SLICE SELECTION MATTERS: STRUCTURED AND ADAPTIVE DIRECTIONS FOR SLICED LATENT REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sliced latent regularization typically uses random projections to pursue Gaussian-target matching, but these directions can conceal discrepancies along directions populated by the encoder. We investigate deterministic rotation and propose a simple radial-adaptive augmentation to better fulfill the regularization objective. For D-dimensional latents, we append K normalized minibatch vectors to D random or rotating base directions, selecting vectors whose lengths are unusually small or large relative to the isotropic Gaussian target. Recomputed at each training step, this construction requires no additional learned parameters or inner optimization and supports both sliced Wasserstein-2 and Epps–Pulley losses. Across autoencoding and JEPA-style self-supervision on MNIST and ImageNet-100, matched-budget controls replace the K adaptive directions with K random directions, separating augmentation from base schedule and slice count. Over ten matched seeds, radial-adaptive augmentation lowers mean held-out latent-direction Kolmogorov–Smirnov discrepancy in all 16 comparisons spanning datasets, objectives, losses, and base schedules. Additional evaluation with disjoint samples preserves these mean improvements. Deterministic rotation alone has no universal advantage, whereas adaptation benefits both base schedules. A residual directional gap remains, so improved projected agreement does not establish complete Gaussian matching. We therefore recommend radial-adaptive augmentation as a simple complement to sliced Gaussian regularization when improved target matching is the goal, together with evaluation using both random and latent directions.

1 INTRODUCTION

Latent regularization shapes representation geometry in both generative modeling and self-supervised learning. Autoencoders often match latents to a simple prior, such as an isotropic Gaussian, including through Wasserstein and sliced Wasserstein objectives (Tolstikhin et al., 2018; Kolouri et al., 2019). Self-supervised methods similarly require non-collapsed representations with useful global structure (Assran et al., 2023; Balestriero & LeCun, 2025; Wu et al., 2026). A scalable approach compares one-dimensional latent projections with the standard normal reference.

Projected matching replaces a high-dimensional problem with inexpensive slice-wise discrepancies, including Wasserstein and characteristic-function criteria (Knop et al., 2020; Cramér & Wold, 1936). Yet most methods sample directions randomly and treat them as a Monte Carlo detail. If the training signal itself is defined by projected discrepancies, this choice need not be neutral.

We study *slice selection* as part of the regularizer. We explore a deterministic rotating frame and radial-adaptive augmentation of either random or rotating base directions, using sliced Wasserstein-2 and Epps–Pulley losses. We evaluate the resulting representations with both random and latent-derived directions across autoencoding and JEPA-style self-supervision on MNIST (LeCun et al., 1998) and ImageNet-100 (Tian et al., 2020), a fixed 100-class subset of ImageNet (Deng et al., 2009; Russakovsky et al., 2015). The central question is whether direction choice changes both the optimization signal and what is concluded about matching the isotropic Gaussian target.

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.

2 Background and Related Work

Latent prior matching through optimal transport and sliced projections underlies Wasserstein Auto-Encoders, Sliced-Wasserstein Autoencoders, and the Cramer–Wold AutoEncoder (Tolstikhin et al., 2018; Kolouri et al., 2019; Knop et al., 2020). Sliced optimal transport includes distributional and max-sliced variants (Nguyen et al., 2021; Deshpande et al., 2019) and efficient estimators (Nguyen et al., 2026; Piening & Beinert, 2026). Random-path sliced Wasserstein constructs optimization-free, data-informed directions from normalized differences between samples from the two input measures (Nguyen et al., 2024). Our contribution is not data-informed slicing in general, but radial-atypicality-based augmentation for latent regularization and comparing random- versus latent-direction evaluation.

JEPA-style methods instead learn representation-space predictions (Assran et al., 2023). Recent work highlights the importance of the anti-collapse mechanism and regularizer (Balestriero & LeCun, 2025; Wu et al., 2026; Ponce et al., 2026; Zhang et al., 2026; Li et al., 2026). We use both autoencoding and JEPA-style SSL to study how structured and adaptive directions affect latent regularization and the interpretation of projected agreement with the isotropic Gaussian target.

3 Method

The framework combines a sliced Gaussianity loss, a training-time direction schedule, and an AE or JEPA-style objective.

3.1 Latent Regularization by Sliced Distribution Matching

An encoder E_θ maps an input $x \in \mathcal{X}$ to a latent representation $z = E_\theta(x) \in \mathbb{R}^D$. For a mini-batch $\mathcal{B} = \{x_n\}_{n=1}^B$, let $Z = [z_1^\top, \dots, z_B^\top]^\top \in \mathbb{R}^{B\times D}$. We regularize its one-dimensional projections toward the standard normal distribution.

At training step t, let $\mathcal{S}_t = \{s_{t,1}, \dots, s_{t,K_t}\}$ be the bank of unit slice directions in $\mathbb{R}^D$, where K_t is the total number of slices used at that step. The unsorted projections are $u_{n,k}^{(t)} = z_n^\top s_{t,k}$ for $n = 1, \dots, B$. Their sorted values along slice k are denoted by $p_{(1),k}^{(t)} \le \cdots \le p_{(B),k}^{(t)}$.

We use transport-based Wasserstein-2 and characteristic-function-based Epps–Pulley losses for training. The CDF-based Kolmogorov–Smirnov criterion is reserved for evaluation because its max-deviation structure provides a less distributed optimization signal.

3.1.1 Sliced Wasserstein-2

Let $q_1, \dots, q_B$ denote the deterministic Gaussian reference quantiles

$$q_i = \Phi^{-1}\left(\frac{i - \frac{1}{2}}{B}\right), \quad i = 1, \dots, B, \tag{1}$$

where Φ is the standard normal cumulative distribution function. The sliced squared Wasserstein-2 approximation is

$$\mathcal{L}_{\mathrm{WS2}}(Z; \mathcal{S}_t) = \frac{1}{K_t B} \sum_{k=1}^{K_t} \sum_{i=1}^{B} \left(p_{(i),k}^{(t)} - q_i\right)^2. \tag{2}$$

The objective averages squared Wasserstein-2 distances to the equal-mass reference at $\{q_i\}_{i=1}^B$, giving a midpoint-quantile approximation to matching $\mathcal{N}(0,1)$. We denote Eq. (2) by WS2.

3.1.2 Sliced Epps–Pulley

We use the Epps–Pulley (EP) characteristic-function criterion (Epps & Pulley, 1983) for Gaussian-target matching, following its use in LeJEPA's SIGReg (Balestriero & LeCun, 2025). For a fixed slice k, let

$$\widehat{\varphi}_k(\tau) = \frac{1}{B} \sum_{n=1}^{B} e^{i\tau u_{n,k}^{(t)}} \tag{3}$$

denote the empirical characteristic function. The standard-normal characteristic function and Gaussian frequency window are $\varphi_{\mathcal{N}}(\tau) = w(\tau) = e^{-\tau^2/2}$. The sliced EP discrepancy is

$$\mathcal{L}_{\mathrm{EP}}(Z; \mathcal{S}_t) = \frac{1}{K_t} \sum_{k=1}^{K_t} B \int_{\mathbb{R}} |\widehat{\varphi}_k(\tau) - \varphi_{\mathcal{N}}(\tau)|^2 \, w(\tau) \, d\tau. \tag{4}$$

In practice, we approximate the integral in Eq. (4) using trapezoidal quadrature on a fixed symmetric grid $\tau_1 < \cdots < \tau_T$ in $[-\tau_{\max}, \tau_{\max}]$. Writing $\Delta_k(\tau_j) = |\widehat{\varphi}_k(\tau_j) - \varphi_{\mathcal{N}}(\tau_j)|^2 w(\tau_j)$, the discrete approximation becomes

$$\mathcal{L}_{\mathrm{EP}}(Z; \mathcal{S}_t) \approx \frac{1}{K_t} \sum_{k=1}^{K_t} B \sum_{j=1}^{T} \omega_j \, \Delta_k(\tau_j), \tag{5}$$

where ω_j are the trapezoidal weights. For a uniform grid with spacing h, $\omega_1 = \omega_T = h/2$ and $\omega_j = h$ for $j = 2, \ldots, T-1$.

In our implementation, we use $T = 17$ uniformly spaced frequencies on $[-5, 5]$. We use EP to denote the discrepancy in Eq. (4), evaluated numerically using Eq. (5).

3.1.3 Sliced Kolmogorov–Smirnov

We also consider the Kolmogorov–Smirnov (KS) discrepancy, which compares projected samples to the Gaussian target through their empirical cumulative distribution functions. For each slice k, define the empirical CDF of the projected latent samples as $\widehat{F}_k$, and let Φ denote the standard normal CDF. The per-slice KS statistic is

$$D_k = \sup_{u \in \mathbb{R}} \left| \widehat{F}_k(u) - \Phi(u) \right|. \tag{6}$$

We summarize the slice-wise discrepancies by their mean,

$$\mathrm{KS}_{\mathrm{avg}} = \frac{1}{K_t} \sum_{k=1}^{K_t} D_k. \tag{7}$$

Unlike WS2 and EP, KS is used only for evaluation.

3.2 Per-Batch Structured Rotating Slice Directions

The choice of slice directions is central. Standard sliced methods typically draw i.i.d. random directions at every iteration, which introduces additional Monte Carlo variation into the regularization process. Instead, we explore a deterministic, per-batch rotating schedule that evolves an orthogonal frame over training.

Let $Q_t \in \mathbb{R}^{D \times D}$ be the orthogonal frame before the update for batch t, with batches indexed from $t = 0$ and the initial frame given by

$$Q_0 = I_D. \tag{8}$$

At each batch, we update Q_t by composing a structured set of Givens rotations. Specifically, for a selected pair of coordinates (i, j) and rotation angle θ, define the Givens rotation $G(i, j, \theta) \in \mathbb{R}^{D \times D}$ so that the submatrix formed by rows and columns i, j is

$$G(i, j, \theta)_{\{i,j\},\{i,j\}} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{9}$$

with all remaining diagonal entries equal to 1 and all other off-diagonal entries equal to 0.

For each batch index t, we construct a round-robin set of disjoint coordinate pairs $\mathcal{P}_t = \{(i_{t,m}, j_{t,m})\}_{m=0}^{M_t - 1}$, chosen so that the pairs systematically cover coordinate interactions over time. Pair indices follow the round-robin order and start at zero, matching the implementation. We associate to each pair a deterministic angle

$$\theta_{t,m} = \frac{\pi}{4} + \frac{\pi}{8} \left(2 \left(\left(t + \frac{m}{M_t} \right) \alpha \bmod 1 \right) - 1 \right), \tag{10}$$

where $\alpha = (\sqrt{5}-1)/2$ is the golden-ratio conjugate. This produces a quasi-uniform deterministic phase schedule across batches and pair indices.

The orthogonal frame is then updated by

$$Q_{t+1} = \left(\prod_{m=0}^{M_t-1} G(i_{t,m}, j_{t,m}, \theta_{t,m}) \right) Q_t. \tag{11}$$

We use the columns of the updated frame as deterministic slice directions,

$$\mathcal{S}_t^{\mathrm{det}} = \{q_{t+1,1}, \ldots, q_{t+1,D}\}, \tag{12}$$

where $q_{t+1,k}$ is the k-th column of Q_{t+1}. Thus, batch t uses the updated frame, which is then carried forward to the next batch. Since Q_{t+1} remains orthogonal, these directions form an orthonormal basis at every batch.

This schedule removes fresh direction-sampling randomness while changing the orthonormal basis across batches. It provides structured variation, without implying uniform coverage of the sphere.

3.3 Radial-Adaptive Slice Augmentation

We consider augmenting either random or rotating base directions with a small number of data-dependent directions extracted from the current latent batch.

Under $z_n \sim \mathcal{N}(0, I_D)$, the squared radius $r_n^2 = \|z_n\|_2^2$ is exactly χ_D^2-distributed, being the sum of D independent squared standard normals. We therefore compute a two-sided radial atypicality score

$$a_n = -\log\left(2\min\{F_{\chi_D^2}(r_n^2), 1 - F_{\chi_D^2}(r_n^2)\} + \varepsilon\right), \tag{13}$$

where $F_{\chi_D^2}$ is the chi-square CDF and $\varepsilon > 0$ is a small numerical constant.

Since adaptive directions are selected without replacement from the latent batch, K_{adapt} cannot exceed its size, which is B for AE and $2B$ for two-view JEPA. We select up to K_{adapt} latent samples with the largest scores, optionally after thresholding $a_n \geq \tau$, and normalize them to obtain adaptive slice directions,

$$\mathcal{S}_t^{\mathrm{adapt}} = \left\{ \frac{z_n}{\|z_n\|_2} \;:\; n \in \mathrm{TopK}_\tau(a_1, \ldots, a_B) \right\}. \tag{14}$$

For the hybrid method, training uses $\mathcal{S}_t^{\mathrm{det}} \cup \mathcal{S}_t^{\mathrm{adapt}}$.

The two-sided score selects latent vectors with unusually small or large Euclidean norms relative to the Gaussian reference. It chooses directions rather than directly penalizing radii or maximizing projected KS. Direction construction is detached from gradient computation, while the sliced loss uses projections of all batch samples onto every selected direction. Thus, selected samples determine where to evaluate the discrepancy, not which samples are regularized. All directions receive equal weight in the sliced average.

3.4 Training Slice Variants

For consistency, all experiments use D base slices, matching the latent dimension. We compare six constructions. Rand-D draws D independent unit directions, whereas Rot-D uses the D directions of the rotating orthogonal frame $\mathcal{S}_t^{\mathrm{det}}$. At budget $D + K_{\mathrm{adapt}}$, Rand-(D+K) draws all directions independently. Rand+Adapt-(D+K) combines D random directions with the K_{adapt} radial-adaptive directions in Eq. (14). Rot+Rand-(D+K) combines $\mathcal{S}_t^{\mathrm{det}}$ with K_{adapt} random directions. Rot+Adapt-(D+K) combines $\mathcal{S}_t^{\mathrm{det}}$ with the radial-adaptive directions. The four $D + K_{\mathrm{adapt}}$ variants form a 2×2 factorial design over base schedule (random or rotating) and augmentation (random or radial-adaptive).

3.5 Task-Specific Training Objectives

AE and JEPA-style pretraining combine a task loss with the sliced regularizer. Frozen linear probes subsequently check retained class information.

3.5.1 WS2 AND EP LOSS SCALE CALIBRATION

For both objectives, $\mathcal{L}_{\text{slice}}$ is $c_{\text{WS2}\to\text{EP}}\mathcal{L}_{\text{WS2}}$ for WS2 training or $\mathcal{L}_{\text{EP}}$ for EP training. To use a common λ_{slice}, the factor $c_{\text{WS2}\to\text{EP}}$ matches deterministic WS2 and population EP values at the partial variance-collapse reference $X \sim \mathcal{N}(0, 0.5^2)$. The calculation uses $B_{\text{eff}} = B$ for AE and $B_{\text{eff}} = 2B$ for two-view JEPA, and the same 17-point EP frequency grid as training. It is a local scale match, not a global equivalence between the objectives.

3.5.2 AUTOENCODING

Let D_ϕ denote a decoder and $\hat{x}_n = D_\phi(z_n)$. We use the reconstruction objective

$$\mathcal{L}_{\text{rec}} = \frac{1}{B}\sum_{n=1}^{B} \ell_{\text{rec}}(x_n, \hat{x}_n), \tag{15}$$

where ℓ_{rec} is typically mean-squared error. The total objective is

$$\mathcal{L}_{\text{AE}} = (1-\lambda_{\text{slice}})\mathcal{L}_{\text{rec}} + \lambda_{\text{slice}}\mathcal{L}_{\text{slice}}(Z; \mathcal{S}_t). \tag{16}$$

3.5.3 SELF-SUPERVISED LEARNING

For two augmented views $x_n^{(1)}, x_n^{(2)}$ of the same input, the shared encoder gives $z_n^{(v)} = E_\theta(x_n^{(v)})$, $v \in \{1, 2\}$. A shared predictor P_ψ operates bidirectionally. Gradients propagate through both prediction targets. No stop-gradient, EMA teacher, or separate target encoder is used. The symmetric prediction objective is

$$\mathcal{L}_{\text{SSL}} = \frac{1}{2BD}\sum_{n=1}^{B}\left[\left\|P_\psi(z_n^{(1)}) - z_n^{(2)}\right\|_2^2 + \left\|P_\psi(z_n^{(2)}) - z_n^{(1)}\right\|_2^2\right]. \tag{17}$$

The prediction error is averaged over both prediction directions, the B samples, and the D latent coordinates, matching the elementwise mean-squared error used in the implementation. We regularize the concatenated encoder-latent batch, before the predictor,

$$Z_{\text{SSL}} = \left[z_1^{(1)}, \ldots, z_B^{(1)}, z_1^{(2)}, \ldots, z_B^{(2)}\right]^\top, \tag{18}$$

and optimize

$$\mathcal{L}_{\text{total}} = (1-\lambda_{\text{slice}})\mathcal{L}_{\text{SSL}} + \lambda_{\text{slice}}\mathcal{L}_{\text{slice}}(Z_{\text{SSL}}; \mathcal{S}_t). \tag{19}$$

3.5.4 SUPERVISED LINEAR PROBING

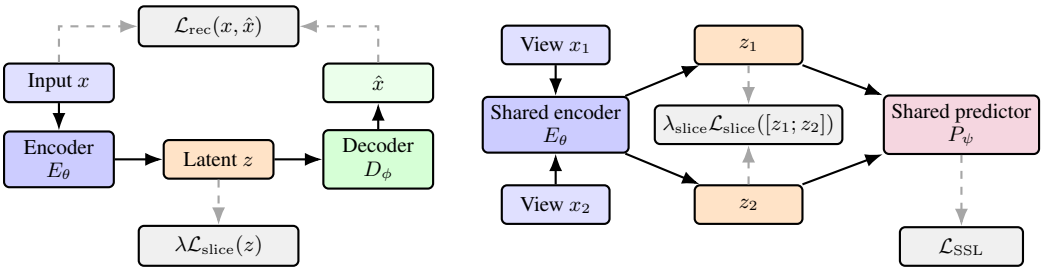
After AE or JEPA pretraining, we freeze the encoder E_θ and train a linear classifier h_ω on labeled training representations using cross-entropy. Probe accuracy checks retained class information as a diagnostic against complete collapse, rather than measuring competitiveness with fully tuned SSL systems.

4 EXPERIMENTS

We study MNIST and ImageNet-100 with matched multi-seed autoencoder (AE) and JEPA-style objectives (Figure 1). MNIST uses $z \in \mathbb{R}^{64}$ and ImageNet-100 uses $z \in \mathbb{R}^{256}$, with D base directions in both cases. Within each dataset and objective, architecture, optimizer, schedule, batch size, latent dimension, and seeds are fixed. Only the training loss (WS2 or EP) and the six slice constructions in Section 3.4 vary. Task losses monitor reconstruction or prediction fit, while separately trained linear probes check class information in frozen representations. Table 1 gives the fixed configuration.

Evaluation uses KS_{avg} and Parzen density plots, while WS2 and EP are training losses. Markers represent training seeds unless stated otherwise. Paired comparisons use exact two-sided Wilcoxon signed-rank tests across ten matched seeds, not individual projections. Reported p-values are unadjusted for multiple comparisons, and significance uses nominal $\alpha = 0.01$.

The ImageNet-100 experiments are deliberately controlled ablations rather than comparisons with full-scale SSL systems. They use a randomly initialized ResNet-18, a 256-dimensional latent space,



(a) Autoencoder **(b) JEPA-style SSL**

Figure 1: Experimental pipelines. **(a)** The AE optimizes reconstruction together with a sliced loss on the encoder latent. **(b)** JEPA-style SSL shares an encoder and predictor across two views and combines the symmetric prediction loss with one sliced loss on Z_{SSL}, the concatenated batch of $2B$ encoder representations before the predictor. Both use WS2 or EP during pretraining and a separately fitted frozen-encoder linear probe for evaluation.

Table 1: Fixed experimental configuration. Both pretraining objectives use $\mathcal{L} = (1 - \lambda_{\text{slice}})\mathcal{L}_{\text{task}} + \lambda_{\text{slice}}\mathcal{L}_{\text{slice}}$.

Setting	MNIST	ImageNet-100
Input and latent	$1 \times 28 \times 28$ grayscale, $D = 64$	$3 \times 224 \times 224$ RGB, $D = 256$
Encoder	Conv 3×3, stride 2, channels $1 \to 32 \to 64$, ReLU, FC $3136 \to 64$	ResNet-18 (He et al., 2016) from random initialization, linear head $512 \to 256$
AE decoder	FC $64 \to 64 \times 7 \times 7$, transposed conv $64 \to 32 \to 1$, ReLU/Sigmoid	FC $256 \to 256 \times 7 \times 7$, five transposed-convolution stages and final 3-channel sigmoid output
JEPA views and predictor	Two independent crop/affine views, shared MLP $64 \to 128 \to 64$	Two independent resized-crop, flip, and color-jitter views, shared MLP $256 \to 512 \to 256$
Pretraining	Adam, learning rate 10^{-3}, batch size 128, 30 epochs	Adam, learning rate 10^{-3}, weight decay 10^{-4}, batch size 128, 30 epochs
Latent regularization	EP or calibrated WS2, $\lambda_{\text{slice}} = 0.1$, $K_{\text{adapt}} = 8$, no radial threshold	EP or calibrated WS2, $\lambda_{\text{slice}} = 0.1$, $K_{\text{adapt}} = 8$, no radial threshold
Linear probe	Adam, learning rate 10^{-2}, weight decay 10^{-4}, 20 epochs	Adam, learning rate 10^{-2}, weight decay 10^{-4}, 20 epochs
Evaluation	10 seeds, random and test-latent directions, 10,000 directions per bank	10 seeds, random and test-latent directions, 5,000 directions per bank

and 30 pretraining epochs to keep the loss, slice-construction, and seed study tractable. Frozen probes confirm retained class-relevant signal in every setting. Across all runs, test accuracy spans approximately 85–98% on MNIST and 3–19% on ImageNet-100, above the respective 10% and 1% chance levels. These probe results serve as collapse diagnostics, not as comparisons with fully tuned SSL systems.

Evaluation slice families. Let $Z^{\text{test}} = \{z_n^{\text{test}}\}_{n=1}^{N_{\text{test}}}$ denote frozen encoder representations from the held-out split, either the MNIST test set or the ImageNet-100 validation set. We evaluate these representations using two slice banks. Random directions are sampled uniformly from the unit sphere,

$$\mathcal{S}^{\text{rand}} = \left\{ \frac{g_k}{\|g_k\|_2} \ : \ g_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_D),\ k = 1, \ldots, K_{\text{eval}} \right\}. \tag{20}$$

Test-latent directions are obtained by normalizing all held-out representations,

$$\mathcal{S}^{\text{test}} = \left\{ \frac{z_n^{\text{test}}}{\|z_n^{\text{test}}\|_2} : n = 1, \ldots, N_{\text{test}} \right\}. \tag{21}$$

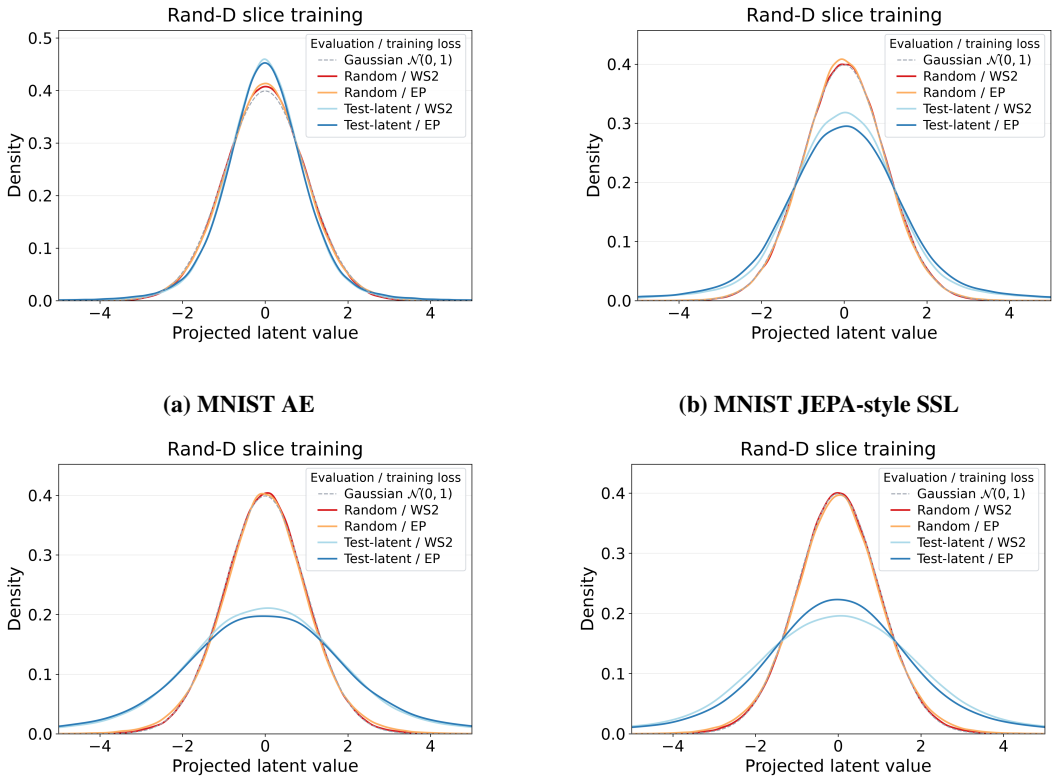
For a metric M, $M(Z^{\text{test}}; \mathcal{S}^a)$ denotes evaluation of the held-out representations using slice bank $a \in \{\text{rand}, \text{test}\}$. Both evaluation banks are constructed after pretraining and held fixed during evaluation. For each direction, one KS statistic is computed from all N_{test} held-out projections,

then these K_{eval} statistics are averaged. Thus, KS_{avg} is a whole-set statistic, not an average of minibatch KS values. During pretraining, random directions are resampled, the rotating frame is updated, and radial-adaptive directions are recomputed from the current latent minibatch at every step, as applicable to each variant.

Although held out from encoder training, the test latents are not independent of their direction bank. The same samples define directions and provide evaluation projections, including each sample's projection onto its own direction. We therefore interpret KS as a directional discrepancy diagnostic, not a calibrated omnibus test of joint Gaussianity. The Wilcoxon tests assess paired seed-level diagnostic differences, not whether the latent distribution is Gaussian.

4.1 Random Directions Can Mask Departures from the Isotropic Gaussian Target

We evaluate held-out test latents using a random bank and the normalized held-out latent vectors as a test-direction bank. Figure 2 visualizes the projected densities. Random-direction evaluation can appear substantially closer to the standard-normal reference than test-latent-direction evaluation, revealing departures from the isotropic Gaussian target along occupied directions. Such discrepancies may reflect mean or covariance mismatch as well as non-Gaussian shape.



(a) MNIST AE **(b) MNIST JEPA-style SSL**

(c) ImageNet-100 AE **(d) ImageNet-100 JEPA-style SSL**

Figure 2: **Random directions can overstate agreement with the Gaussian target.** Parzen densities of held-out projections after Rand-D training with WS2 or EP. Random directions can appear close to $\mathcal{N}(0,1)$, whereas test-latent directions reveal larger departures from this reference. Curves pool sign-symmetrized samples from ten seeds. KS uses the original per-slice projections.

Figure 3 reports paired KS gaps. The Rand-D results quantify the discrepancy across both datasets. Let $\Delta_{\text{dir}} = \text{KS}_{\text{avg}}(\text{test-latent directions}) - \text{KS}_{\text{avg}}(\text{random directions})$ on held-out test latents, where positive values indicate the discrepancy masked by random directions. Across ten seeds, MNIST AE gives $\Delta_{\text{dir}} = 0.0307 \pm 0.0027$ for calibrated WS2 training and 0.0369 ± 0.0035 for EP

training. MNIST JEPA gives 0.0480 ± 0.0028 and 0.0565 ± 0.0049, respectively. The ImageNet-100 gaps are larger, at 0.1482 ± 0.0068 and 0.1701 ± 0.0113 for AE, and 0.1609 ± 0.0126 and 0.1494 ± 0.0068 for JEPA, under WS2 and EP, respectively. Values are mean $\pm$ sample standard deviation across seeds. For Rand-D training, all ten seeds have positive directional gaps in each of the eight dataset, model, and loss combinations, giving an exact two-sided Wilcoxon signed-rank $p = 0.00195$ for every comparison.

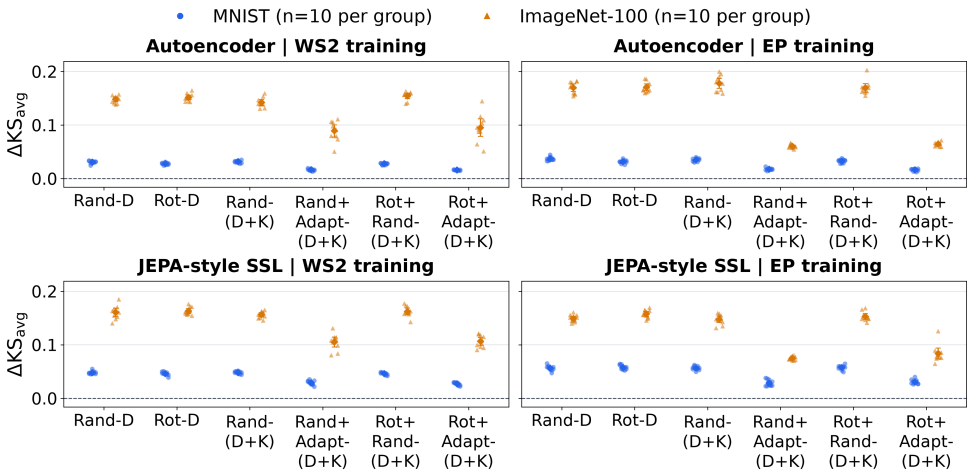


Figure 3: Directional gaps on held-out representations. Points are seeds. Diamonds are means with approximate 95% confidence intervals. Positive $\mathrm{KS}_{\mathrm{avg}}(\text{test-latent}) - \mathrm{KS}_{\mathrm{avg}}(\text{random})$ values indicate larger standard-normal discrepancies under test-latent directions.

4.2 Structured Rotation as a Training-Time Slice Baseline

Before adding data-dependent directions, we compare Rand-D and Rot-D at the same budget of D training slices per batch. The paired test-latent-direction results give no universal ordering. At nominal $\alpha = 0.01$, rotation improves all four MNIST conditions, namely AE with WS2 and EP (both $p = 0.00195$), and JEPA with WS2 ($p = 0.00977$) and EP ($p = 0.00586$). ImageNet-100 JEPA with EP favors random slices ($p = 0.00195$). The remaining three ImageNet-100 conditions show no statistically detectable paired difference at this threshold. Thus, structured rotation is a competitive, reproducible equal-budget baseline rather than a uniformly superior replacement for random sampling. This motivates testing radial adaptation while holding the base schedule fixed.

4.3 Radial-Adaptive Augmentation at a Matched Slice Budget

The $D + K$ ablation uses all four combinations of random or rotating base directions and random or radial-adaptive augmentation. The direct comparisons Rand-(D+K) versus Rand+Adapt-(D+K) and Rot+Rand-(D+K) versus Rot+Adapt-(D+K) isolate adaptive selection while holding the base schedule fixed. Figure 4 reports these two contrasts under test-latent-direction evaluation, the protocol that reveals departures from the isotropic Gaussian target.

Radial-adaptive augmentation lowers mean test-latent-direction $\mathrm{KS}_{\mathrm{avg}}$ in all 16 comparisons across the two base schedules and eight combinations of dataset, model, and loss. Relative reductions range from 2.9–28.0% on MNIST and 17.6–53.1% on ImageNet-100, computed as the decrease in seed-averaged KS divided by the random-augmentation baseline mean. Fifteen comparisons are significant at nominal $\alpha = 0.01$ ($p \leq 0.00391$). With a random base, every condition improves in all ten seeds (all $p = 0.00195$). With a rotating base, seven of eight conditions are significant. The exception is MNIST AE with WS2, where the mean decreases from 0.0507 to 0.0492 but the paired difference is not significant ($p = 0.27539$). The mean directional gap also decreases in all 16 comparisons, although it remains positive after adaptive augmentation.

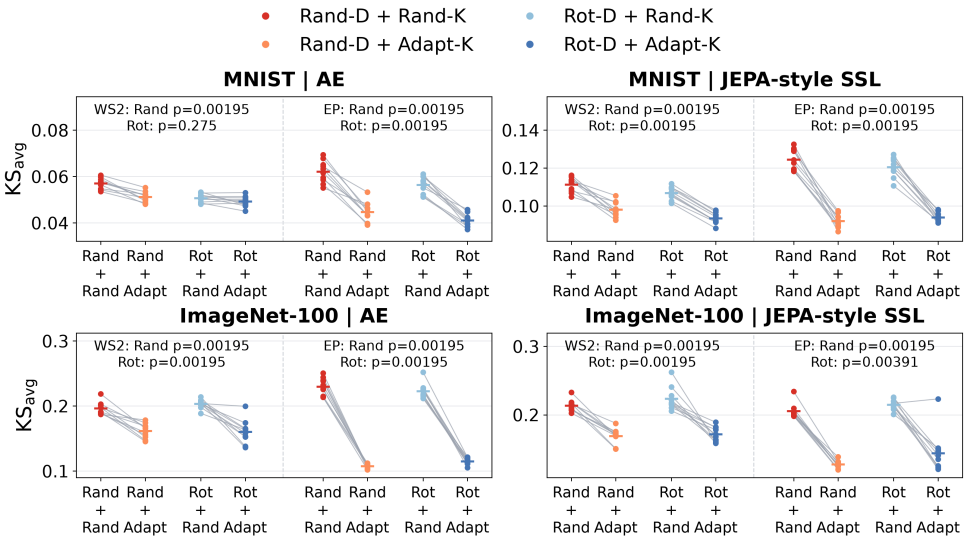


Figure 4: **Factorial radial-adaptive ablation under held-out latent-direction evaluation.** All methods use $D + K$ training directions. Red/orange markers compare random-K and radial-adaptive-K augmentation with a random base schedule. Light/dark blue markers make the corresponding comparison with a rotating base schedule. Gray lines pair seeds and horizontal bars denote means. Lower is better. Displayed p-values are exact paired Wilcoxon tests.

5 Discussion

We studied how slice construction and evaluation directions affect agreement with the isotropic Gaussian target. Random projections can conceal discrepancies along occupied directions. Additional evaluation controls indicate that the gap persists with disjoint samples and can arise from mean or covariance mismatch without non-Gaussian shape.

Several limitations should be considered. The experiments use $K = 8$ without thresholding, leaving budget sensitivity, threshold selection, and comparison with unranked latent-derived augmentation open. Above-chance frozen probes indicate retained class information but do not rule out partial collapse. Fuller SSL studies are needed to assess downstream competitiveness.

At matched $D + K$ budgets, radial-adaptive augmentation improves mean latent-direction KS under WS2 and EP and both base schedules. Additional evaluation controls support these mean improvements. The augmentation reuses minibatch latents to select and normalize radially atypical vectors, requiring no additional learned parameters or inner optimization.

Rotation is competitive but not consistently superior to random sampling. We therefore recommend augmenting either base with a small number of radial-adaptive directions and evaluating representations along both random and latent-derived directions when improved agreement with the isotropic Gaussian target is the goal.

6 Conclusion

Two practical recommendations follow. Evaluate representations along both random and latent-derived directions, and augment training banks with K radial-adaptive minibatch directions. Across AE and JEPA-style learning on MNIST and ImageNet-100 with WS2 and EP, adaptation lowers mean held-out latent-direction KS in all 16 matched-budget comparisons, with 15 significant at nominal $\alpha = 0.01$. The construction requires no additional learned parameters or inner optimization. A rotating base is competitive but not consistently superior to a random base, and either supports adaptive augmentation. The residual directional gap motivates complementary evaluation while the adaptive improvement provides a practical route toward better matching the isotropic Gaussian target.

AI USE STATEMENT

In this work, we primarily used generative AI tools for language editing, including improving the grammar, clarity, and readability of the manuscript. Generative AI tools were also used to assist with drafting code and code skeletons, planning experimental setups, debugging, and formatting figures and tables. All AI-assisted code and experimental scripts were manually reviewed, executed, and checked by the authors. All experiments were initiated and run by the authors, and the resulting outputs and experimental results were manually inspected and verified. Generative AI tools were not used to generate or alter experimental data or results, or to generate or fabricate citations. All AI-assisted text was reviewed and revised by the authors to ensure accuracy and consistency with the methods and results. The authors take full responsibility for the final content of this work, including all text, claims, code, figures, tables, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies latent regularization and representation learning methods on standard benchmark datasets. The experiments use existing MNIST and ImageNet-100 datasets and involve no new data collection, human-participant studies, or safety-critical deployment. The primary risks are indirect and relate to the broader use of representation learning systems in downstream applications, including potential bias inherited from training data and misuse in real-world decision-making settings. We encourage evaluation on diverse datasets and careful consideration of downstream deployment contexts.

REPRODUCIBILITY STATEMENT

The main paper provides mathematical definitions of the training objectives, sliced losses, direction schedules, and adaptive slice-selection procedures. The experimental section specifies datasets, preprocessing, hyperparameters, slice budgets, evaluation protocols, and reporting metrics. Appendix B supports reproducibility with implementation details and PyTorch-style pseudocode for the sliced losses, KS diagnostics, rotating directions, and radial-adaptive augmentation, together with the derivation of the WS2 and EP loss-scale calibration.

REFERENCES

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*, pp. 15619–15629, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Assran_Self-Supervised_Learning_From_Images_With_a_Joint-Embedding_Predictive_Architecture_CVPR_2023_paper.html.

Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics. arXiv preprint arXiv:2511.08544, 2025. URL https://arxiv.org/abs/2511.08544.

Harald Cramér and Herman Wold. Some theorems on distribution functions. *Journal of the London Mathematical Society*, s1-11(4):290–294, 1936. doi: 10.1112/jlms/s1-11.4.290. URL https://londmathsoc.onlinelibrary.wiley.com/doi/10.1112/jlms/s1-11.4.290.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. URL https://www.image-net.org/static_files/papers/imagenet_cvpr09.pdf.

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G. Schwing. Max-sliced Wasserstein distance and its use for GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10648–10656, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Deshpande_Max-Sliced_Wasserstein_Distance_and_Its_Use_for_GANs_CVPR_2019_paper.html.

T. W. Epps and Lawrence B. Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983. doi: 10.1093/biomet/70.3.723. URL https://academic.oup.com/biomet/article-abstract/70/3/723/248040.

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*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Szymon Knop, Przemysław Spurek, Jacek Tabor, Igor Podolak, Marcin Mazur, and Stanisław Jastrzębski. Cramer-Wold auto-encoder. *Journal of Machine Learning Research*, 21(164):1–28, 2020. URL https://www.jmlr.org/papers/v21/19-560.html.

Soheil Kolouri, Phillip E. Pope, Charles E. Martin, and Gustavo K. Rohde. Sliced-Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1xaJn05FQ.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. URL https://leon.bottou.org/papers/lecun-98h.

Xianhang Li, Chen Huang, Chun-Liang Li, Eran Malach, Joshua Susskind, Vimal Thilak, and Etai Littwin. Rethinking JEPA: Compute-efficient video self-supervised learning with frozen teachers. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e6d1d6195f6f3e32a930643e0ef46332-Abstract-Conference.html.

Khai Nguyen, Nhat Ho, Tung Pham, and Hung Bui. Distributional sliced-Wasserstein and applications to generative modeling. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=QYjO70ACDK.

Khai Nguyen, Shujian Zhang, Tam Le, and Nhat Ho. Sliced Wasserstein with random-path projecting directions. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37879–37899. PMLR, 2024. URL https://proceedings.mlr.press/v235/nguyen24l.html.

Khai Nguyen, Hai Nguyen, and Nhat Ho. Fast estimation of Wasserstein distances via regression on sliced Wasserstein distances. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007397.

Moritz Piening and Robert Beinert. Slicing Wasserstein over Wasserstein via functional optimal transport. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007722.

Jean Ponce, Basile Terver, Martial Hebert, and Michael Arbel. Dual perspectives on non-contrastive self-supervised learning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/349a45f211fb1b3850da1ccd829e869e-Abstract-Conference.html.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision – ECCV 2020*, volume 12356 of *Lecture Notes in Computer Science*, pp. 776–794. Springer, 2020. doi: 10.1007/978-3-030-58621-8_45. URL https://www.ecva.net/papers/eccv_2020/papers_ECCV/html/1453_ECCV_2020_paper.php.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

Haiyu Wu, Randall Balestriero, and Morgan Levine. VISReg: Variance-invariance-sketching regularization for JEPA training. arXiv preprint arXiv:2606.02572, 2026. URL https://arxiv.org/abs/2606.02572.

Yipeng Zhang, Hafez Ghaemi, Jungyoon Lee, Shahab Bakhtiari, Eilif B. Muller, and Laurent Charlin. Self-supervised learning from structural invariance. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d26d1923ce22d5afc3ee0db15ef4d900-Abstract-Conference.html.

A Additional Rotation Ablation

Rand-D and Rot-D both use exactly D training directions per batch. They differ only in whether directions are sampled independently or supplied by the rotating orthogonal frame. For each dataset, model, and loss, we pair runs by seed and compare $\mathrm{KS}_{\mathrm{avg}}$ on held-out test latents evaluated with held-out test-latent directions. Figure 5 reports exact two-sided Wilcoxon signed-rank tests of the paired Rot-D minus Rand-D differences.

The equal-budget comparison does not support a universal ranking. On MNIST AE, Rot-D improves test-latent-direction $\mathrm{KS}_{\mathrm{avg}}$ for WS2 and EP (mean Rot-D minus Rand-D differences $\Delta = -0.0064$ and -0.0087, both $p = 0.00195$). MNIST JEPA also favors Rot-D for WS2 ($\Delta = -0.0052$, $p = 0.00977$) and EP ($\Delta = -0.0050$, $p = 0.00586$). ImageNet-100 JEPA with EP favors Rand-D ($\Delta = 0.0146$, $p = 0.00195$). The ImageNet-100 AE comparisons ($p = 0.37500$ for WS2 and $p = 0.76953$ for EP) and JEPA with WS2 ($p = 0.13086$) are not significant at nominal $\alpha = 0.01$. Thus, four conditions favor rotation, one favors random sampling, and three show no detectable paired difference. The latter do not establish equivalence.

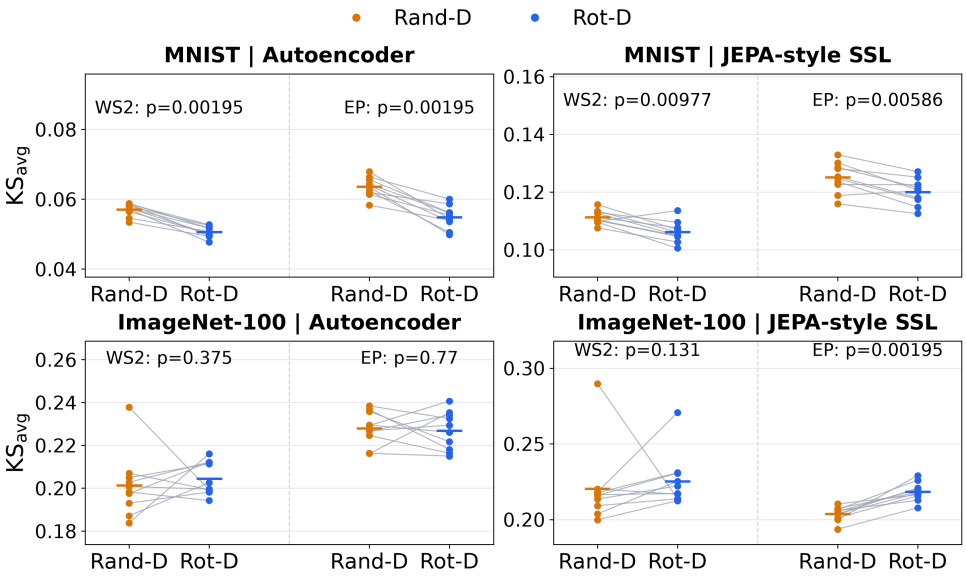


Figure 5: **Equal-budget rotation ablation.** Rand-D and Rot-D each use D training directions per batch. Within each WS2 or EP group, gray lines pair the same seed, colored points are seed-level whole-test-set latent-direction $\mathrm{KS}_{\mathrm{avg}}$ values, and horizontal bars denote group means. Lower is better. The displayed p-values are exact two-sided Wilcoxon signed-rank tests for paired Rot-D versus Rand-D differences across ten seeds.

B Implementation Details and Pseudocode

This appendix gives concise PyTorch-style implementations of the losses, diagnostics, and slice-generation procedures defined in the main paper. Each row of Z is a latent vector, and $S \in \mathbb{R}^{K \times D}$ contains unit slice directions as rows. In the loss and slice-generation listings, `B = Z.size(0)` is the effective latent-batch size, equal to the input batch size for AE and twice that size for two-view JEPA. The KS listing instead receives all held-out latents.

Additional experimental settings. Images are converted to float32 tensors in $[0, 1]$, without channel-wise mean or variance normalization. MNIST AE training, probing, and plain-latent evaluation use unaugmented images. Each MNIST JEPA view independently applies a 28×28 random crop after two-pixel padding, followed by a random affine transform with rotation in $[-15°, 15°]$, translation up to 10% in each axis, and scale in $[0.9, 1.1]$. ImageNet-100 AE training and probe fitting use `RandomResizedCrop`(224) with area fraction $[0.8, 1.0]$ and horizontal flips with probability 0.5. Each ImageNet-100 JEPA view instead uses crop area fraction $[0.5, 1.0]$, the same flip probability, and `ColorJitter` strengths $(0.4, 0.4, 0.2, 0.1)$ for brightness, contrast, saturation, and hue. Remaining transform arguments use torchvision defaults. ImageNet-100 plain-latent and probe evaluation resize the shorter edge to 256 and center-crop to 224×224. The MNIST encoder's two convolutions use padding 1. Both JEPA predictors are MLPs with a ReLU between two linear layers, widths $D \to 2D \to D$ and no output activation. The MNIST decoder reshapes its linear output to $64 \times 7 \times 7$, then uses 4×4 transposed convolutions with stride 2 and padding 1, channels $64 \to 32 \to 1$, an intermediate ReLU, and a final sigmoid. The ImageNet-100 decoder reshapes to $256 \times 7 \times 7$, then applies five 4×4 transposed convolutions with stride 2, padding 1, channels $256 \to 128 \to 64 \to 32 \to 16 \to 8$, and ReLU after each, followed by a 3×3 convolution from 8 to 3 channels with padding 1 and a sigmoid. Reconstruction MSE averages over samples, channels, and pixels. Learning rates are constant, and MNIST pretraining uses no weight decay. Pretraining drops incomplete minibatches. Probe fitting and plain-latent evaluation retain them. During probing, the encoder is frozen in evaluation mode, including its batch-normalization statistics.

B.1 WS2 and EP Loss-Scale Calibration

For $X_\sigma \sim \mathcal{N}(0, \sigma^2)$, let $q_i = \Phi^{-1}((i - \frac{1}{2})/B_{\text{eff}})$. The deterministic sorted sample is $p_i(\sigma) = \sigma q_i$, so

$$\mathcal{L}^{\text{grid}}_{\text{WS2}}(\sigma) = \frac{1}{B_{\text{eff}}} \sum_{i=1}^{B_{\text{eff}}} (\sigma q_i - q_i)^2 = (\sigma - 1)^2 \frac{1}{B_{\text{eff}}} \sum_{i=1}^{B_{\text{eff}}} q_i^2. \tag{22}$$

The population characteristic function is $\varphi_\sigma(\tau) = \exp(-\sigma^2\tau^2/2)$. We evaluate its EP discrepancy by the same discrete rule used in training, on $\mathcal{T} = \text{linspace}(-5, 5, 17)$,

$$\mathcal{L}^{\text{pop}}_{\text{EP}}(\sigma) = B_{\text{eff}} \ \text{Trapz}_{\tau \in \mathcal{T}} \left[\left(e^{-\sigma^2\tau^2/2} - e^{-\tau^2/2} \right)^2 e^{-\tau^2/2} \right]. \tag{23}$$

The applied scale is

$$c_{\text{WS2}\to\text{EP}} = \frac{\mathcal{L}^{\text{pop}}_{\text{EP}}(0.5)}{\mathcal{L}^{\text{grid}}_{\text{WS2}}(0.5)}. \tag{24}$$

No Monte Carlo samples are used. We set $B_{\text{eff}} = B$ for AE and $B_{\text{eff}} = 2B$ for JEPA. With input batch size $B = 128$, the resulting scales are approximately 78.443176 for AE and 156.108690 for JEPA. This calibration matches the losses at the anchor, not over the entire variance path.

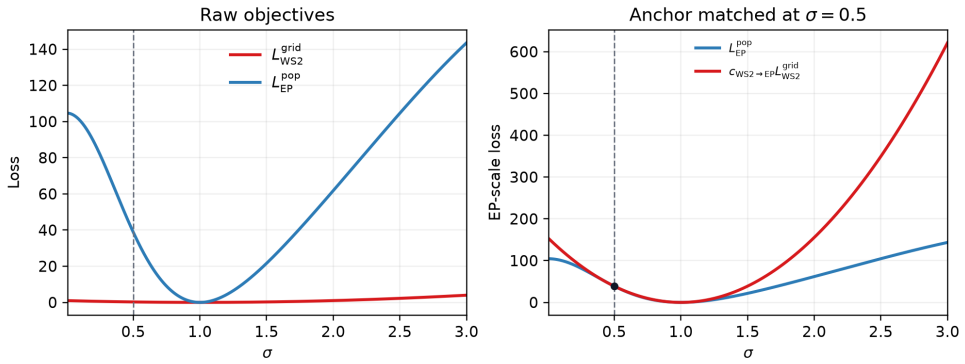


Figure 6: Deterministic WS2-grid and population EP losses along $X_\sigma \sim \mathcal{N}(0, \sigma^2)$, plotted for $B_{\text{eff}} = 256$ (JEPA). The factor $c_{\text{WS2}\to\text{EP}} \approx 156.108690$ in Eq. (24) matches the losses at $\sigma = 0.5$. AE uses $B_{\text{eff}} = 128$ and its separately computed scale.

Deterministic WS2 and EP loss-scale calibration. The following listing implements Eq. (24) in double precision. Compute `c = WS2ToEPScale(B_eff)` once for the effective batch size (B for AE, $2B$ for JEPA), then use `c * WS2(Z, S)` as the sliced training loss. This matches loss values at the anchor, not their gradients.

```
import torch

def WS2ToEPScale(B_eff):
    # INPUT: effective latent-batch size; OUTPUT: scalar scale
    sigma = 0.5
    u = (torch.arange(B_eff, dtype=torch.float64) + 0.5) / B_eff
    q = 2.0 ** 0.5 * torch.erfinv(2.0 * u - 1.0)
    ws2_grid = ((sigma - 1.0) * q).square().mean()
    t = torch.linspace(-5, 5, 17, dtype=torch.float64)
    phi = torch.exp(-0.5 * t.square())
    phi_sigma = torch.exp(-0.5 * sigma ** 2 * t.square())
    err = (phi_sigma - phi).square() * phi
    ep_pop = B_eff * torch.trapz(err, t)
    return (ep_pop / ws2_grid).item()
```

B.2 Batched Sliced Losses and Diagnostics

The three shared helpers below support the sliced losses, diagnostics, and rotation procedure. All listings assume `import math` and `import torch`.

Gaussian reference quantiles. This helper implements Eq. (1), using $\Phi^{-1}(u) = \sqrt{2}\,\mathrm{erf}^{-1}(2u - 1)$.

```
def gaussian_reference_quantiles(B, device, dtype):
    # INPUT: sample count, device, dtype; OUTPUT: quantiles (B,)
    u = (
        torch.arange(1, B + 1, device=device, dtype=dtype) - 0.5
    ) / B
    return u.new_tensor(2.0).sqrt() * torch.erfinv(2.0 * u - 1.0)
```

Standard-normal CDF. This helper evaluates the target CDF Φ in Eq. (6) as $\Phi(x) = \frac{1}{2}[1 + \mathrm{erf}(x/\sqrt{2})]$ elementwise, preserving the input tensor's shape, device, and dtype.

```
def normal_cdf(x):
    # INPUT: floating-point tensor; OUTPUT: CDF values, same shape
    return 0.5 * (1.0 + torch.erf(x / x.new_tensor(2.0).sqrt()))
```

Deterministic rotation angle. This helper implements Eq. (10) for batch t and pair m, with $M = M_t > 0$ and zero-based indices $t \geq 0$ and $0 \leq m < M$. It is used by the rotating-frame procedure below.

```
def theta_tm(t, m, M):
    # INPUT: batch index, pair index, pair count; OUTPUT: radians
    alpha = (math.sqrt(5.0) - 1.0) / 2.0
    phase = ((t + m / M) * alpha) % 1.0
    return math.pi / 4.0 + (math.pi / 8.0) * (2.0 * phase - 1.0)
```

B.2.1 Sliced Wasserstein-2

The empirical projections are sorted once along the batch dimension and compared with fixed standard-normal quantiles, implementing Eq. (2). This function returns raw WS2. Training multiplies it by the calibration factor above.

```
def WS2(Z, S):
    # Z: (B, D), S: (K, D)
    B = Z.size(0)
    q = gaussian_reference_quantiles(
        B, device=Z.device, dtype=Z.dtype
    )
    U = Z @ S.T
    P = U.sort(dim=0).values
    return ((P - q[:, None]) ** 2).mean()
```

B.2.2 Sliced Epps–Pulley

The empirical characteristic function is evaluated on a fixed symmetric frequency grid and integrated by the trapezoidal rule in Eq. (5).

```
def EP(Z, S):
    # Z: (B, D), S: (K, D)
    B = Z.size(0)
    t = torch.linspace(-5, 5, 17, device=Z.device, dtype=Z.dtype)
    phi = torch.exp(-0.5 * t ** 2)
    U = Z @ S.T
    x_t = U.unsqueeze(2) * t
    ecf = (1j * x_t).exp().mean(0)
    err = (ecf - phi).abs().square().mul(phi)
    return torch.trapz(err, t, dim=1).mean() * B
```

B.2.3 Sliced Kolmogorov–Smirnov Diagnostics

Each per-slice KS statistic uses all N_{test} held-out projections. Only directions may be chunked for memory efficiency. In addition to the main-paper mean, we compute two optional robustness summaries, namely the worst-direction discrepancy

$$\mathrm{KS}_{\max} = \max_{1 \leq k \leq K_{\text{eval}}} D_k, \tag{25}$$

and the empirical pass rate under the asymptotic one-sample KS threshold,

$$\mathrm{KSPR}_\alpha = \frac{1}{K_{\text{eval}}} \sum_{k=1}^{K_{\text{eval}}} \mathbb{I}\left[D_k \leq \frac{c_\alpha}{\sqrt{N_{\text{test}}}} \right], \qquad \alpha \in \{0.05, 0.10\}, \tag{26}$$

where $c_{0.05} = 1.3581$ and $c_{0.10} = 1.2238$. $\mathrm{KS}_{\max}$ highlights a single poorly matched direction, whereas KSPR_α reports the fraction below the threshold. For data-dependent test-latent directions, these fractions are descriptive, not calibrated normality-test pass rates.

Sliced diagnostics. The procedure below calls the CDF helper on sorted projections and returns $(\mathrm{KS}_{\mathrm{avg}}, \mathrm{KS}_{\max}, \mathrm{KSPR}_{0.05}, \mathrm{KSPR}_{0.10})$.

```
def KS(Z, S):
    # Z: all held-out latents (N, D), S: (K, D)
    N = Z.size(0)
    U = Z @ S.T
    P = U.sort(dim=0).values
    F = normal_cdf(P)
    i = torch.arange(
        1, N + 1, device=Z.device, dtype=Z.dtype
    )[:, None]
    d_plus = i / N - F
    d_minus = F - (i - 1.0) / N
    D = torch.maximum(d_plus.max(0).values, d_minus.max(0).values)
    crit05 = 1.358101515740619 / math.sqrt(N)
    crit10 = 1.2238478702170825 / math.sqrt(N)
    return (
        D.mean(), D.max(),
        (D <= crit05).float().mean(),
        (D <= crit10).float().mean(),
    )
```

B.3 Structured and Adaptive Slice Generation

The following procedures construct the training directions used by the sliced losses. We first describe the rotating orthogonal frame, then the radial selection rule that augments either a random or rotating base with minibatch directions. The final listings provide the coordinate pairing and Givens rotation helpers. Each returned slice bank stores unit directions as rows.

B.3.1 Rotating Orthogonal Frame

Starting from $Q_0 = I_D$, the routine takes `Q_prev`$= Q_t$ as the state before batch t. The helper `theta_tm` evaluates Eq. (10), and the Givens row updates implement Eq. (11). The routine returns the updated state Q_{t+1} for the next batch and the slice matrix $Q_{t+1}^{\top}$. Its rows are the column directions specified in Eq. (12), so the current batch uses the updated frame.

```
def RotatingSlices(D, t, Q_prev):
    # Q_prev: (D, D)
    pairs = round_robin_pairs_for_t(D, t)
    Q = Q_prev.clone()
    for m, (i, j) in enumerate(pairs):
        theta = theta_tm(t, m, len(pairs))
        Q = apply_givens_rows(Q, i, j, theta)
    return Q.detach(), Q.T.detach()
```

B.3.2 RADIAL-ADAPTIVE AUGMENTATION

The score in Eq. (13) ranks minibatch latents by radial atypicality. Following Eq. (14), up to K_{adapt} samples with the largest scores are normalized and appended to the random or rotating base directions. If enabled, the optional threshold τ first restricts eligibility to $a_n \geq \tau$ and can reduce the number selected. Figure 7 illustrates selection without thresholding.

```
@torch.no_grad()
def RadialAdaptiveSlices(Z, K_adapt, tau, eps=1e-12):
    # Z: (B, D)
    B, D = Z.shape
    r2 = Z.pow(2).sum(dim=1).clamp_min(eps)
    cdf = torch.special.gammainc(
        torch.tensor(D / 2.0, device=Z.device, dtype=Z.dtype), r2 / 2.0
    ).clamp(eps, 1.0 - eps)
    scores = -(2.0 * torch.minimum(cdf, 1.0 - cdf) + eps).log()
    order = torch.argsort(scores, descending=True)
    if tau is not None:
        order = order[scores[order] >= tau]
    S_adapt = Z[order[:K_adapt]]
    return S_adapt / S_adapt.norm(dim=1, keepdim=True).clamp_min(eps)
```

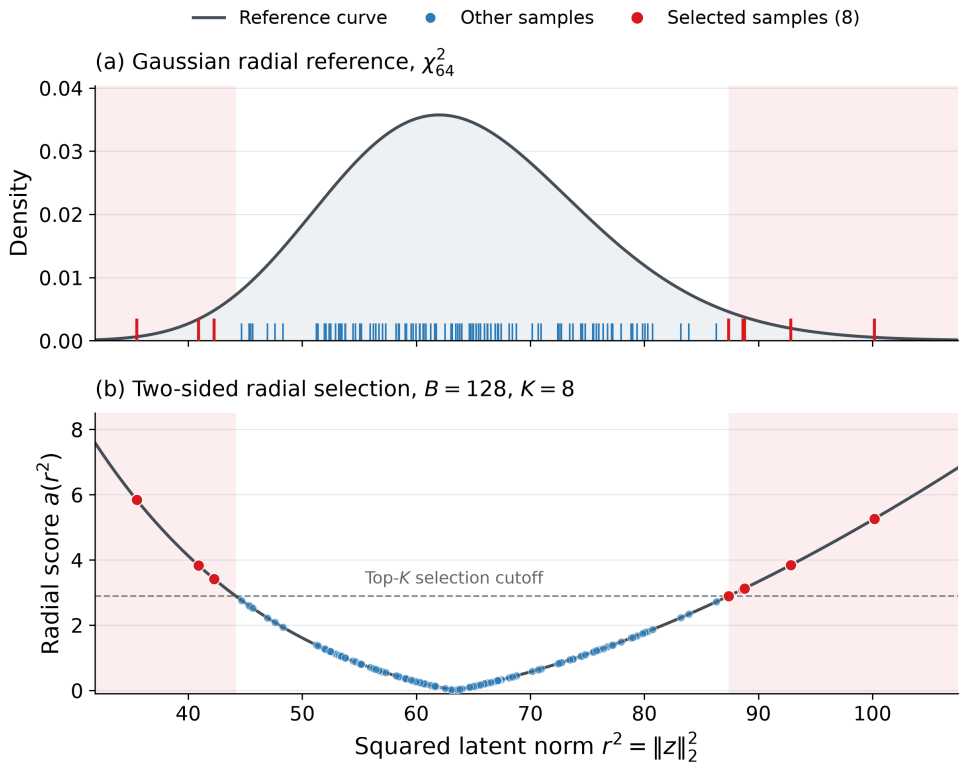


Figure 7: **Radial-adaptive selection from an illustrative Gaussian batch.** A synthetic batch of $B = 128$ vectors is drawn from $\mathcal{N}(0, I_{64})$ using seed 0. The upper panel shows the theoretical χ^2_{64} density of $r^2 = \|z\|_2^2$, with batch values marked along the baseline. The lower panel shows their radial scores. Red marks the $K = K_{\text{adapt}} = 8$ selected samples, whose normalized vectors provide adaptive directions. The dashed line is the smallest selected score, and shading indicates scores at or above this cutoff. No fixed threshold τ is applied. Selection identifies relative radial atypicality even under the exact Gaussian target, not evidence of a Gaussianity failure. This is a synthetic illustration, not a trained-model result.

B.3.3 Rotation Helpers

The following helpers define the round-robin pairing schedule and the in-place row update used by the rotating frame.

```python
def round_robin_pairs_for_t(D, t):
    if D < 2:
        return []
    D_aug = D + (D % 2)
    dummy = D if D % 2 else None
    others = list(range(1, D_aug))
    t = t % (D_aug - 1)
    others = others[-t:] + others[:-t]
    pairs = [(0, others[-1])]
    pairs += [
        (others[m], others[-m - 2])
        for m in range(D_aug // 2 - 1)
    ]
    return [p for p in pairs if dummy not in p]
```

```python
def apply_givens_rows(Q, i, j, theta):
    theta = torch.as_tensor(theta, device=Q.device, dtype=Q.dtype)
    c, s = torch.cos(theta), torch.sin(theta)
    Qi, Qj = Q[i].clone(), Q[j].clone()
    Q[i] = c * Qi - s * Qj
    Q[j] = s * Qi + c * Qj
    return Q
```

B.4 Projected-Density Visualization

For Figure 2, each seed and evaluation bank supplies projections of 317 randomly selected held-out latents onto 316 randomly selected directions, sampled without replacement within each set. The resulting 100,172 values are paired with their negatives and pooled across ten seeds. Each curve uses a random subsample of 200,000 pooled values and a Gaussian-kernel Parzen estimate with bandwidth $h = \max(1.06\,\widehat{\sigma}M^{-1/5}, 10^{-3})$, where M and $\widehat{\sigma}$ are the subsample size and sample standard deviation. Densities are evaluated at 501 equally spaced points in $[-5, 5]$. Subsampling uses fixed random seeds. Sign symmetrization and pooling apply only to visualization. KS is computed separately along each original direction using all held-out projections, without these operations.

C SUPPLEMENTARY EVALUATIONS OF FROZEN REPRESENTATIONS

The supplementary controls reinforce the main directional-evaluation and adaptive-augmentation findings while sharpening their interpretation. The KS gap between random and latent directions reported in Section 4.1 remains positive for all 480 models when direction-source and evaluation samples are separated. The 480 trained model instances comprise two datasets (MNIST and ImageNet-100), two objectives (AE and JEPA-style SSL), two regularization losses (WS2 and EP), six slice constructions (Rand-D, Rot-D, Rand-(D+K), Rand+Adapt-(D+K), Rot+Rand-(D+K), and Rot+Adapt-(D+K)), and ten seeds per configuration. Under this sample-disjoint evaluation, radial-adaptive augmentation also lowers mean latent-direction KS in all 16 matched comparisons, supporting the result in Section 4.3. Thus, neither finding is explained solely by evaluating source samples along their own directions.

The Gaussian controls clarify what the gap measures. Samples from the exact isotropic target produce essentially no directional gap, whereas Gaussians fitted to the learned means and covariances produce gaps at least as large as the measured ones in six of eight baseline conditions. The main-paper finding is therefore a discrepancy from the *isotropic standard-Gaussian target*, not necessarily non-Gaussian distributional shape. MNIST AE is the exception. Its fitted Gaussian controls do not reproduce the observed gap.

All analyses reuse frozen representations, without retraining or replacing original metrics. Sample-disjoint comparisons cover all 480 models. Gaussian controls use the 80 Rand-D models and independent simulations. KS uses every representation in the specified evaluation set, with directions chunked for memory efficiency. Fold results are averaged within each training seed before inference.

C.1 SAMPLE-DISJOINT DIRECTIONAL EVALUATION

Two controls separate the examples defining directions from those being projected. First, the saved class-balanced training-direction bank is evaluated on the full held-out set. This bank is formed by normalizing 1,000 training representations per class on MNIST and 50 per class on ImageNet-100, giving the same direction counts as the original evaluation banks. Its source examples are disjoint from the test examples, although they were used to train the encoder. This control therefore provides sample separation, not independence from model fitting.

Second, the held-out representations are randomly partitioned into two equally sized subsets A and B. A bank of 512 normalized representations from A and a matched bank of 512 independent uniform-sphere directions are evaluated on every representation in B. The roles are then reversed, and the two fold-level statistics are averaged. Thus, the evaluation sample count per fold is 5,000 for MNIST and 2,500 for ImageNet-100. Partitions and direction-sampling seeds are fixed and shared across methods within each training seed. Because cross-fitting changes both the evaluation sample count and direction count, its absolute KS values need not equal the full-set values. The relevant comparisons use matched counts within each protocol.

Table 2: Rand-D directional gaps (latent-direction minus random-direction $\mathrm{KS}_{\mathrm{avg}}$). Values are mean $\pm$ sample SD over ten training seeds. Full-set protocols use the original banks. Cross-fitting uses 512 directions per bank and all observations in each evaluation half.

Condition	Original test bank	Train-source bank	Cross-fitted bank
MNIST AE / WS2	0.0307 ± 0.0027	0.0313 ± 0.0026	0.0298 ± 0.0025
MNIST AE / EP	0.0369 ± 0.0035	0.0376 ± 0.0033	0.0360 ± 0.0032
MNIST JEPA / WS2	0.0480 ± 0.0028	0.0480 ± 0.0030	0.0467 ± 0.0036
MNIST JEPA / EP	0.0565 ± 0.0049	0.0556 ± 0.0048	0.0555 ± 0.0054
ImageNet-100 AE / WS2	0.1482 ± 0.0068	0.1474 ± 0.0069	0.1477 ± 0.0066
ImageNet-100 AE / EP	0.1701 ± 0.0113	0.1699 ± 0.0109	0.1688 ± 0.0116
ImageNet-100 JEPA / WS2	0.1609 ± 0.0126	0.1602 ± 0.0121	0.1595 ± 0.0125
ImageNet-100 JEPA / EP	0.1494 ± 0.0068	0.1485 ± 0.0065	0.1489 ± 0.0067

The gap persists after separating the direction-source and evaluation samples. For Rand-D, the cross-fitted mean gap ranges from 0.0298 to 0.0555 on MNIST and from 0.1477 to 0.1688 on ImageNet-100, close to the original full-set gaps. Across all six training constructions, the seed-level cross-fitted gap is positive for all 480 models. The training-source bank gives a similar pattern.

Thus, the observed directional discrepancy is not explained by the direct reuse of a source sample in its own evaluation projection.

Bound on the direct self-projection effect. For a fixed direction obtained from sample i, let $\widehat{F}_N$ be the projected empirical CDF including that sample and $\widehat{F}_{-i}$ the CDF after removing only that sample, without changing the direction. Since $\widehat{F}_N = (1 - 1/N)\widehat{F}_{-i} + \mathbb{I}[u_i \leq \cdot]/N$, the corresponding KS statistics satisfy

$$|D_N - D_{-i}| \leq \left\|\widehat{F}_N - \widehat{F}_{-i}\right\|_\infty \leq \frac{1}{N}. \tag{27}$$

The same bound holds after averaging across fixed directions. It is 0.0001 for MNIST and 0.0002 for ImageNet-100. This bounds the direct contribution of each source sample to its own slice, not every possible effect of data-dependent direction construction.

Table 3: Adaptive-minus-random augmentation contrasts at fixed base and $D + K$ budget. Negative differences favor adaptation. The last two columns test cross-fitted latent-direction KS using exact paired signed-rank tests, before and after Holm correction across the 16 contrasts.

Condition	Base	Δ test	Δ train-src	Δ cross-fit	p	p_{Holm}
MNIST AE / WS2	Rand	-0.0058	-0.0049	-0.0059	0.00195	0.03125
MNIST AE / WS2	Rot	-0.0015	-0.0005	-0.0015	0.16016	0.16016
MNIST AE / EP	Rand	-0.0174	-0.0174	-0.0173	0.00195	0.03125
MNIST AE / EP	Rot	-0.0153	-0.0150	-0.0152	0.00195	0.03125
MNIST JEPA / WS2	Rand	-0.0132	-0.0124	-0.0131	0.00195	0.03125
MNIST JEPA / WS2	Rot	-0.0134	-0.0128	-0.0135	0.00195	0.03125
MNIST JEPA / EP	Rand	-0.0324	-0.0316	-0.0325	0.00195	0.03125
MNIST JEPA / EP	Rot	-0.0266	-0.0267	-0.0264	0.00195	0.03125
ImageNet-100 AE / WS2	Rand	-0.0346	-0.0344	-0.0351	0.00195	0.03125
ImageNet-100 AE / WS2	Rot	-0.0429	-0.0437	-0.0421	0.00195	0.03125
ImageNet-100 AE / EP	Rand	-0.1222	-0.1224	-0.1222	0.00195	0.03125
ImageNet-100 AE / EP	Rot	-0.1080	-0.1081	-0.1086	0.00195	0.03125
ImageNet-100 JEPA / WS2	Rand	-0.0446	-0.0456	-0.0443	0.00195	0.03125
ImageNet-100 JEPA / WS2	Rot	-0.0519	-0.0518	-0.0512	0.00195	0.03125
ImageNet-100 JEPA / EP	Rand	-0.0780	-0.0791	-0.0780	0.00195	0.03125
ImageNet-100 JEPA / EP	Rot	-0.0709	-0.0717	-0.0708	0.00391	0.03125

Paired comparisons use exact two-sided Wilcoxon signed-rank tests across ten seed-level differences, under the usual symmetry assumption, with Holm correction across the 16 matched-base contrasts. These are post-hoc robustness checks, not independent confirmatory training experiments.

The adaptive result also persists under both sample-disjoint controls. All 16 mean adaptive-minus-random augmentation differences remain negative. Under cross-fitting, 15 comparisons are significant at nominal $\alpha = 0.01$. After Holm correction across the 16 contrasts, these 15 have $p_{\mathrm{Holm}} = 0.03125$, meeting a family-wise 0.05 criterion but not 0.01. The exception remains the rotating-base MNIST AE/WS2 contrast ($p = 0.16016$ under cross-fitting). These results retain the direction and broad consistency of the adaptive effect without relying on the original self-containing bank.

C.2 Gaussian Reference Controls

The first reference consists of ten independent draws from $\mathcal{N}(0, I_D)$ for each dataset-sized configuration, using $(N, D) = (10000, 64)$ and $(5000, 256)$. Each draw is evaluated with 512 random directions and 512 directions obtained from the simulated samples themselves, and with the same two-fold cross-fitting procedure. Finite-sample KS values are not expected to vanish even under the exact target distribution. These repetitions are simulation replicates, not additional training seeds.

Table 4: Isotropic Gaussian controls, mean ± SD over ten independent simulations per dataset-sized configuration. These are simulation replicates, not trained-model seeds.

Configuration	Latent KS	Random KS	Full-set gap	Cross-fitted gap
MNIST-sized	0.00870 ± 0.00029	0.00868 ± 0.00027	0.00002 ± 0.00020	-0.00001 ± 0.00011
ImageNet-sized	0.01220 ± 0.00014	0.01218 ± 0.00010	0.00003 ± 0.00011	0.00002 ± 0.00022

The isotropic controls produce mean full-set gaps of approximately 1.8×10^{-5} and 2.8×10^{-5} for the MNIST-sized and ImageNet-sized configurations, respectively. Cross-fitted mean gaps are likewise close to zero. The substantially larger learned-representation gaps are therefore not reproduced by this finite-sample evaluation procedure under the exact isotropic Gaussian reference.

The second reference isolates the explanatory role of first and second moments. For each Rand-D model, the mean $\widehat{\mu}$ and sample covariance $\widehat{\Sigma}$ are estimated from all frozen test latents. An independent synthetic sample of the same size is then drawn from $\mathcal{N}(\widehat{\mu}, \widehat{\Sigma})$ and evaluated with the same 512-direction protocol. The measured latent sample is also reevaluated with 512 directions for this comparison. These fitted references are descriptive controls, not calibrated Gaussianity tests or independent estimates of generalization performance. In particular, a gap reproduced by a moment-matched Gaussian cannot by itself establish higher-order non-Gaussianity. It may reflect mismatch to the *isotropic standard-normal target*.

Table 5: Descriptive moment-matched Gaussian controls for Rand-D. Both measured and synthetic gaps use 512 directions per bank and the full held-out sample count. Mean and covariance are fitted to each frozen test representation set. Effective rank is $(\operatorname{tr}\widehat{\Sigma})^2/\operatorname{tr}(\widehat{\Sigma}^2)$, averaged across seeds.

Condition	Measured gap	Matched-Gaussian gap	$\|\hat{\mu}\|$	Eff. rank
MNIST AE / WS2	0.0310 ± 0.0027	0.0002 ± 0.0023	0.409	59.9
MNIST AE / EP	0.0369 ± 0.0032	0.0059 ± 0.0054	0.377	58.2
MNIST JEPA / WS2	0.0479 ± 0.0041	0.0724 ± 0.0048	1.058	28.1
MNIST JEPA / EP	0.0564 ± 0.0051	0.0837 ± 0.0042	1.080	23.5
ImageNet-100 AE / WS2	0.1482 ± 0.0081	0.1894 ± 0.0208	1.675	43.6
ImageNet-100 AE / EP	0.1699 ± 0.0113	0.2168 ± 0.0112	1.844	34.2
ImageNet-100 JEPA / WS2	0.1614 ± 0.0135	0.2085 ± 0.0267	1.908	36.7
ImageNet-100 JEPA / EP	0.1495 ± 0.0068	0.2068 ± 0.0123	1.707	38.7

The moment-matched controls give a different result. They produce gaps at least as large as the measured gaps in six of eight conditions, including both MNIST JEPA conditions and all four ImageNet-100 conditions. Their magnitude is much smaller for MNIST AE (0.0002 and 0.0059, versus measured gaps of 0.0310 and 0.0369 with the matched 512-direction protocol). Thus, first- and second-moment structure can generate gaps of the observed size in several regimes, but does not reproduce the MNIST AE result in this control. At any fixed direction s, a Gaussian with moments (μ, Σ) projects to $\mathcal{N}(s^\top\mu, s^\top\Sigma s)$, which need not be standard normal. The directional gap is consequently evidence of failure to match the isotropic target, not by itself proof that projection shapes are non-Gaussian after location and scale are accounted for.