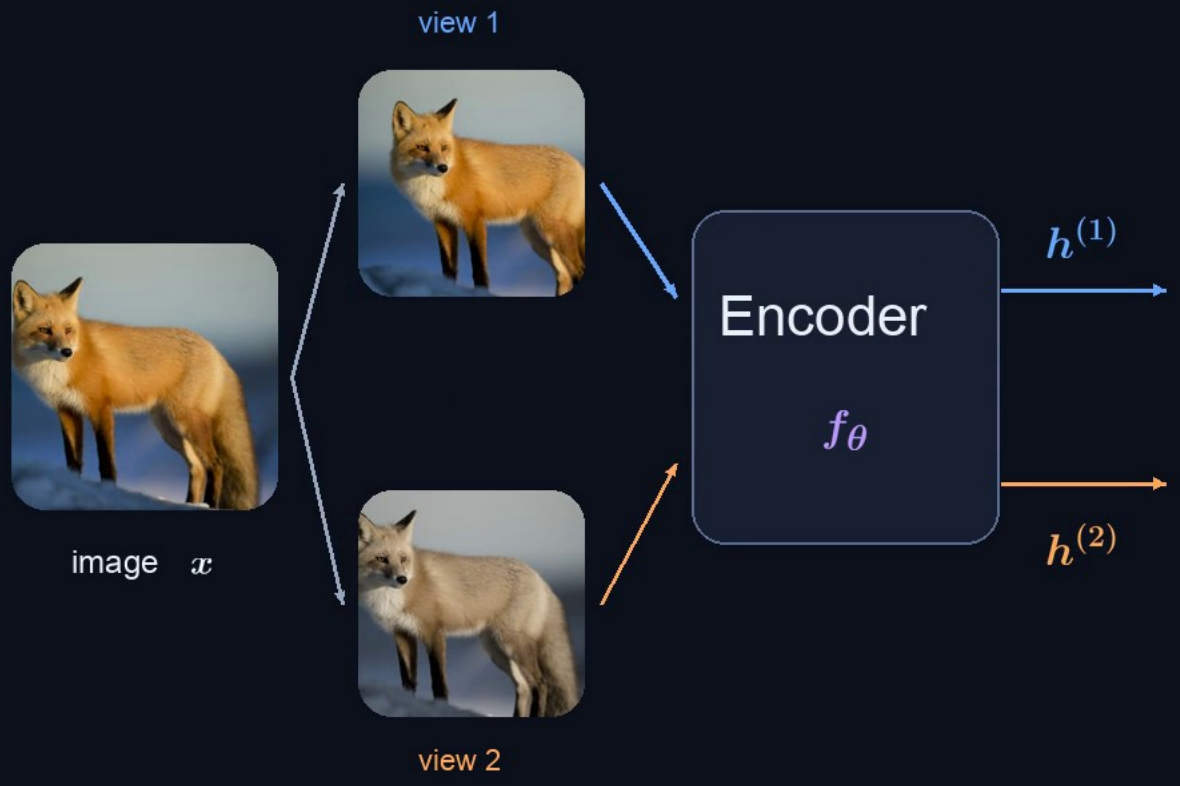
A score captures representation geometry

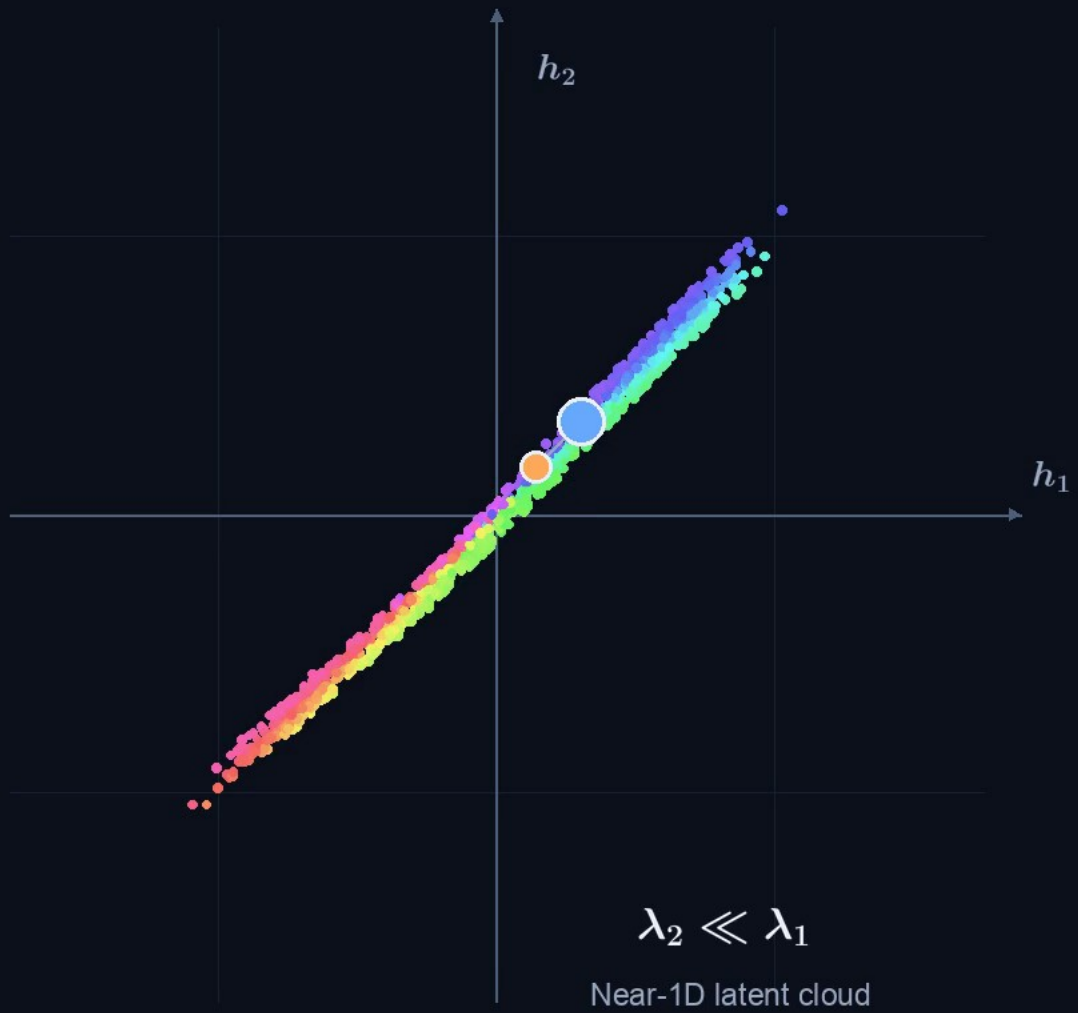
A learned score captures the local geometry of the representation distribution, pointing toward higher density.



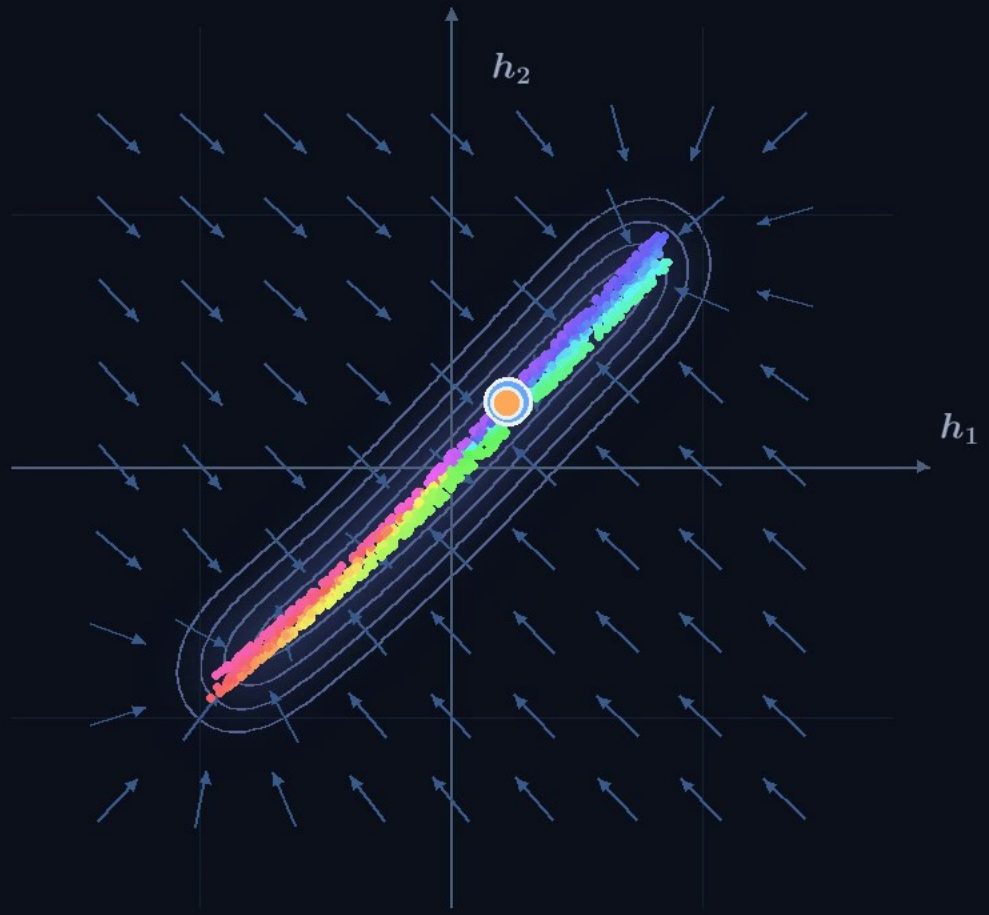
Can we use the learned score for representation learning?

Two views. One encoder.

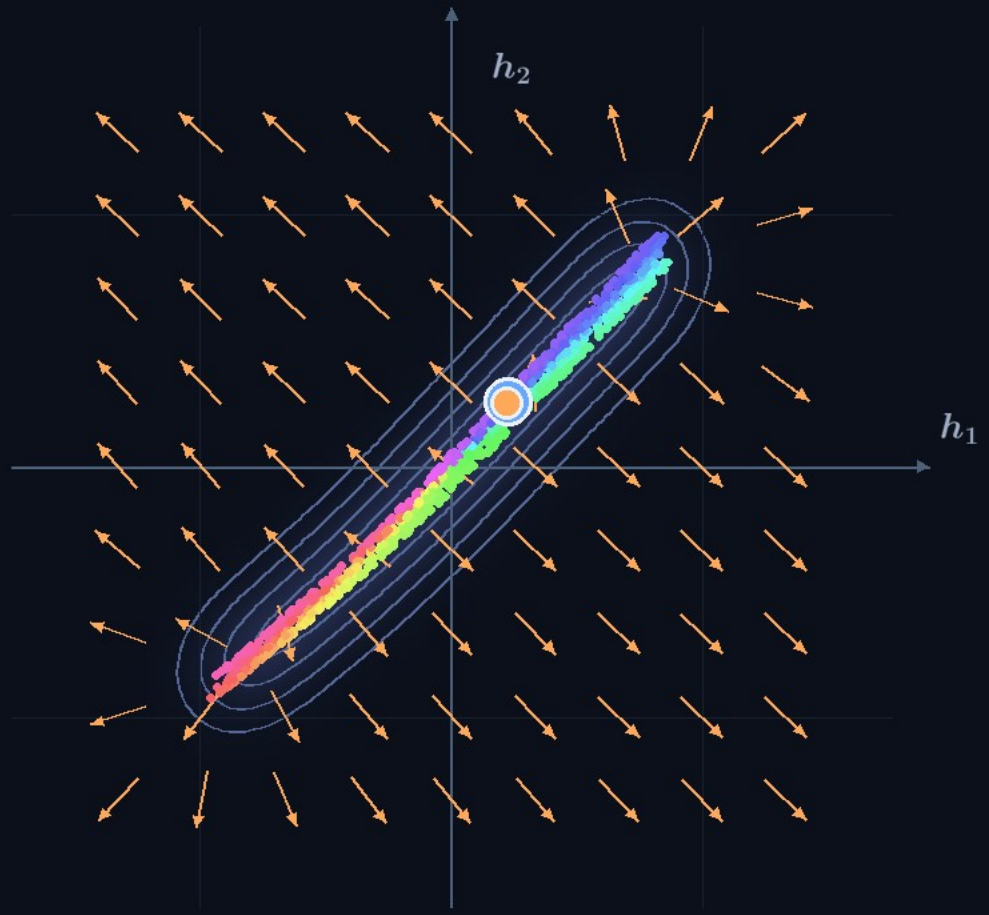




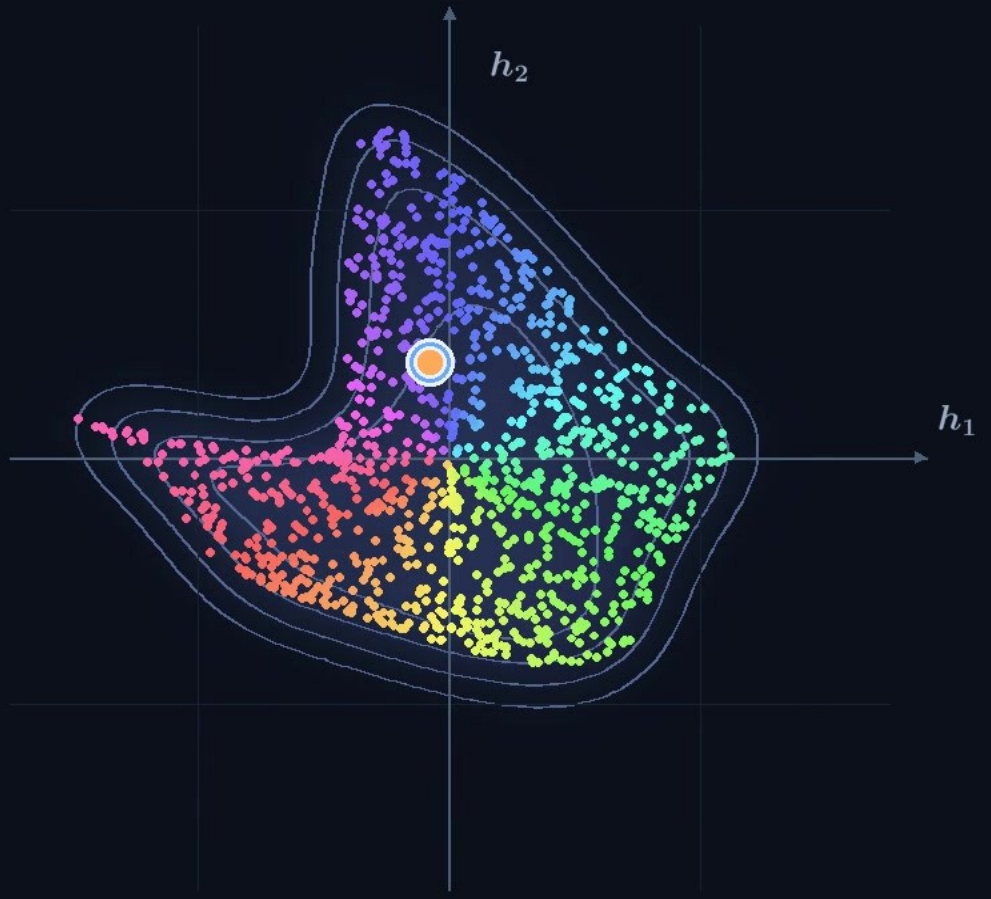
Can a learned latent score prevent collapse?



Can a learned latent score prevent collapse?



Preserve alignment. Recover diversity.



EBReg

Self-Supervised Collapse Prevention with
Score-Based Entropy Maximization

EBReg (Entropy Based Regularization)

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{align}} + \gamma \underbrace{\mathbb{E}\left[\text{sg}\left(s_{\psi,\sigma}(\tilde{h})\right)^{\top} \tilde{h}\right]}_{\mathcal{L}_{\text{EBReg}}}$$

$$\mathcal{L}_{\text{align}} = \frac{1}{d}\,\mathbb{E}\left\|h^{(1)} - h^{(2)}\right\|_2^2, \qquad \tilde{h} = h + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

Alignment: bring the two views of the same input together.

EBReg (Anti-Collapse Mechanism): use the learned score to encourage embedding diversity.

Hold the score fixed; update the encoder through $\tilde{h}$.

sg: stop-gradient.

An MI lower bound motivates alignment and entropy

Gaussian noise gives a well-defined density:

$$\tilde{h} = h + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad H_\sigma(h) = H(\tilde{h}), \quad \sigma > 0.$$

A conditional-entropy lower bound

$$I(h; h') \geq I(\tilde{h}; h') = \underbrace{H_\sigma(h)}_{\text{spread / anti-collapse}} - \underbrace{H(\tilde{h} \mid h')}_{\text{agreement / prediction}}.$$

An MI lower bound motivates alignment and entropy

Gaussian noise gives a well-defined density:

$$\tilde{h} = h + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad H_\sigma(h) = H(\tilde{h}), \quad \sigma > 0.$$

A conditional-entropy lower bound

$$I(h; h') \geq I(\tilde{h}; h') = \underbrace{H_\sigma(h)}_{\text{spread / anti-collapse}} - \underbrace{H(\tilde{h} \mid h')}_{\text{agreement / prediction}}.$$

A Gaussian variational conditional gives the alignment--entropy lower bound:

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

An MI lower bound motivates alignment and entropy

Gaussian noise gives a well-defined density:

$$\tilde{h} = h + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad H_\sigma(h) = H(\tilde{h}), \quad \sigma > 0.$$

A conditional-entropy lower bound

$$I(h; h') \geq I(\tilde{h}; h') = \underbrace{H_\sigma(h)}_{\text{spread / anti-collapse}} - \underbrace{H(\tilde{h} \mid h')}_{\text{agreement / prediction}}.$$

A Gaussian variational conditional gives the alignment--entropy lower bound:

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

Thus, the resulting lower bound gives the ideal objective:

$$\min_\theta \ \mathcal{L}_{\text{align}} - \gamma H_\sigma(h).$$

Our full objective is a gradient surrogate

Entropy is difficult to evaluate. **Its gradient only requires the score.**

$$s_\sigma(\tilde{h}) = \nabla_{\tilde{h}} \log p_{h,\sigma}(\tilde{h}), \qquad \nabla_\theta H_\sigma(h) = -\mathbb{E}\left[\left(\frac{\partial h}{\partial \theta}\right)^\top s_\sigma(\tilde{h})\right].$$

Our full objective is a gradient surrogate

Entropy is difficult to evaluate. **Its gradient only requires the score.**

$$s_\sigma(\tilde{h}) = \nabla_{\tilde{h}} \log p_{h,\sigma}(\tilde{h}), \qquad \nabla_\theta H_\sigma(h) = -\mathbb{E}\left[\left(\frac{\partial h}{\partial \theta}\right)^\top s_\sigma(\tilde{h})\right].$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{align}} + \gamma \underbrace{\mathbb{E}\left[\text{sg}\left(s_{\psi,\sigma}(\tilde{h})\right)^\top \tilde{h}\right]}_{\mathcal{L}_{\text{EBReg}}}.$$

Our full objective is a gradient surrogate

Entropy is difficult to evaluate. **Its gradient only requires the score.**

$$s_\sigma(\tilde{h}) = \nabla_{\tilde{h}} \log p_{h,\sigma}(\tilde{h}), \qquad \nabla_\theta H_\sigma(h) = -\mathbb{E}\left[\left(\frac{\partial h}{\partial \theta}\right)^\top s_\sigma(\tilde{h})\right].$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{align}} + \gamma \underbrace{\mathbb{E}\left[\text{sg}\left(s_{\psi,\sigma}(\tilde{h})\right)^\top \tilde{h}\right]}_{\mathcal{L}_{\text{EBReg}}}.$$

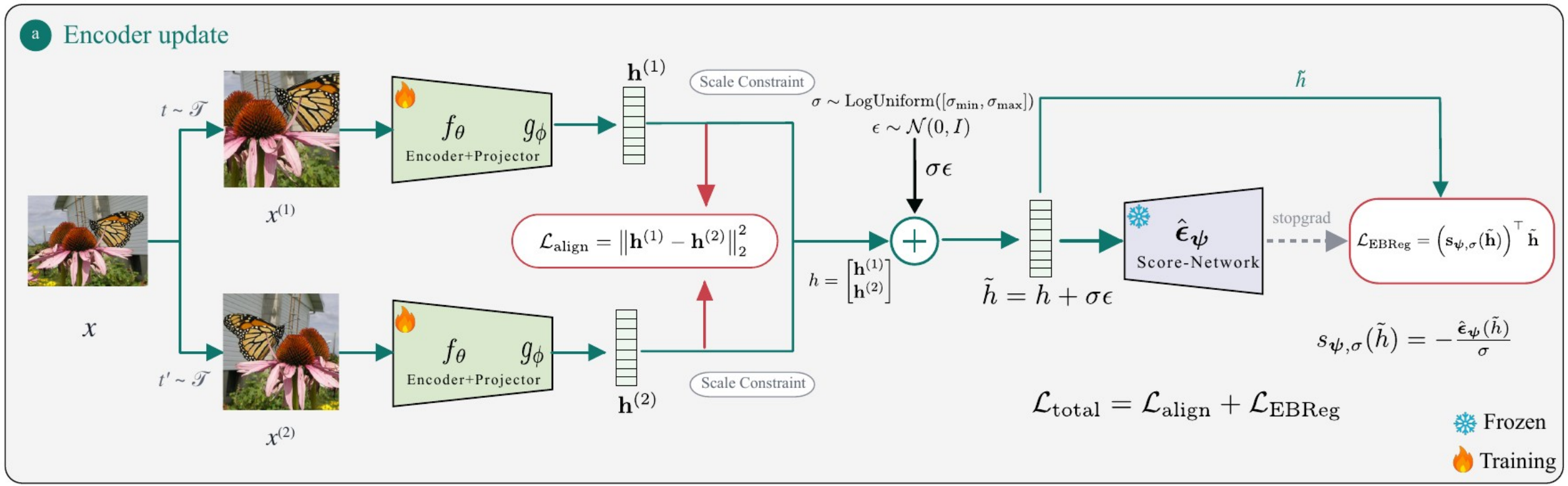
With a learned score, $s_{\psi,\sigma} \approx s_\sigma$:

$$\boxed{\nabla_\theta \mathcal{L}_{\text{total}} \approx \nabla_\theta \left[\mathcal{L}_{\text{align}} - \gamma H_\sigma(h)\right]}$$

The full objective is a gradient surrogate for the ideal objective.

Training alternates encoder updates and score fitting

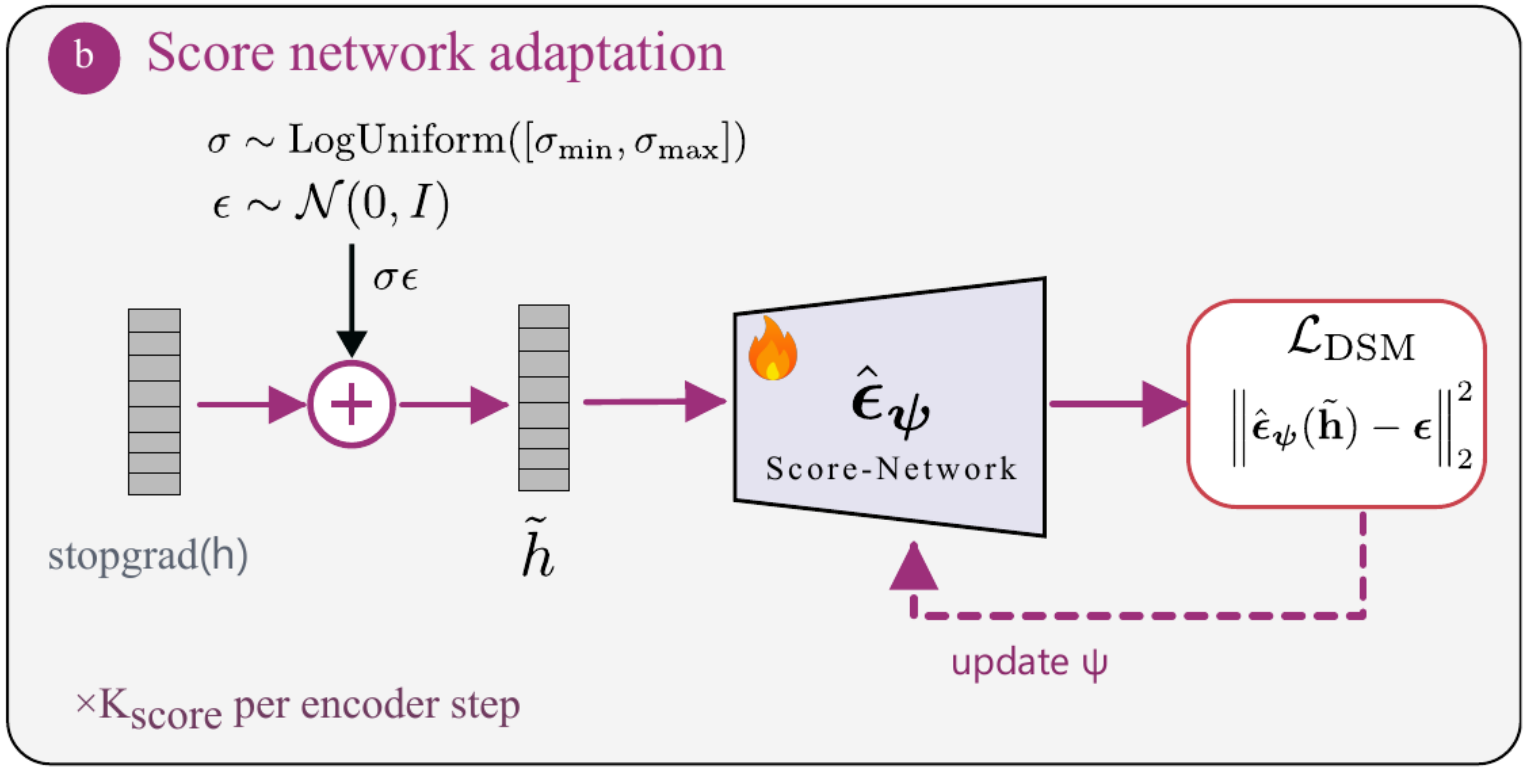
1. Update the encoder



Keep the score fixed; update the encoder with alignment and EBReg.

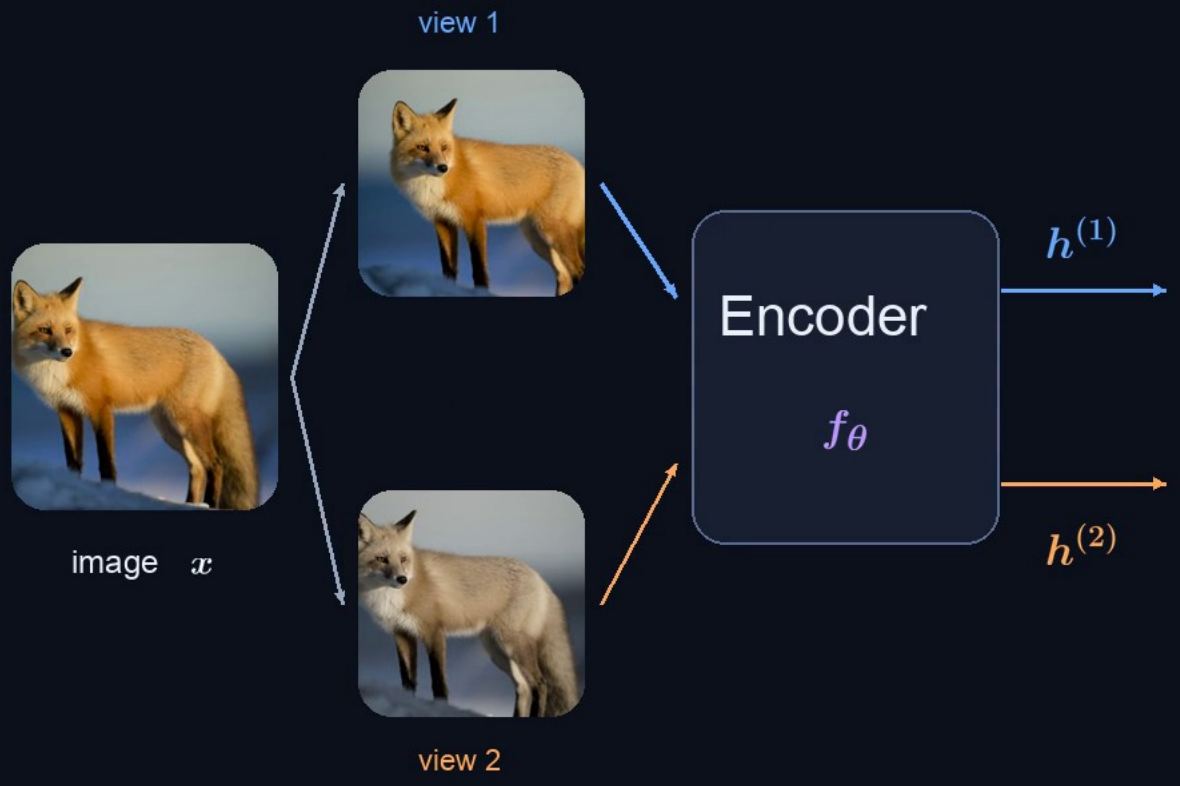
Training alternates encoder updates and score fitting

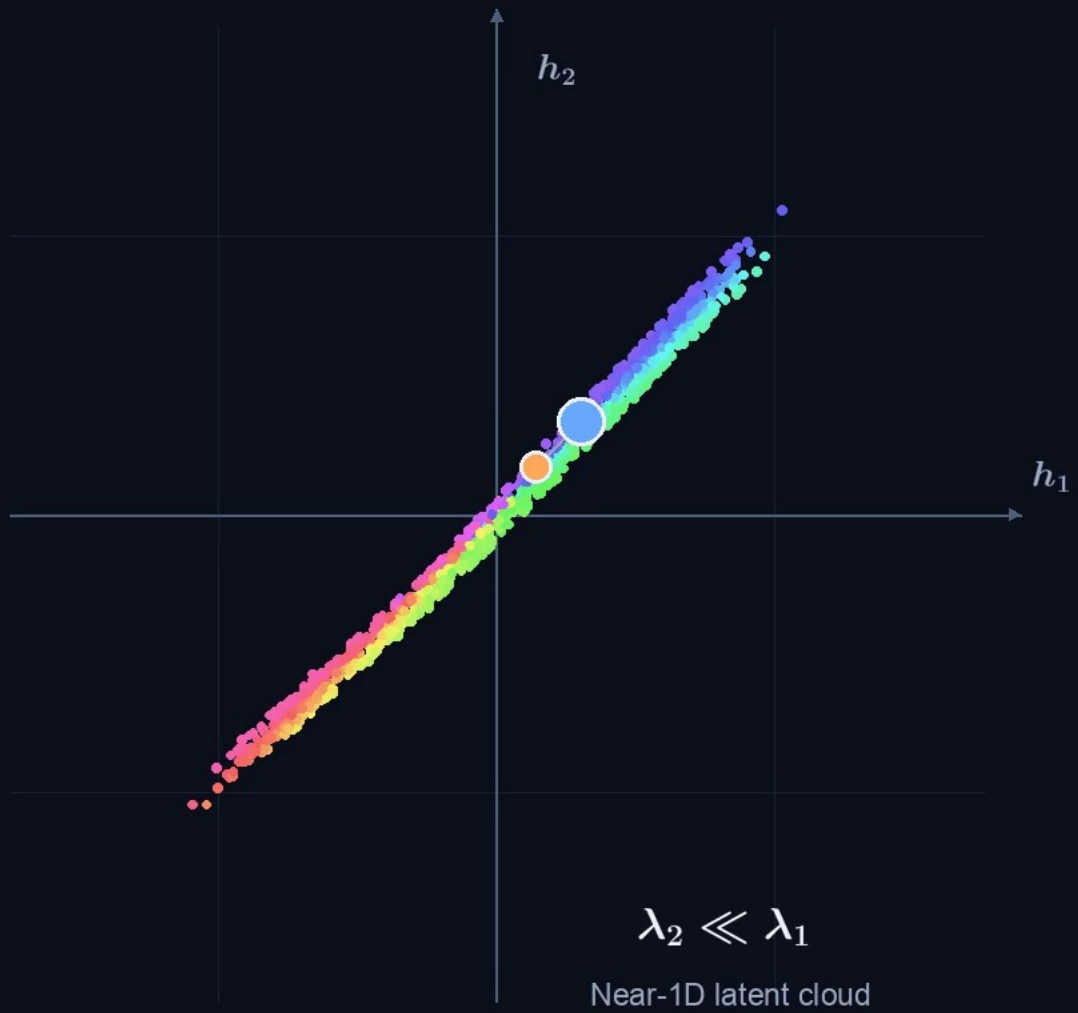
2. Fit the score to current embeddings



Detach embeddings; fit the noise predictor for K_{score} steps.

Two views. One encoder.

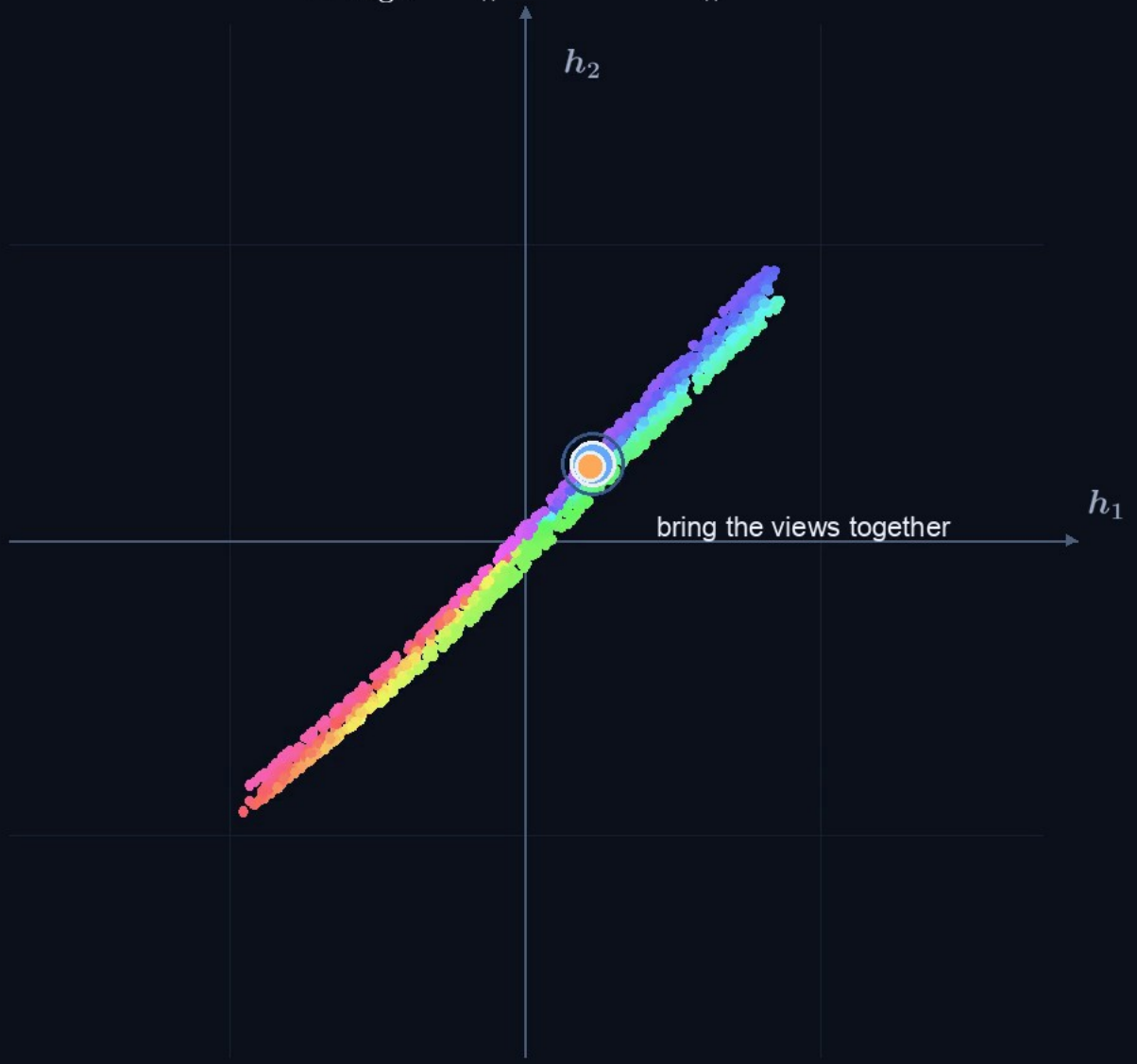




$$\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\, \mathbb{E}[\, \text{sg}(s_\psi(\tilde{h}))^\top \tilde{h}]$$

Bring the two views together.

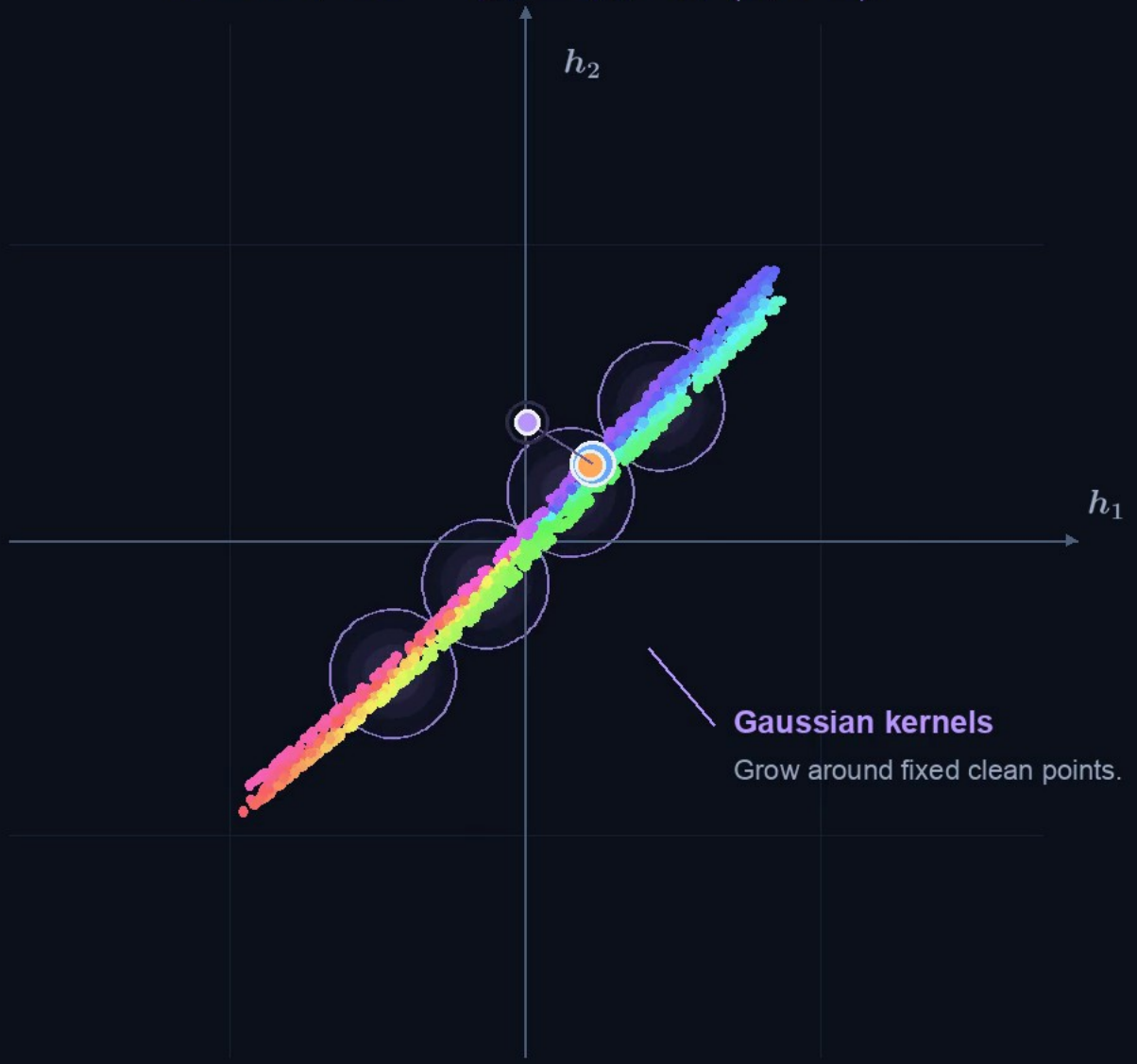
$$\mathcal{L}_{\text{align}} \propto \|h^{(1)} - h^{(2)}\|^2$$



$$\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\, \mathbb{E}[\, \text{sg}(s_\psi(\tilde{h}))^\top \tilde{h}]$$

Add Gaussian noise to smooth the latent density.

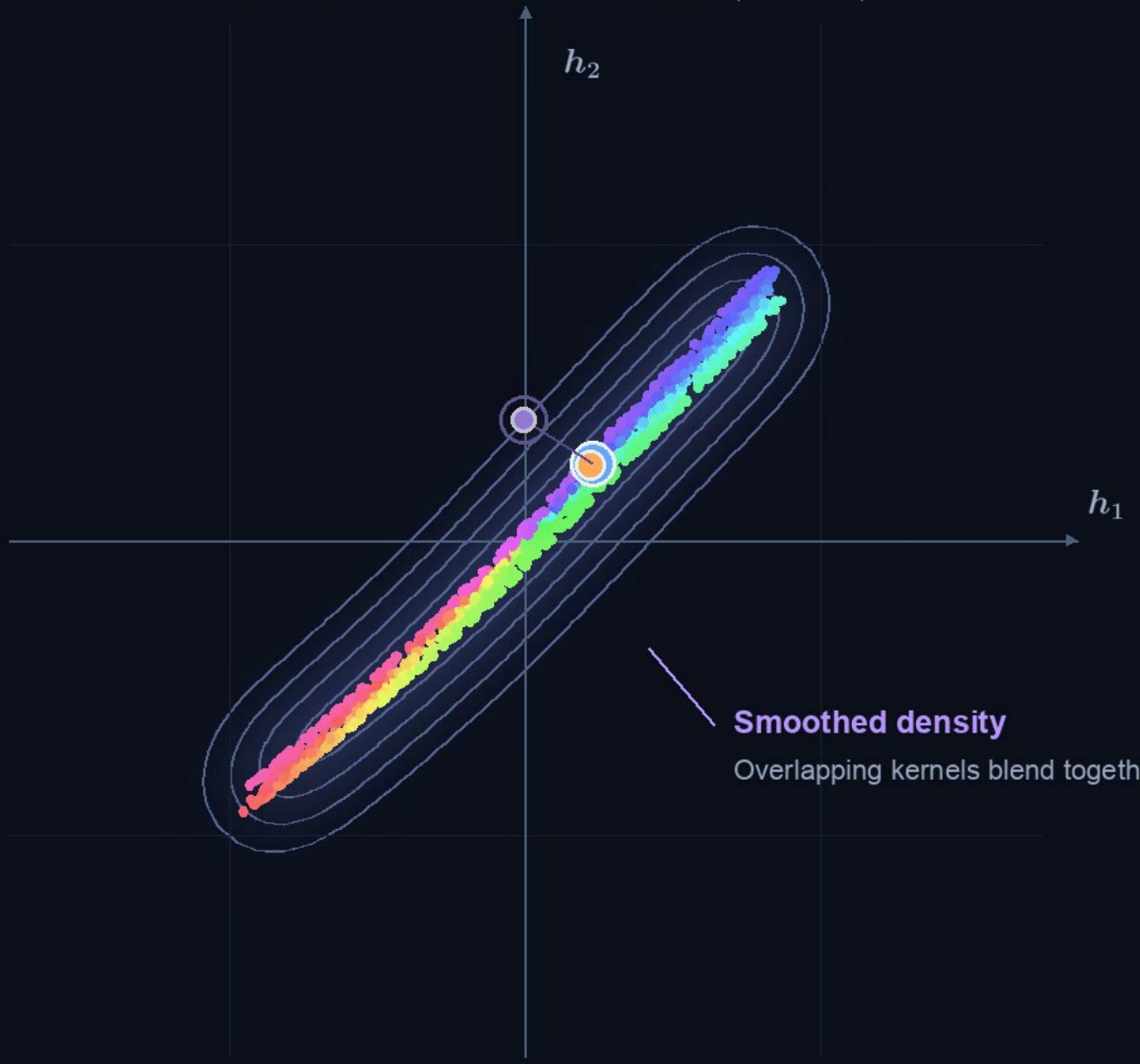
$$\tilde{h} = h + \sigma\epsilon \qquad p_\sigma = p_h * \mathcal{N}(0, \sigma^2 I)$$



$$\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\, \mathbb{E}[\, \text{sg}(s_\psi(\tilde{h}))^\top \tilde{h}]$$

Add Gaussian noise to smooth the latent density.

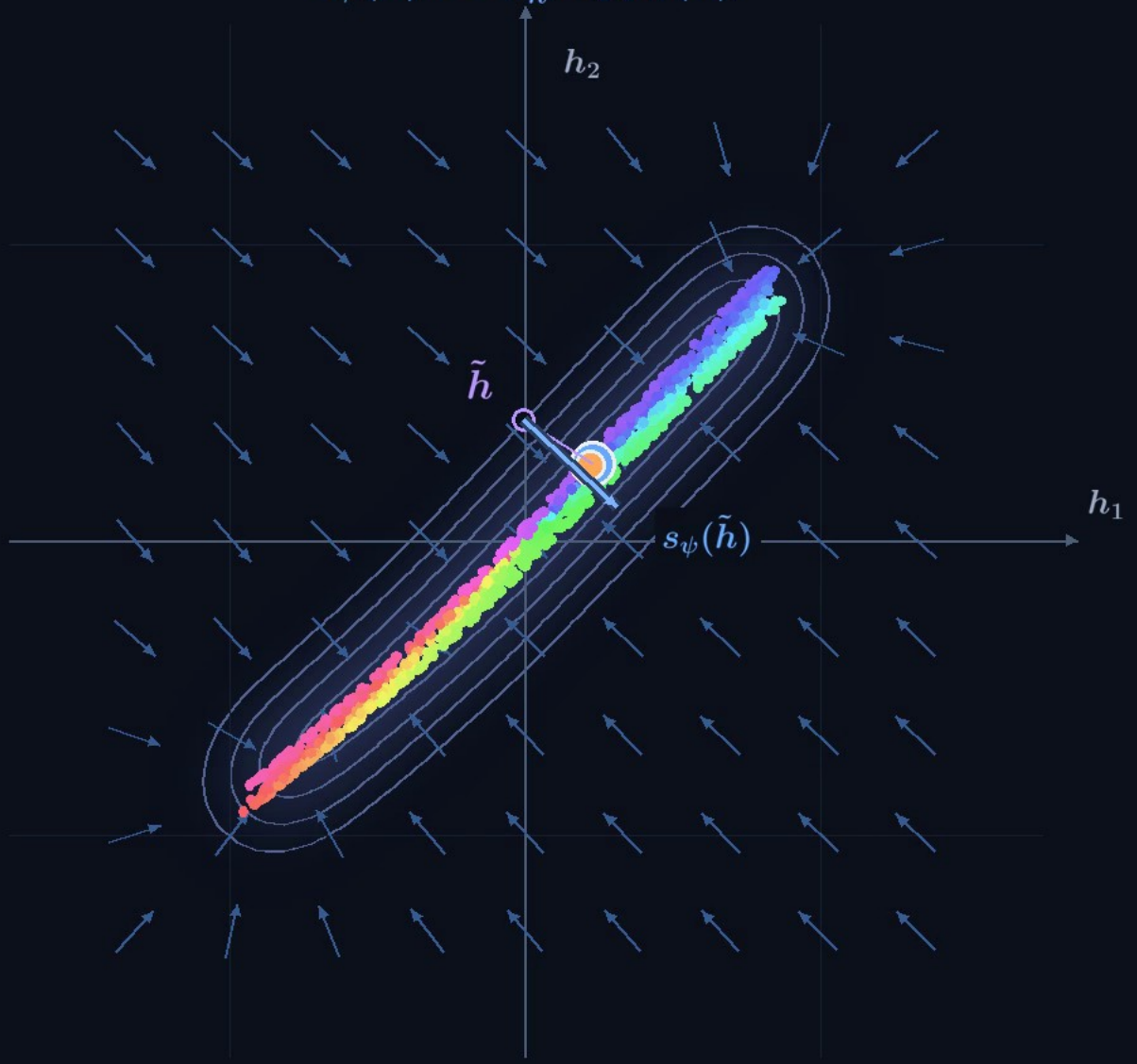
$$\tilde{h} = h + \sigma\epsilon \qquad p_\sigma = p_h * \mathcal{N}(0, \sigma^2 I)$$



$$\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\, \mathbb{E}[\, \text{sg}(s_\psi(\tilde{h}))^\top \tilde{h}]$$

Learn the score; encoder fixed.

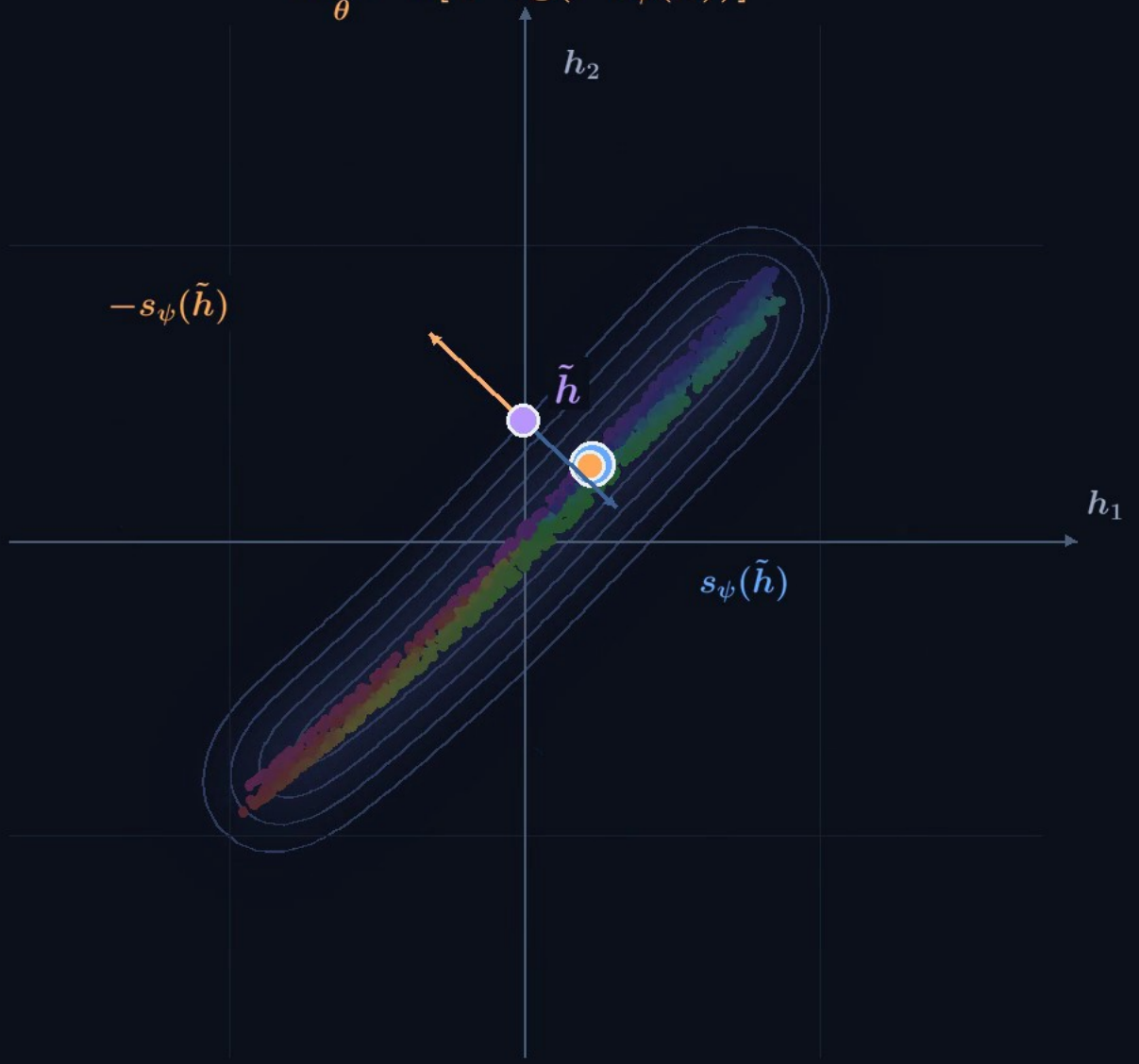
$$s_\psi(\tilde{h}) \approx \nabla_{\tilde{h}} \log p_\sigma(\tilde{h})$$



$$\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\, \mathbb{E}[\,\text{sg}(s_\psi(\tilde{h}))^\top \tilde{h}]$$

Increase similarity to the fixed negative score.

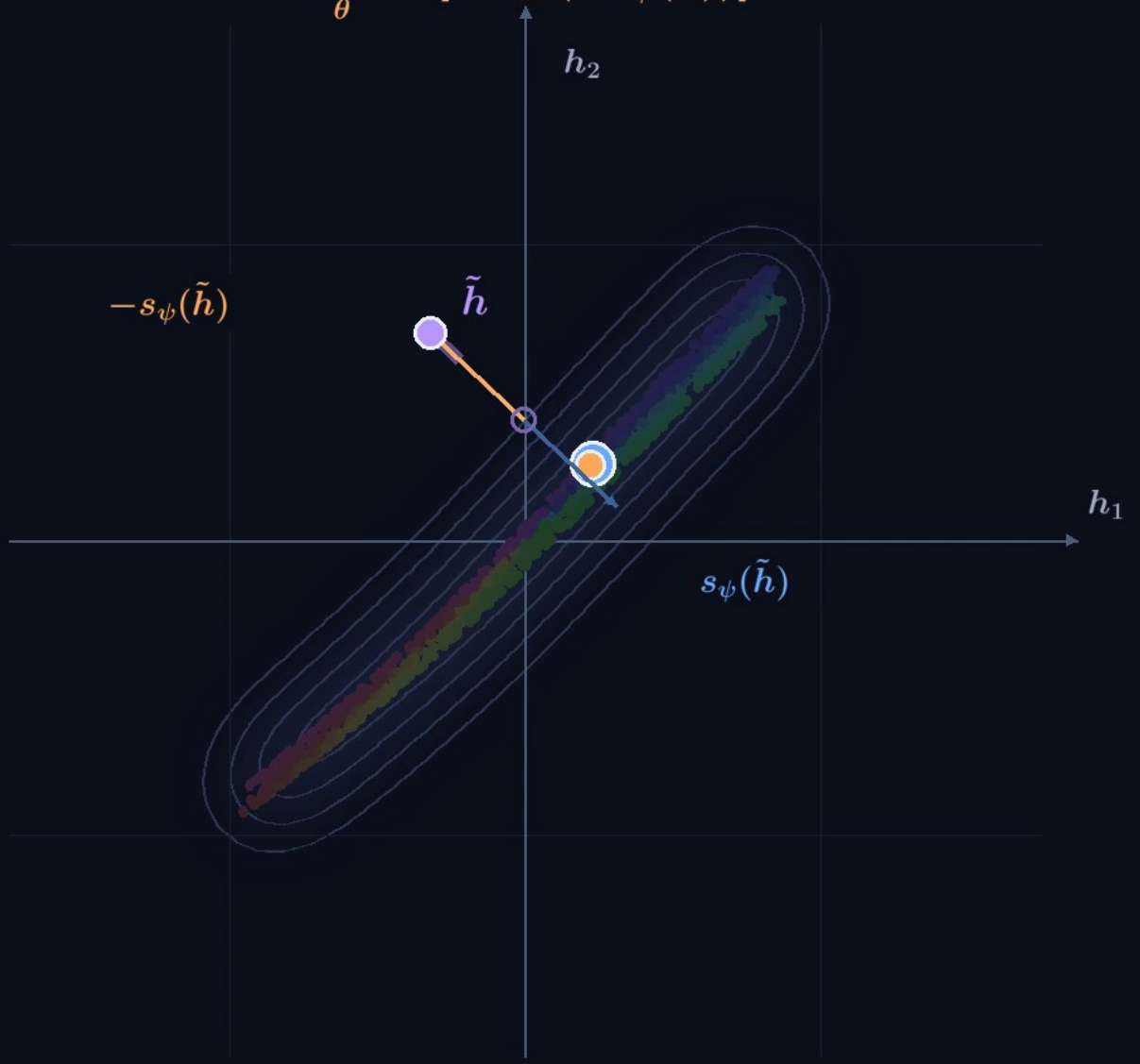
$$\max_{\theta}\ \mathbb{E}[\tilde{h}^\top \text{sg}(-s_\psi(\tilde{h}))]$$



$$\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\, \mathbb{E}[\, \text{sg}(s_\psi(\tilde{h}))^\top \tilde{h}]$$

Increase similarity to the fixed negative score.

$$\max_\theta \; \mathbb{E}[\tilde{h}^\top \text{sg}(-s_\psi(\tilde{h}))]$$



$$\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\, \mathbb{E}[\, \text{sg}(s_\psi(\tilde{h}))^\top \tilde{h}]$$

Optimize both terms; recover weak-direction variance.



PRINCIPAL VARIANCES

λ_1 1.06

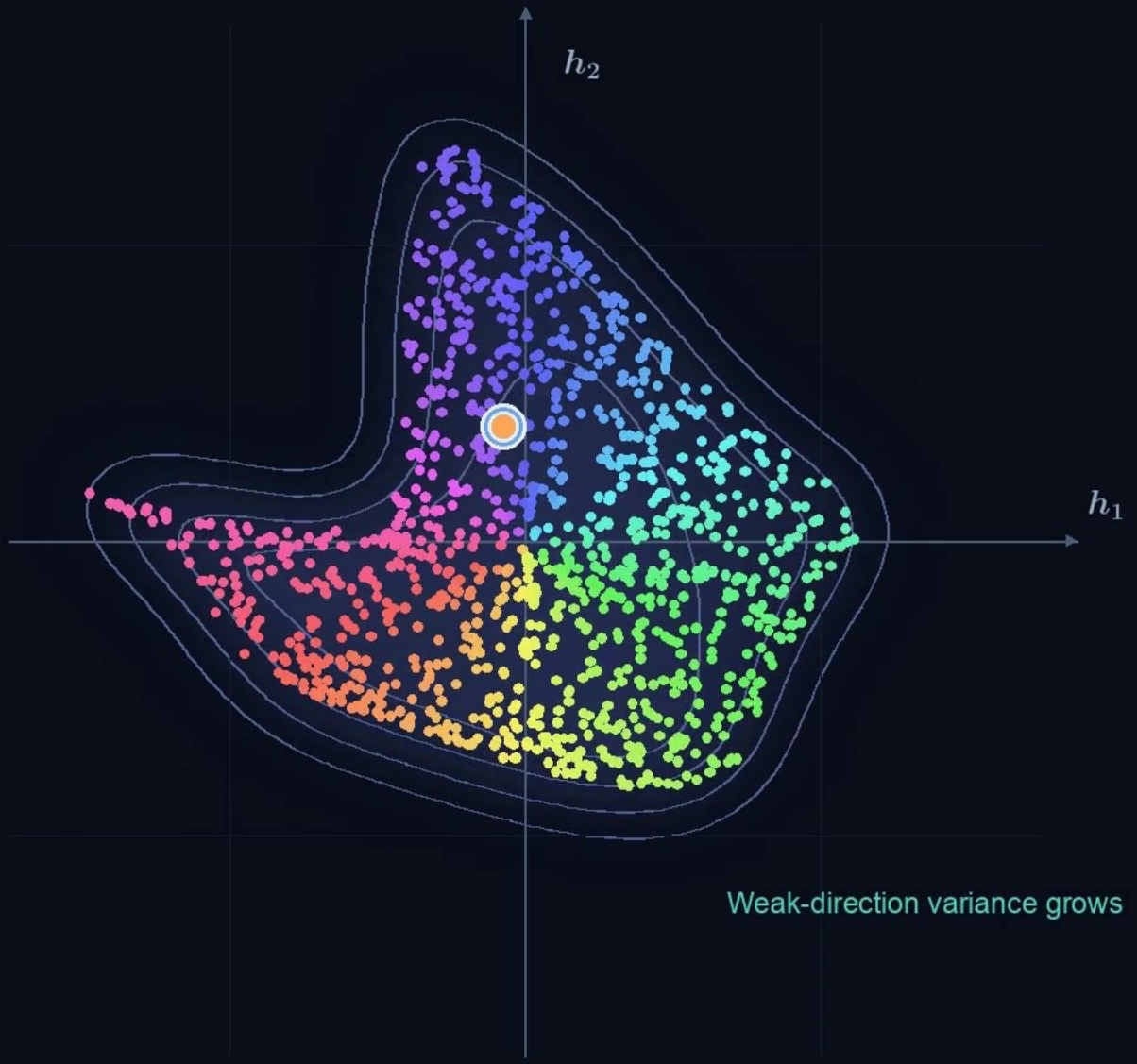
λ_2 0.94

$$\text{tr}\,\text{Cov}(h) = 2$$

Weak-direction variance grows

$$\mathcal{L} = \mathcal{L}_{\text{align}} + \gamma\, \mathbb{E}[\, \text{sg}(s_\psi(\tilde{h}))^\top \tilde{h}]$$

Optimize both terms; recover weak-direction variance.



PRINCIPAL VARIANCES

λ_1 1.06

λ_2 0.94

$$\text{tr}\,\text{Cov}(h) = 2$$

Weak-direction variance grows

Scale constraints connect EBReg to existing regularizers

$$H(ah) = H(h) + d \log |a|, \qquad a \neq 0.$$

Constrain scale to prevent entropy gains from rescaling alone.

Latent constraint	**Typical realization**	**Max.-entropy target**	**SSL connection**
$\mathbb{E}[h] = 0$, $\operatorname{tr} \operatorname{Cov}(h) = d$	Batch normalization without affine parameters	Isotropic Gaussian	SIGReg: Gaussian target; VICReg: first and second moments
$\|h\|_2 = r$	ℓ_2, RMS, or layer normalization	Uniform on the sphere	Contrastive learning
$h \in \{e_1, \dots, e_K\}$	Discrete prototype / code assignment	Uniform code usage	SwAV / MSN: balanced prototype usage
$\mathbb{E}\|h\|_p^p = M$, $h \in S$	p-th moment scaling with support constraint	Generalized Gaussian (truncated to S)	Rectified LpJEPA

ImageNet-1K: similar accuracy, higher effective rank

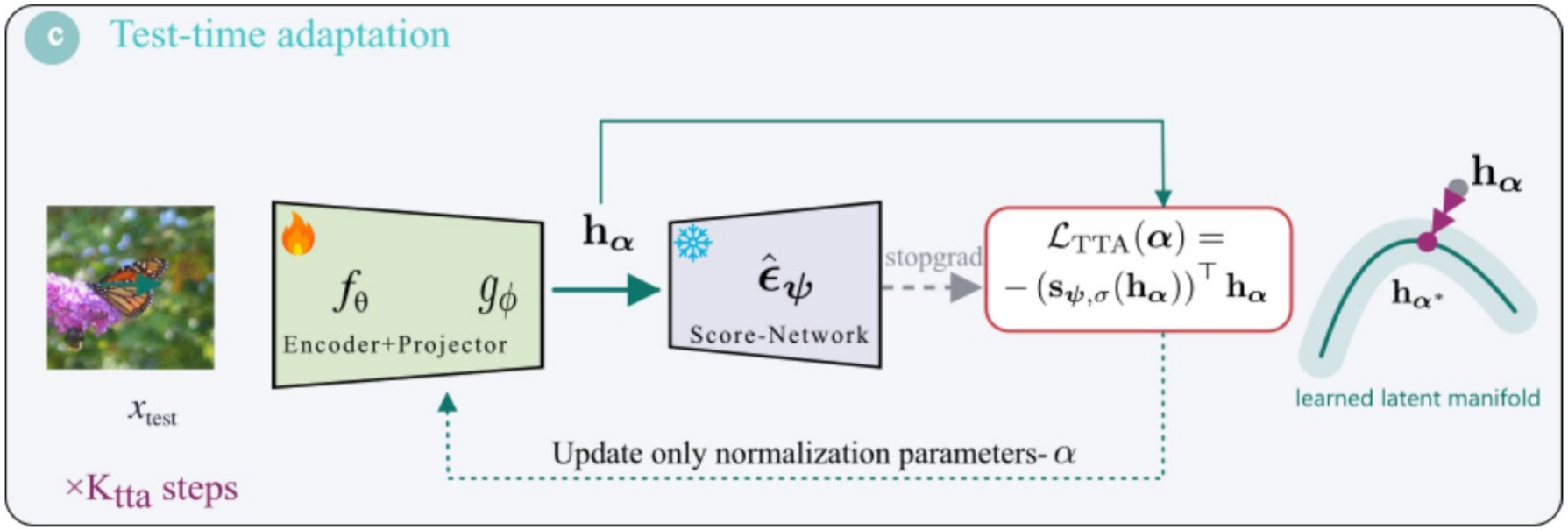
Backbone / comparison	Linear probe (%)	Change	Effective rank
ResNet-50 VICReg → EBReg	68.02 → 67.92	−0.10 pp	**1368 → 1431**
ViT-L SIGReg → EBReg	75.32 → 75.57	+0.25 pp	**648 → 759**
ViT-H StoP → StoP + EBReg	79.33 → 79.41	+0.08 pp	**527 → 957**

ViT-L: 13.6% longer epochs, under 2% extra parameters

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

A compact score model with moderate training overhead.

Reversing the sign guides test-time adaptation



Reverse the sign: pretraining promotes diversity; adaptation moves toward higher source density.

1. Freeze: keep the score model and all encoder--projector parameters fixed except normalization parameters α.

2. Adapt: take K_{TTA} gradient steps on unlabeled test inputs:

$$\mathscr{L}_{\mathrm{TTA}}(\alpha) = -\mathbb{E}\Big[\mathrm{sg}\big(s_{\psi,\sigma}(h_\alpha)\big)^\top h_\alpha\Big].$$

3. Predict: use the adapted encoder.

Five adaptation steps improve ImageNet-C mean accuracy

ViT-L: +1.63 pp **ViT-H: +3.26 pp** **ResNet-50: +1.83 pp**

Mean accuracy gains across all 15 corruptions.

Accuracy (%): before → after **five adaptation steps**.

Corruption	**ViT-L**	**ViT-H**	**ResNet-50**
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 (all 15)	**46.55 → 48.18**	**57.79 → 61.05**	**28.67 → 30.50**

Seven corruptions selected by mean gain, with gains for every model; the final row averages all 15 corruptions.

Takeaways and future directions

1. **EBReg:** alternate denoising score fitting with alignment + detached-score encoder updates.
2. **Contrastive learning, but no negatives :)**
3. **A compact score model with moderate training overhead** and test-time benefits.
4. **Mathematically grounded:** derived from a mutual-information lower bound and the entropy-gradient identity.
5. **Beyond individual instances:** this idea could extend to clusters or prototypes, as in DINO.

Thank You.

Questions?