CLOUDSPLAT: Distributional Contrastive Learning for Tabular Data

Amir Raza [1] **Mayank Jauhari** [1] **Vikash Sharma** [1] **Vipul Joshi** [1] **Anurag Tripathi** [1] **Christos Faloutsos** [2]

Abstract

Evaluating whether a learned encoder separates classes well currently requires training a downstream classifier, an expensive, protocol-dependent step that conflates representation quality with classifier choice. We propose the CLOUDSPLAT Separability Index (CSSI), a differentiable two-sample test statistic that answers this question directly: it tests whether the distributions of within-class and between-class embedding distances are separable. CSSI sorts both distance collections and compares them position-by-position via a penalized 1D Wasserstein distance with a generalized p-mean: at p=1 it reduces to a standard average-difference test (analogous to a t-statistic on sorted distances), while $p \to \infty$ focuses on the single worst-separated quantile position (the W_∞ distance on sorted distances). Unlike kNN accuracy or silhouette (which are non-differentiable), CSSI is differentiable, enabling its direct use as a training loss. We demonstrate this bridge in two roles: (a) **As a training loss:** plugging CSSI into a contrastive learning framework (CLOUDSPLAT) yields state-of-the-art tabular embeddings, winning on 6/7 datasets at 5% labels against five loss baselines, 7/11 benchmarks against SCARF, and achieving rank 1.9 among six SSL methods, all with a frozen encoder; (b) **As a probe-free diagnostic:** CSSI achieves Spearman $\rho \in [+0.73, +0.92]$ with linear-probe accuracy across three foundation-model architectures on 19 datasets, correctly identifies separability on non-convex geometries where centroid-based metrics fail, and (unlike all competing diagnostics) can be directly optimized. The p parameter provides explicit sensitivity control: low p tests average separation, high p tests worst-case separation, enabling practitioners to tune the test's power profile to their application.

[1]Amazon, India [2]Amazon, USA. Correspondence to: Amir Raza <amirraz@amazon.com>.

ICML 2026 Workshop on Hypothesis Testing, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

1. Introduction

How should we test whether a learned embedding separates classes? Standard metrics (kNN accuracy, silhouette score, centroid-based indices) are either non-differentiable or assume convex cluster geometry, making them unsuitable as training objectives. We propose framing embedding evaluation as a *two-sample test*: given the empirical distributions of within-class (positive) and between-class (negative) pairwise distances, test H_0: both distributions are identical against H_1: negatives are stochastically larger than positives.[1]

Our test statistic, the CLOUDSPLAT Separability Index (CSSI), sorts both distance collections and compares them position-by-position via a penalized Wasserstein distance with a generalized p-mean. The key insight is that this statistic is *differentiable with respect to the embeddings*: sorting is piecewise linear, and the softplus margin penalty is smooth. This enables a direct bridge between *testing* (evaluating embedding quality post hoc) and *optimization* (using the same functional as a contrastive training loss).

The p parameter interpolates between classical test statistics: p=1 computes the mean signed difference across all quantile positions (analogous to a paired t-test on sorted distances); $p \to \infty$ focuses on the single worst-separated quantile position (the W_∞ distance between sorted distributions). Intermediate $p \in [20, 100]$ provides a practical sweet spot, concentrating gradient on the hardest negatives without the instability of low-temperature InfoNCE (Appendix C).

A fundamental challenge when optimizing any distributional test statistic as a loss is preventing representation collapse (Bardes et al., 2021; Hollmann et al., 2025): the encoder can trivially maximize separation by mapping all points to a single direction. Existing methods can suffer from such collapse (Figure 1), producing degenerate similarity distributions that render the test statistic meaningless.

Informal Problem 1. (Differentiable separability testing)

- **Given** pairwise distance distributions $\mathcal{D}_{\text{pos}}$ (within-class) and $\mathcal{D}_{\text{neg}}$ (between-class) from an embedding $f_\theta : \mathbb{R}^d \to \mathbb{S}^{d_{\text{emb}}-1}$,
- **construct** a test statistic $T(f_\theta)$ that (i) measures

[1]We evaluate on publicly available datasets from OpenML: https://www.openml.org/.

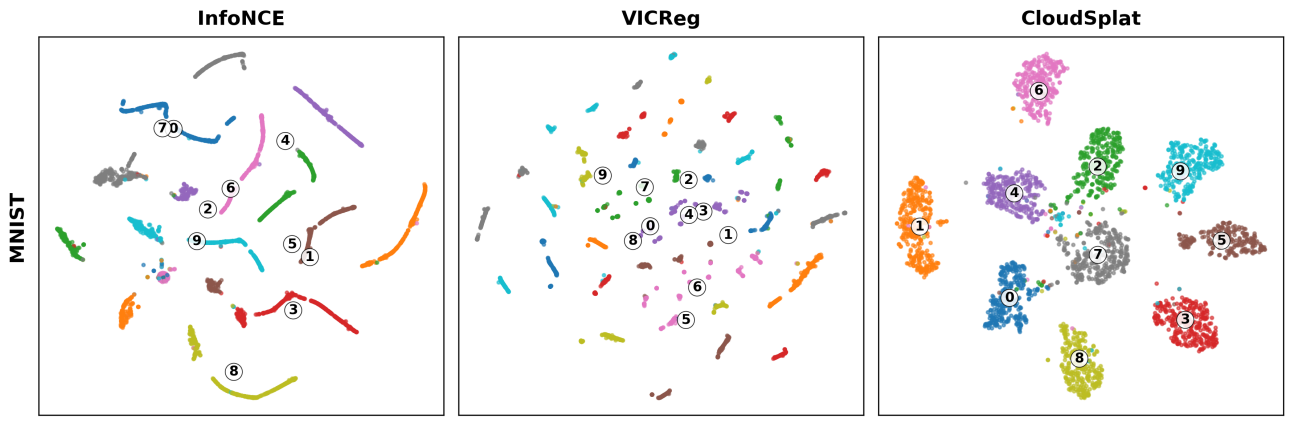


Figure 1. **CLOUDSPLAT (rightmost) produces compact, well-separated clusters.** t-SNE on MNIST (10 classes) with class labels: "0", "1", . . . "9". Notice that other baselines separate the members of each class to multiple, disconnected clusters.

whether $\mathcal{D}_{\text{neg}}$ is stochastically larger than $\mathcal{D}_{\text{pos}}$, (ii) is differentiable w.r.t. θ, and (iii) provides tunable sensitivity via a power parameter p,

- **such that** optimizing T as a training loss produces embeddings that maximize downstream task performance.

Our contributions span the testing-to-optimization bridge:

1. **A differentiable two-sample test statistic.** CSSI generalizes prior non-differentiable separability indices (DSI/KS) into a smooth, p-parameterized functional that admits backpropagation while retaining diagnostic power: Spearman $\rho \in [+0.73, +0.92]$ with downstream accuracy across three foundation models on 19 datasets (Table 1, Appendix A).
2. **From test to loss: contrastive learning via CSSI.** Using CSSI as a training objective yields CLOUDSPLAT, which wins on several datasets over recent baselines (Tables 2, 3, and Figure 2 (right)), including 7/11 OpenML-CC18 benchmarks against SCARF (Appendix D), demonstrating that optimizing a well-designed test statistic directly improves embedding quality.
3. **Sensitivity control via p.** The generalized-mean power p provides explicit control over the test's sensitivity profile: low p tests average separation, high p tests worst-case separation (Table 4). This is agnostic to encoder architecture (MLP, ResNet, Transformer; Section 4), label fraction, and evaluation protocol.

2. Related Work

Self-supervised learning (SSL) for tabular data must overcome the absence of natural augmentations that drive success in vision and language (Borisov et al., 2022). Only CLOUDSPLAT fulfills all specifications of Figure 2; we organize related work into two groups.

Augmentation strategies. The dominant input-space approach corrupts raw features to form positive pairs: SCARF (Bahri et al., 2021) replaces features with marginal samples, VIME (Yoon et al., 2020) adds mask estimation, and SubTab (Ucar et al., 2021) uses random feature subsets. All input-space methods corrupt without regard to the data manifold (Goyal et al., 2020), risking out-of-distribution samples or trivial shortcuts from intact correlated features. In representation space, RaTab (Eo et al., 2025) augments via truncated SVD but the perturbation is static; SwitchTab (Wu et al., 2024), BinRecon (Lee et al., 2024), and T-JEPA (Thimonier et al., 2025) use reconstruction or masked-prediction pretext tasks rather than contrastive objectives. CLOUDSPLAT differs by performing *learned* Gaussian perturbations in latent space with *varying* corruption rates across K views.

Loss objectives and distributional separation. The InfoNCE loss (Oord et al., 2018) mediates alignment and uniformity through a single temperature τ (Wang & Isola, 2020); low τ focuses on hard negatives but risks instability (Chuang et al., 2020), while high τ yields uninformative gradients. Dynamic schedules (Kukleva et al., 2023) and collapse-prevention regularizers such as VICReg (Bardes et al., 2021) have been explored, but primarily in vision. The Distributional Separation Index (DSI) (Guan & Loew, 2020, 2022) measures class separability by comparing distance distributions using the non-differentiable KS statistic. Optimal transport (OT) has been connected to contrastive learning (Shi et al., 2023; Chen et al., 2024), but existing work uses OT to model couplings between samples, not to compare distributions of positive vs. negative distances. Recent work learns deep kernels to maximize MMD test power (Liu et al., 2020), but the learned representations serve the test alone, not downstream tasks. CLOUDSPLAT inverts this: we use a kernel-free Wasserstein-based test statistic directly as a self-supervised loss, producing repre-

Property	SCARF	VIME	SubTab	PTaRL	T-JEPA	RaTab	CLOUDSPLAT
Latent-space augment.					✓	✓	✓
Collapse regularization				✓	✓		✓
Hard-negative separation							✓
Distributional loss							✓
Multi-augmented views			✓		✓		✓
Any encoder architecture	✓	✓		✓			✓
Feat. corr. preservation						✓	✓
Label-free encoder	✓	✓	✓		✓	✓	✓
Frozen encoder eval			✓		✓		✓

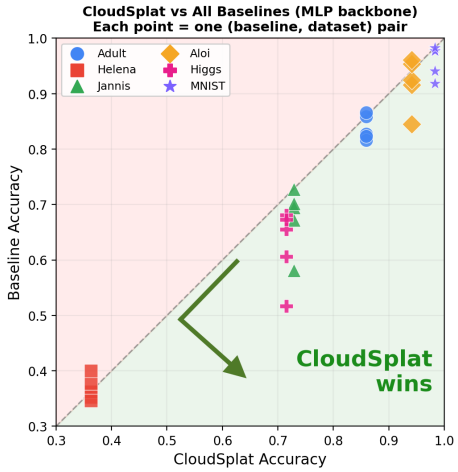


Figure 2. **CLOUDSPLAT wins on features (left) and accuracy (right).** *(Left)* Nine design axes; CLOUDSPLAT is the only method covering all nine. Bottom two rows: "label-free encoder" = no labels in encoder training; "frozen encoder eval" = encoder not fine-tuned downstream. *(Right)* Accuracy parity vs. SSL baselines (Table 3, MLP, frozen encoder, 100% labels). Each point is one (baseline, dataset) pair; below diagonal = CLOUDSPLAT wins. California excluded (RMSE metric).

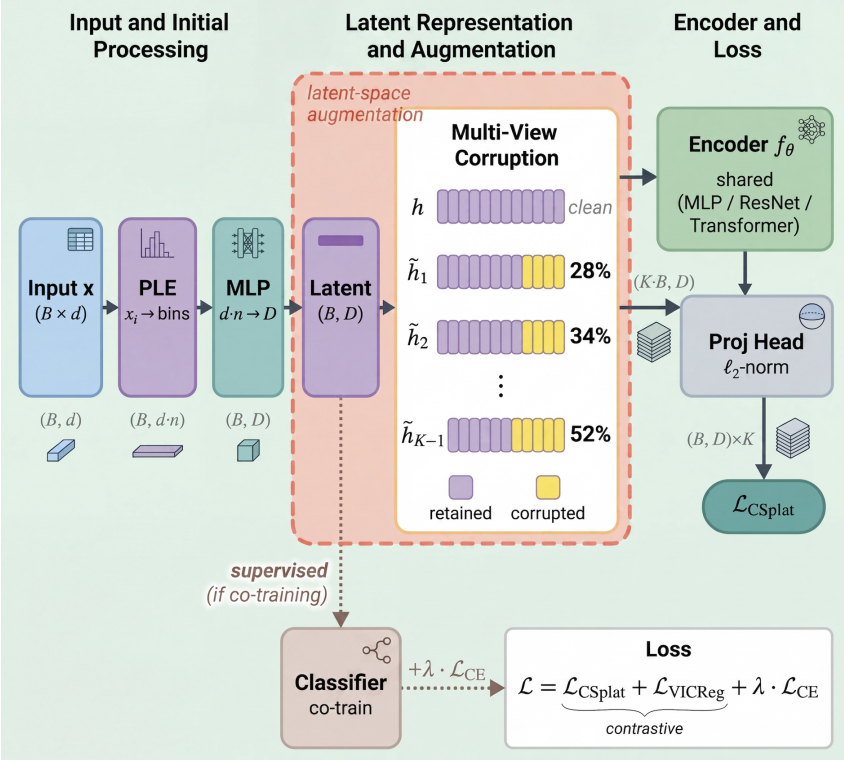


Figure 3. **CLOUDSPLAT pipeline:** Heterogeneous tabular input $\mathbf{x}_i$ is encoded through a unified MLP to latent representation $\mathbf{z}$. CLOUDSPLAT applies selective Gaussian noise in latent space ($\tilde{\mathbf{z}} = \mathbf{z} + \boldsymbol{\epsilon}$) with probability ρ. Both original and augmented representations pass through shared encoder and projection head to produce ℓ_2-normalized embeddings, optimized using either the CLOUDSPLAT$_\tau$ loss or the CLOUDSPLAT distributional margin loss with VICReg regularization.

sentations for downstream classification while eliminating kernel selection and providing explicit sensitivity control via p.

3. Method

3.1. Problem Setting and Architecture

Let $\mathcal{X} = \{x_i\}_{i=1}^N$ be a tabular dataset with n_d numerical and n_c categorical features. We learn an embedding function f_θ :

$\mathbb{R}^d \to \mathbb{S}^{d_{\text{emb}}-1}$ via self-supervised contrastive pretraining, then freeze the encoder and train a downstream classifier on the fixed embeddings (Figure 3).

Feature encoding + latent augmentation. Numerical features are encoded via PLE (Piecewise Linear Encoding) and mapped to a latent space $\mathbf{h} \in \mathbb{R}^D$ by a projection MLP. Unlike methods that corrupt raw input features, CLOUDSPLAT augments in this learned latent space:

$$\tilde{\mathbf{h}} = \mathbf{h} + \boldsymbol{\epsilon}, \quad \epsilon_j \sim \begin{cases} \mathcal{N}(0, \eta^2) & \text{w.p. } \rho, \\ 0 & \text{w.p. } 1-\rho, \end{cases} \tag{1}$$

creating positive pairs $(\mathbf{h}, \tilde{\mathbf{h}})$ that stay close to the data manifold (Algorithm 1, Step 1).

Encoder backbone f_θ. Any differentiable architecture (MLP, ResNet, Transformer) can serve as the encoder backbone. A projection head g_ϕ maps encoder outputs to ℓ_2-normalized embeddings on $\mathbb{S}^{D'-1}$.

From temperature schedules to distributional separation. The standard InfoNCE loss (Section 2) for a batch of B samples is:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(s_{i,i}/\tau)}{\sum_{k=1}^{B}\exp(s_{i,k}/\tau)}, \tag{2}$$

where $s_{i,j}$ is the cosine similarity between embeddings and $\tau > 0$ controls the alignment-uniformity tradeoff (Section 2).

CLOUDSPLAT$_\tau$: decoupled temperature as a first step. We decouple InfoNCE into separate temperatures for the positive pair (τ_{num}) and negatives (τ_{denom}):

$$\mathcal{L}_{\text{decoupled}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(s_{i,i}/\tau_{\text{num}})}{\sum_{k=1}^{B}\exp(s_{i,k}/\tau_{\text{denom}})}, \tag{3}$$

equivalent to the concentration parameter $\beta = 1/\tau_{\text{denom}}$ discussed in Section 2. We anneal τ_{denom} from 1.0 to a terminal value in $[0.05, 0.1]$ (selected per dataset by Optuna) during training. This improves over vanilla InfoNCE (Table 2), but requires a hand-designed schedule and the log-sum-exp still collapses the negative distribution into a single scalar. Moreover, without VICReg regularization, small τ_{denom} concentrates gradients on a few hard negatives, allowing the encoder to satisfy the loss with a rank-1 projection, producing dimensional collapse.

The CLOUDSPLAT loss: distributional margin via sorted comparison. Rather than tuning temperature schedules, we directly compare the *distributions* of positive and negative pairwise distances, sidestepping the retrieval-versus-clustering trade-off where no single τ is optimal across tasks. We construct two point clouds on the real line: $\mathcal{D}_{\text{pos}}$ (within-sample distances) and $\mathcal{D}_{\text{neg}}$ (between-sample distances), sort both, and compare position by position. The penalty at each position k is zero when the negative is sufficiently farther than the positive, and positive otherwise:

$$\mathcal{L}_{\text{CSplat}} = \left(\frac{1}{N_{\text{pairs}}} \sum_{k=1}^{N_{\text{pairs}}} \left[\max(0,\ d^+_{(k)} - d^-_{(k)} + m)\right]^p \right)^{1/p}, \tag{4}$$

where $d^+_{(k)}$ and $d^-_{(k)}$ are the k-th smallest positive and negative distances (sorted ascending), N_{pairs} is the matched cardinality after subsampling (B for single-view, $\binom{K}{2}B$ for multi-view), $m > 0$ is the margin, and $p \geq 1$ controls the generalized mean. In practice, we replace $\max(0, \cdot)$ with $\text{softplus}(x) = \log(1 + e^x)$, which provides gradient for all finite inputs rather than zeroing out once the margin is satisfied.

This extends the Distributional Separation Index (DSI (Guan & Loew, 2020, 2022), Section 2), which compares distance distributions using the non-differentiable Kolmogorov-Smirnov statistic as a supervised evaluation metric. Our differentiable variant, the CloudSplat Separability Index (CSSI), replaces the KS statistic with a 1D Wasserstein formulation: at p=1 and m=0 it reduces to the signed W_1 gap between sorted distances (a Wasserstein analogue of DSI), and as $p \to \infty$ it focuses on the single worst-separated quantile position (W_∞). We make three changes: (a) replace class labels with view correspondence (self-supervised), (b) replace KS with the 1D Wasserstein distance (differentiable), and (c) add a directional margin penalty. For equal-size empirical distributions, 1D Wasserstein admits a closed form: sort both sets and compare position-by-position (Algorithm 1, Step 4). While KS captures only the single point of maximum divergence, the Wasserstein formulation with p=1 penalizes all positions equally and higher p concentrates on the worst-separated positions.

Two departures from standard losses.

1. **Distributions, not individual pairs.** Triplet loss compares one (d^+, d^-) pair per anchor; InfoNCE collapses all negatives into one scalar. CLOUDSPLAT compares full sorted distributions, providing gradient to every pair.
2. **Generalized p-mean.** p=1 averages penalties; p=2 upweights the worst-separated positions; $p\to\infty$ focuses on the single worst position. Empirically, accuracy improves sharply from p=1 to p=20 and plateaus at $p \in [20, 100]$, with slight decline at p=200 (Appendix B).

To prevent collapse, we add VICReg regularization (Sec-

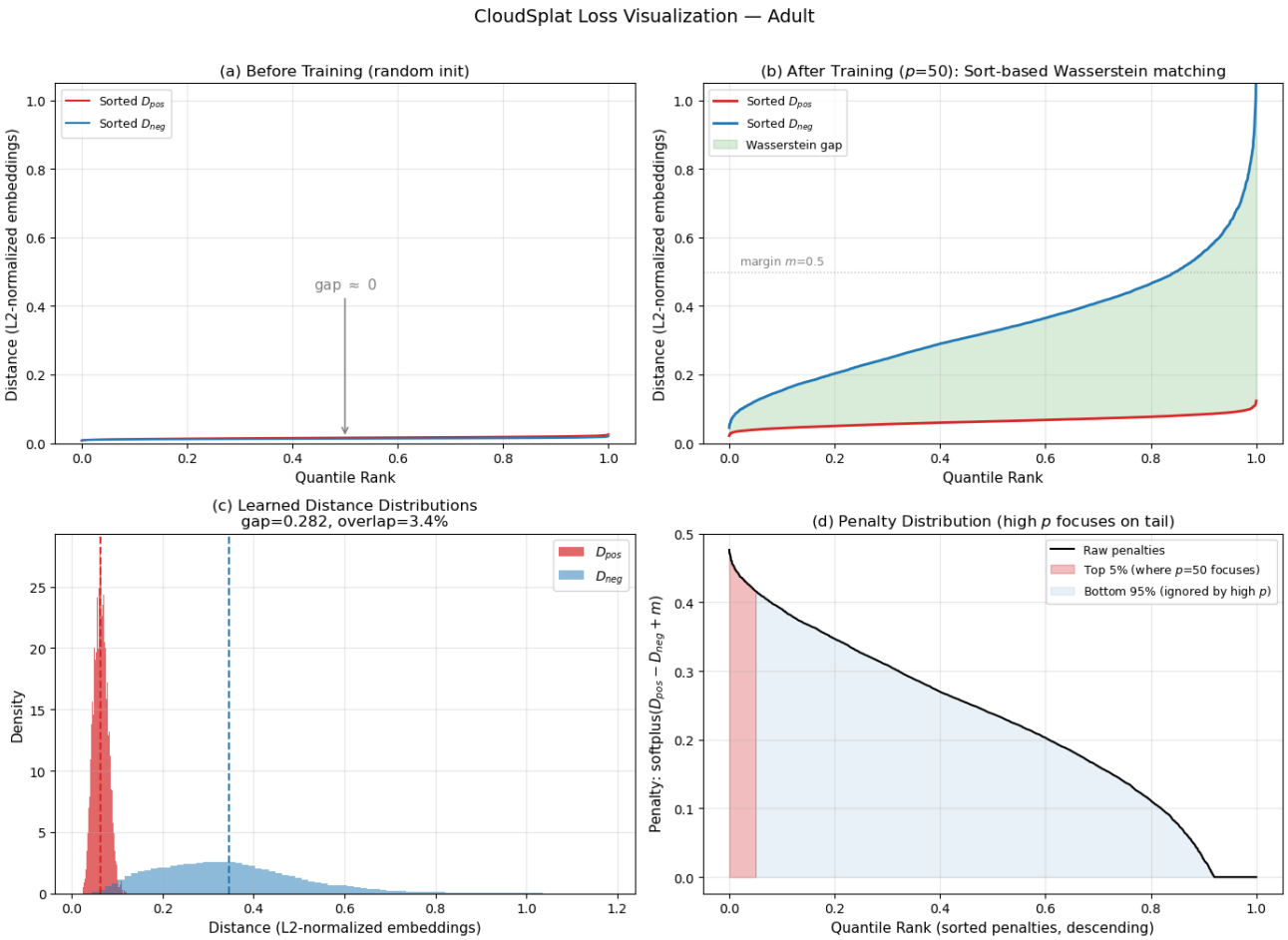


Figure 4. **CLOUDSPLAT loss mechanism on Adult (p=50).** (a) Before training: sorted positive and negative distances are indistinguishable (gap ≈ 0). (b) After training: sort-based Wasserstein matching (Eq. 4) produces separation across all quantile ranks; the dashed line shows margin m=0.5. (c) Learned distance distributions: 3.4% overlap, mean gap $= 0.282$ (margin satisfied at upper quantiles where $d^- - d^+ > m$). (d) Penalty distribution (sorted descending): high p concentrates gradient on the top 5% of hardest violations (pink), leaving the satisfied 95% with near-zero gradient.

tion 2):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CSplat}} + \lambda_{\text{var}} \cdot \mathcal{L}_{\text{var}} + \lambda_{\text{cov}} \cdot \mathcal{L}_{\text{cov}}. \tag{5}$$

3.2. Multi-View Extension (MultiCloudSplat)

The single-view setting produces only B positive distances but $B(B-1)$ negatives, undersampling the positive distribution. **MultiCloudSplat** generates K views per sample at linearly spaced corruption rates, yielding $\binom{K}{2} \cdot B$ positive distances that span easy-to-hard pairs. Negatives are computed from a single view pair (cost scales linearly with K). The full procedure is in Algorithm 2 (Appendix).

Variant summary. We denote the single-view distributional margin loss as CLOUDSPLAT $_0$(Algorithm 1), its multi-view extension as CLOUDSPLAT (Algorithm 2, Appendix), and the variant that adds a supervised cross-entropy term ($\lambda \cdot \mathcal{L}_{\text{CE}}$) on labeled data during pretraining as CLOUDSPLAT$_{\text{cotrain}}$.

Curriculum scheduling. Both variants benefit from progressive training: CLOUDSPLAT$_\tau$ anneals temperature and corruption jointly, while the CLOUDSPLAT loss anneals corruption rate alone (coarse-to-fine).

4. Experimental Results

We evaluate the CSSI statistic in two roles: first as a diagnostic (Sections 4.1–4.2), then as a training loss (Sections 4.3–4.4). All experiments use publicly available OpenML tabular datasets (Vanschoren et al., 2013) with preprocessing from (Cui et al., 2024) and 5 runs per configuration.

Algorithm 1 CLOUDSPLAT $_0$: Single-View Distributional Margin Loss

Require: Batch $\{x_i\}_{i=1}^B$; corruption rate ρ; noise scale η; margin m; power p
Ensure: Scalar loss $\mathcal{L}_{\text{total}}$
1: **// Step 1: Project to latent space, create corrupted view**
2: $\mathbf{h}_i \leftarrow \text{MLP}(x_i)$; $\tilde{\mathbf{h}}_i \leftarrow \mathbf{h}_i + \text{Bernoulli}(\rho) \odot \mathcal{N}(0, \eta^2)$ $\forall i$
3: **// Step 2: Encode + ℓ_2-normalize both views (shared weights)**
4: $\hat{\mathbf{z}}_i^{(0)} \leftarrow \text{Encode}(\mathbf{h}_i)$; $\hat{\mathbf{z}}_i^{(1)} \leftarrow \text{Encode}(\tilde{\mathbf{h}}_i)$ $\forall i$
5: **// Step 3: Pairwise squared distances**
6: $d_i^+ \leftarrow \|\hat{\mathbf{z}}_i^{(0)} - \hat{\mathbf{z}}_i^{(1)}\|^2$ ▷ *B within-sample (positive) distances*
7: $d_{ij}^- \leftarrow \|\hat{\mathbf{z}}_i^{(0)} - \hat{\mathbf{z}}_j^{(1)}\|^2,\ i \neq j$ ▷ *Between-sample (negative) distances*
8: Subsample B negatives uniformly to match $|\{d^+\}|$ ▷ *Equal sizes for exact 1D Wasserstein*
9: **// Step 4: Sort both, compare position by position (core of CLOUDSPLAT)**
10: $\mathbf{d}^+ \leftarrow \text{Sort}(\{d_i^+\})$; $\mathbf{d}^- \leftarrow \text{Sort}(\text{subsampled negatives})$
11: $\mathcal{L}_{\text{CSplat}} \leftarrow \left(\frac{1}{B}\sum_{k=1}^B \left[\text{softplus}(d_k^+ - d_k^- + m)\right]^p\right)^{1/p}$
12: **// Step 5: VICReg on encoder outputs before ℓ_2-normalization (Eq. 5)**
13: $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{CSplat}} + \lambda_{\text{var}} \cdot \mathcal{L}_{\text{var}} + \lambda_{\text{cov}} \cdot \mathcal{L}_{\text{cov}}$ $\mathcal{L}_{\text{total}}$

Table 1. **CSSI as a diagnostic:** Spearman ρ(metric, LP accuracy) across $n = 19$ datasets for three tabular FM configurations. CSSI achieves strong rank correlation ($\rho \in [+0.73, +0.92]$) while being the only metric that is differentiable and admits explicit sensitivity control via p.

Signal	TabPFN-solo	TabPFN-ens	TabICL	ICL-mean
CSSI_1 (ours)	0.92	0.91	0.73	0.83
kNN-10 acc.	0.89	**0.97**	**0.95**	**0.94**
Silhouette	**0.95**	0.96	0.88	0.93
−Davies-Bouldin	0.91	0.93	0.83	0.91
Calinski-Harab.	0.79	0.85	0.70	0.75

4.1. CSSI as a Diagnostic Test Statistic

Beyond its use as a training loss, CSSI serves as a probe-free diagnostic for evaluating frozen embeddings, analogous to running a hypothesis test on the learned representation. Table 1 reports Spearman ρ between five probe-free metrics and linear-probe accuracy across 19 OpenML datasets for three tabular foundation models. While kNN-10 and Silhouette achieve numerically higher ρ on some configurations, neither is differentiable: kNN requires discrete neighbor assignment and Silhouette assumes convex cluster geometry. CSSI uniquely combines (a) competitive diagnostic power ($\rho \in [+0.73, +0.92]$), (b) differentiability via sorting, and (c) explicit sensitivity control through p.

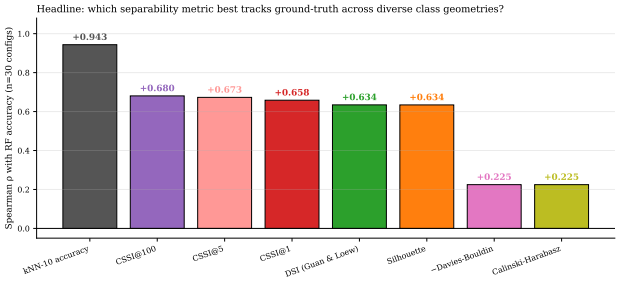


Figure 5. **Which separability metric best tracks ground truth?** Spearman ρ(metric, RF accuracy) across 30 (shape, noise) configurations. Centroid-based metrics (Davies-Bouldin, Calinski-Harabasz) fail on non-convex shapes ($\rho \approx 0.22$). CSSI maintains $\rho \approx 0.67$ while being the only differentiable option among the distance-distribution metrics.

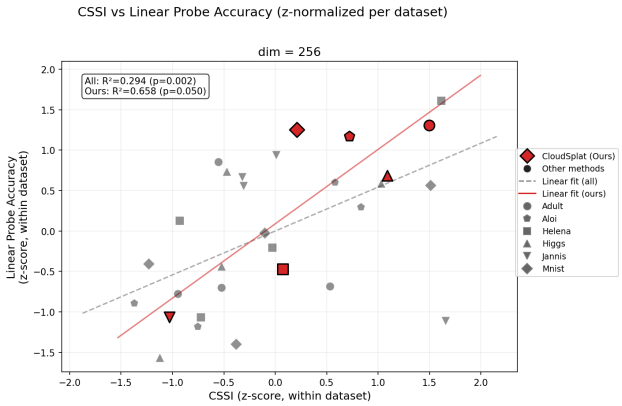


Figure 6. **CSSI as diagnostic:** CLOUDSPLAT separability index (CSSI) vs. downstream accuracy across six datasets, z-score normalized per (dataset, d). Most points are along the diagonal. Notice that the points for CLOUDSPLAT (red) consistently occupy the upper-right quadrant (highest CSSI and accuracy).

Figure 5 stress-tests these metrics on 30 controlled synthetic configurations (6 shapes $\times$ 5 noise levels). Centroid-based metrics collapse on non-convex geometries ($\rho \approx 0.22$), while CSSI maintains $\rho \approx 0.67$. A permutation test confirms CSSI retains statistical power while providing exact finite-sample Type-I error control (Appendix A.3). Full synthetic geometry analysis is in Appendix A.

4.2. Sensitivity Control via p

The p parameter provides explicit control over the test's sensitivity profile, interpolating between average-case testing (p=1, analogous to a paired t-test on sorted distances) and worst-case testing ($p \to \infty$, focusing on the single

Table 2. **CLOUDSPLAT wins on loss ablation:** All five losses use the same MLP+PLE architecture and hyperparameter budget (150 Optuna trials). Downstream: linear probe on frozen embeddings. Best and 2nd best per column. California uses RMSE (↓); all others use accuracy (↑).

(a) Label fraction = 0.05 (5% labeled data)

Loss	Helena	Jannis	Adult	Higgs	Aloi	MNIST	Calif.↓	Avg Rank
Baselines								
Vanilla InfoNCE	27.37±0.26	73.34±0.27	83.46±0.56	67.67±1.09	65.46±0.40	89.57±3.17	0.491±.011	3.43
VICReg	26.29±0.55	71.85±0.75	84.15±0.61	64.37±2.47	70.89±0.68	93.49±0.80	0.520±.014	3.57
Our proposed methods								
CLOUDSPLAT$_\tau$	27.49±0.26	73.50±0.22	72.50±24.6	68.85±0.24	70.85±0.56	93.69±0.29	0.497±.009	2.43
CLOUDSPLAT$_0$	27.44±0.19	68.69±4.48	84.87±0.20	66.19±1.60	67.54±0.55	93.92±0.32	0.502±.009	3.00
CLOUDSPLAT	28.47±0.32	72.84±0.33	82.85±0.61	68.98±0.32	67.06±0.64	92.76±0.49	0.490±.011	2.57
							Proposed methods win:	**6/7**

(b) Label fraction = 1.0 (100% labeled data)

Loss	Helena	Jannis	Adult	Higgs	Aloi	MNIST	Calif.↓	Avg Rank
Baselines								
Vanilla InfoNCE	36.25±0.27	77.61±0.33	85.85±0.34	77.01±0.31	93.18±0.15	96.39±3.85	0.395±.009	3.36
VICReg	34.71±0.15	75.03±2.57	85.76±0.35	77.01±0.88	94.84±0.18	98.42±0.11	0.405±.010	3.93
Our proposed methods								
CLOUDSPLAT$_\tau$	35.95±0.12	76.42±0.79	85.42±0.70	77.50±0.27	95.99±0.18	98.37±0.12	0.390±.008	3.14
CLOUDSPLAT$_0$	36.05±0.26	77.09±0.29	85.91±0.19	77.85±0.15	95.68±0.13	98.38±0.17	0.389±.007	1.71
CLOUDSPLAT	36.44±0.44	77.00±0.32	85.89±0.16	77.72±0.35	—	98.03±0.05	0.391±.007	2.50
							Proposed methods win:	**5/7**

CLOUDSPLAT$_\tau$ = InfoNCE with asymmetric $\tau_{\text{num}} \neq \tau_{\text{denom}}$; Vanilla InfoNCE = standard InfoNCE with single τ.

worst-separated quantile). Accuracy improves sharply from p=1 to p=20 (+15.4pp average gain), then plateaus at $p \in [20, 100]$ before declining slightly at p=200 (Table 4). Higher p concentrates gradient on the hardest margin violations, implementing implicit hard-negative mining; the practical optimum at large but finite p validates the theoretical prediction that $p \to \infty$ focuses on the single worst-separated position (full analysis in Appendix B).

4.3. From Test to Loss: Baseline Comparison

Having established CSSI as a valid diagnostic, we now demonstrate the testing-optimization bridge: using the same functional as a contrastive training loss. The full training set is used for self-supervised contrastive pretraining (no labels); when a label fraction $f < 1$ is specified, an optional co-training weight $\lambda > 0$ adds supervised loss on the labeled subset. The encoder is then frozen and a downstream classifier is trained on fixed embeddings: (i) **linear probe** (Table 2), isolating representation quality; (ii) **deep MLP** (Table 3), tuned by Optuna to match T-JEPA (Thimonier et al., 2025). We report at f=0.05 and f=1.0; accuracy for classification, RMSE for regression. All loss variants share the same MLP+PLE architecture (256-dim, 4-layer) and hyperparameter budget (150 Optuna trials per dataset per loss). All three variants outperform both baselines in aggregate (6/7 and 5/7 wins); CLOUDSPLAT achieves lowest variance (max std < 1.1 vs. CLOUDSPLAT$_\tau$'s 24.6 on Adult).

Table 3 compares CLOUDSPLAT against 6 SSL baselines on 7 datasets. Among frozen-encoder methods (excluding PTaRL/SwitchTab which use labels in encoder training, and SCARF/RaTab which fine-tune downstream), CLOUDSPLAT achieves rank 1.9 (MLP) and 2.5 (ResNet, behind T-JEPA's 1.7). Notably, CLOUDSPLAT never ranks below 3.5 on any dataset, whereas T-JEPA collapses to rank 6 on Higgs (0.517, below no-pretrain). Unlike T-JEPA, which stores three separate networks during training (context encoder, EMA target, predictor), CLOUDSPLAT reuses a single encoder across all K views, obtaining $\binom{K}{2}$=15 positive pairs per sample with no additional parameters (∼15× richer contrastive signal at ∼3× fewer training-time parameters). With labels during pretraining, CLOUDSPLAT$_{\text{cotrain}}$ achieves rank 1.9, outperforming XGBoost, CatBoost, PTaRL, and SwitchTab on 5/7 datasets. As a diagnostic, Figure 6 confirms that CLOUDSPLAT consistently achieves both higher CSSI and higher downstream accuracy than baselines across datasets and embedding dimensions.

Table 3. **CLOUDSPLAT gets most consistent results and best avg ranks** SSL methods with MLP and ResNet downstream models. CLOUDSPLAT uses frozen encoder + deep downstream model with Bayesian hyperparameter optimization. Best and 2nd best per downstream choice. Gray rows use labeled data during pretraining or fine-tune end-to-end. $\text{CLOUDSPLAT}_{\text{cotrain}}$ achieves the best overall rank (1.9), outperforming all baselines including XGBoost and CatBoost.

	AD ↑	HE ↑	JA ↑	AL ↑	CA ↓	HI ↑	MNIST ↑	Avg Rank ↓
MLP downstream model								
No pretrain	0.827 ±1e-3	0.353 ±1e-3	0.672 ±1e-3	0.916 ±4e-3	0.511 ±3e-3	0.681 ±4e-3	0.978 ±2e-3	4.1
+VIME	0.859 ±3e-3	0.362 ±6e-3	0.695 ±5e-3	0.925 ±5e-3	0.505 ±4e-2	0.655 ±6e-3	0.941 ±2e-3	3.9
+BinRecon	0.816 ±8e-3	0.346 ±2e-2	0.581 ±8e-2	0.845 ±5e-2	0.498 ±1e-2	0.606 ±8e-2	0.918 ±5e-2	5.4
+SubTab	0.823 ±3e-3	0.378 ±2e-3	0.702 ±1e-3	0.954 ±2e-3	0.519 ±3e-2	0.673 ±2e-3	0.977 ±2e-3	3.6
+T-JEPA	0.866 ±2e-3	0.400 ±4e-3	0.728 ±3e-3	0.961 ±6e-3	0.468 ±4e-2	0.517 ±8e-2	0.983 ±1e-3	2.1
MLP downstream model + CLOUDSPLAT *(ours)*								
+CLOUDSPLAT	0.859 ±4e-3	0.363 ±6e-3	0.729 ±4e-3	0.941 ±4e-3	0.447 ±6e-3	0.715	0.983	1.9
ResNet downstream model								
No pretrain	0.814 ±7e-3	0.351 ±2e-3	0.666 ±3e-3	0.919 ±2e-3	0.534 ±2e-3	0.674 ±4e-3	0.979 ±1e-3	5.6
+VIME	0.851 ±1e-3	0.372 ±2e-3	0.699 ±3e-3	0.959 ±1e-3	0.505 ±1e-2	0.688 ±3e-3	0.933 ±1e-3	3.8
+BinRecon	0.828 ±9e-3	0.327 ±1e-2	0.699 ±3e-3	0.944 ±1e-3	0.471 ±1e-2	0.711 ±3e-3	0.981 ±2e-3	3.8
+SubTab	0.823 ±3e-3	0.365 ±3e-3	0.702 ±1e-3	0.958 ±1e-3	0.487 ±2e-2	0.675 ±3e-3	0.984 ±6e-4	3.6
+T-JEPA	0.865 ±3e-3	0.401 ±2e-3	0.718 ±3e-3	0.964 ±1e-3	0.441 ±8e-2	0.705 ±5e-3	0.983 ±2e-3	1.7
ResNet downstream model + CLOUDSPLAT *(ours)*								
+CLOUDSPLAT	0.858	0.382	0.730	0.954 ±5e-4	0.427 ±1e-3	0.732	0.979	2.5
Labels used throughout								
+PTaRL (MLP)	0.868 ±4e-3	0.383 ±2e-3	0.710 ±4e-3	0.917 ±3e-3	0.489 ±2e-3	0.713 ±2e-3	0.977 ±1e-3	4.3
+SwitchTab (MLP)	0.867 ±N/A	0.387 ±N/A	0.726 ±N/A	0.942 ±N/A	0.452 ±N/A	0.724 ±N/A	N/A	3.3
+PTaRL (ResNet)	0.862 ±5e-3	0.383 ±2e-3	0.723 ±5e-3	0.895 ±1e-3	0.498 ±1e-3	0.713 ±2e-3	0.973 ±1e-3	4.9
XGBoost	0.872 ±5e-4	0.375 ±1e-3	0.721 ±1e-3	0.951 ±1e-3	0.433 ±2e-3	0.729 ±1e-3	0.980 ±1e-3	3.0
CatBoost	0.873 ±1e-3	0.381 ±1e-3	0.721 ±1e-3	0.946 ±9e-4	0.430 ±7e-4	0.726 ±8e-4	0.972 ±3e-3	3.3
$+\text{CLOUDSPLAT}_{\text{cotrain}}$ (ours)	0.870	0.374 ±2e-3	0.788 ±3e-3	0.961	0.402 ±8e-3	0.820 ±2e-2	0.984	1.9

AD=Adult, HE=Helena, JA=Jannis, AL=Aloi, CA=California (RMSE), HI=Higgs. Where ± is omitted, result reflects a single seed.

Table 4. **Higher p concentrates gradient on hard negatives and improves accuracy.** Linear probe accuracy vs. generalized-mean power p. Low p averages penalties uniformly; high p focuses on the worst-separated positions (implicit hard-negative mining). Best per dataset; all other hyperparameters fixed (MLP encoder, d=256, K=6, 5 seeds).

	p=1	p=2	p=5	p=10	p=20	p=50	p=100	p=200
Adult	$.784_{\pm.011}$	$.783_{\pm.015}$	$.792_{\pm.015}$	$.807_{\pm.013}$	$.828_{\pm.009}$	$.830_{\pm.008}$	$.826_{\pm.003}$	$.825_{\pm.008}$
Helena	$.145_{\pm.010}$	$.138_{\pm.014}$	$.203_{\pm.020}$	$.223_{\pm.012}$	$.248_{\pm.006}$	$.242_{\pm.010}$	$.253_{\pm.016}$	$.233_{\pm.008}$
MNIST	$.563_{\pm.015}$	$.531_{\pm.042}$	$.680_{\pm.050}$	$.812_{\pm.028}$	$.819_{\pm.026}$	$.831_{\pm.012}$	$.819_{\pm.010}$	$.816_{\pm.014}$

4.4. Direct Comparison with SCARF

We also compare CLOUDSPLAT (frozen encoder) against SCARF (Bahri et al., 2021) (end-to-end fine-tuned) on 11 OpenML-CC18 datasets (Appendix D); CLOUDSPLAT wins on 7/11 despite the harder protocol.

5. Conclusion

We presented CSSI, a differentiable two-sample test statistic for embedding separability, and demonstrated the testing-to-optimization bridge it enables:

- **As a test statistic,** CSSI achieves Spearman $\rho \in [+0.73, +0.92]$ with linear-probe accuracy across three foundation-model architectures on 19 datasets (Table 1), correctly identifies separability on non-convex geometries where centroid-based metrics fail (Figure 5), and (unlike kNN or silhouette) admits back-propagation.
- **As a training loss,** CSSI directly optimizes embedding quality: CLOUDSPLAT wins on 6/7 datasets at 5% labels and 5/7 at 100% (Table 2), outperforms SCARF on 7/11 OpenML-CC18 benchmarks (Appendix D), and achieves rank 1.9 (MLP) against T-JEPA and five SSL baselines (Table 3), all without labels during encoder training.
- **Sensitivity control via p.** The generalized-mean power interpolates between average-case (p=1, t-test analogue) and worst-case ($p\to\infty$, W_∞ analogue) testing, with the practical optimum at $p \in [20, 100]$ (Table 4). This control is agnostic to encoder architecture (MLP, ResNet, Transformer), label fraction, and downstream protocol.

Impact Statement

This work advances representation learning methodology. We foresee no specific negative societal consequences.

References

Bahri, D., Jiang, H., Tay, Y., and Metzler, D. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

Bardes, A., Ponce, J., and LeCun, Y. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., and Kasneci, G. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 2022.

Chen, Z., Lin, C.-H., Liu, R., Xiao, J., and Dyer, E. L. Your contrastive learning problem is secretly a distribution alignment problem. In *Advances in Neural Information Processing Systems*, 2024.

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., and Jegelka, S. Debiased contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8765–8775, 2020.

Cui, W., Hosseinzadeh, R., Ma, J., Wu, T., Sui, Y., and Golestan, K. Tabular data contrastive learning via class-conditioned and feature-correlation based augmentation. *arXiv preprint arXiv:2404.17489*, 2024.

Eo, M., Lee, K., Cho, H.-S., Kim, D., Sim, Y. S., and Lim, W. Representation space augmentation for effective self-supervised learning on tabular data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11625–11633, 2025.

Goyal, S., Raghunathan, A., Jain, M., Simhadri, H. V., and Jain, P. Drocc: Deep robust one-class classification. In *International conference on machine learning*, pp. 3711–3721. PMLR, 2020.

Guan, S. and Loew, M. Data separability for neural network classifiers and the development of a separability index. *arXiv preprint arXiv:2005.13120*, 2020.

Guan, S. and Loew, M. A novel intrinsic measure of data separability. *Applied Intelligence*, 52(16):19289–19302, 2022.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Kukleva, A., Böhle, M., Schiele, B., Kuehne, H., and Rupprecht, C. Temperature schedules for self-supervised contrastive methods on long-tail data. *arXiv preprint arXiv:2303.13664*, 2023.

Lee, K., Sim, Y. S., Cho, H.-S., Eo, M., Yoon, S., Yoon, S., and Lim, W. Binning as a pretext task: Improving self-supervised learning in tabular domains. *arXiv preprint arXiv:2405.07414*, 2024.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., and Sutherland, D. J. Learning deep kernels for non-parametric two-sample tests. In *International Conference on Machine Learning*, pp. 6316–6326. PMLR, 2020.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Shi, L., Zhang, G., Zhen, H., Fan, J., and Yan, J. Understanding and generalizing contrastive learning from the inverse optimal transport perspective. In *International Conference on Machine Learning*, pp. 31408–31421. PMLR, 2023.

Thimonier, H., De Melo Costa, J. L., Popineau, F., Rimmel, A., and Doan, B.-L. Tabular data pre-training via joint embedding predictive architecture. In *International Conference on Learning Representations*, 2025.

Ucar, T., Hajiramezanali, E., and Edwards, L. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34:18853–18865, 2021.

Vanschoren, J., van Rijn, J. N., Bischl, B., and Torgo, L. Openml: networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL http://doi.acm.org/10.1145/2641190.264119.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Wu, J., Chen, S., Zhao, Q., Sergazinov, R., Li, C., Liu, S., Zhao, C., Xie, T., Guo, H., Ji, C., et al. Switchtab: Switched autoencoders are effective tabular learners. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15924–15933, 2024.

Yoon, J., Zhang, Y., Jordon, J., and Van der Schaar, M. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in neural information processing systems*, 33:11033–11043, 2020.

Appendix

A. CSSI as a Diagnostic: Comparison to Alternative Test Statistics

CSSI is a differentiable two-sample test statistic that measures whether positive (within-class) and negative (between-class) embedding distances are separable. Beyond its use as a training loss, CSSI serves as a probe-free diagnostic for evaluating frozen embeddings, analogous to running a hypothesis test on the learned representation. We validate the diagnostic power of this test via two experiments: (1) a metric comparison on real foundation-model embeddings, and (2) a controlled synthetic-geometry power analysis.

A.1. Metric comparison on foundation-model embeddings

Results are presented in Table 1 (Section 4.1).

A.2. Synthetic geometry power analysis

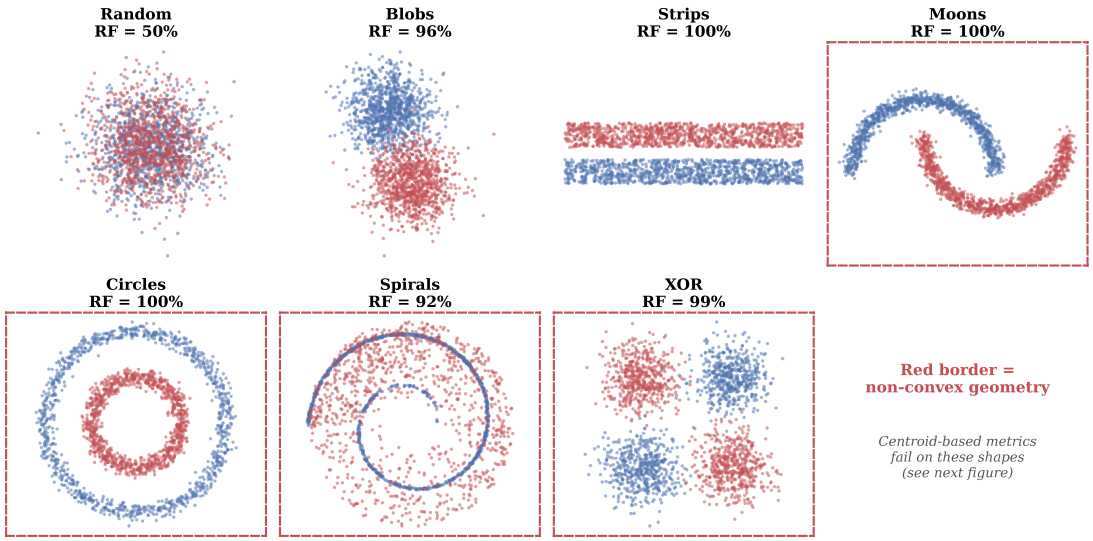


Figure 7. **Geometry zoo:** Seven controlled 2D binary-classification geometries (1,000 points per class). Dashed red borders mark non-convex geometries where centroid-based metrics fail. All shapes except Random are separable (RF accuracy ≥ 0.92).

Figure 7 shows seven controlled 2D class geometries used to stress-test separability metrics. On convex geometries (Blobs, Strips) all metrics agree. On non-convex geometries (Circles: Sil=0.12, RF=1.00; Spirals: Sil=0.00, RF=0.92; XOR: Sil=0.13, RF=0.99), Silhouette under-reports separability (Figure 8b), while CSSI_{100} correctly assigns high values to all separable shapes regardless of convexity (Figure 8c). The p-sensitivity analysis (Figure 9) confirms that increasing p monotonically sharpens the separability signal across all geometries.

Figure 5 (Section 4.1) scales this analysis to 30 configurations (6 shapes $\times$ 5 noise levels), computing Spearman ρ against Random Forest accuracy as ground truth. Centroid-based metrics collapse ($\rho \approx +0.22$) while distributional metrics including CSSI remain robust ($\rho \in [+0.63, +0.68]$). The kNN-10 accuracy baseline achieves $\rho = 0.94$ but is non-differentiable, precluding its use as a training loss.

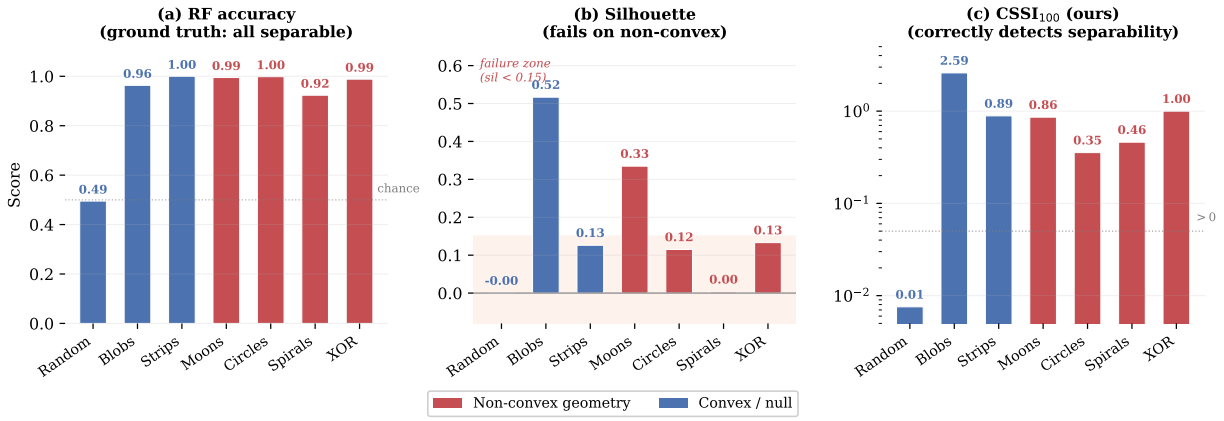


Figure 8. **Silhouette fails on non-convex geometries; CSSI does not.** (a) Random Forest accuracy confirms all shapes (except Random) are separable. (b) Silhouette scores non-convex shapes in the failure zone (< 0.15), incorrectly reporting "not separable." (c) CSSI_{100} correctly assigns high values to all separable shapes regardless of convexity (log scale; threshold line at > 0). Red bars = non-convex geometry; blue = convex or null.

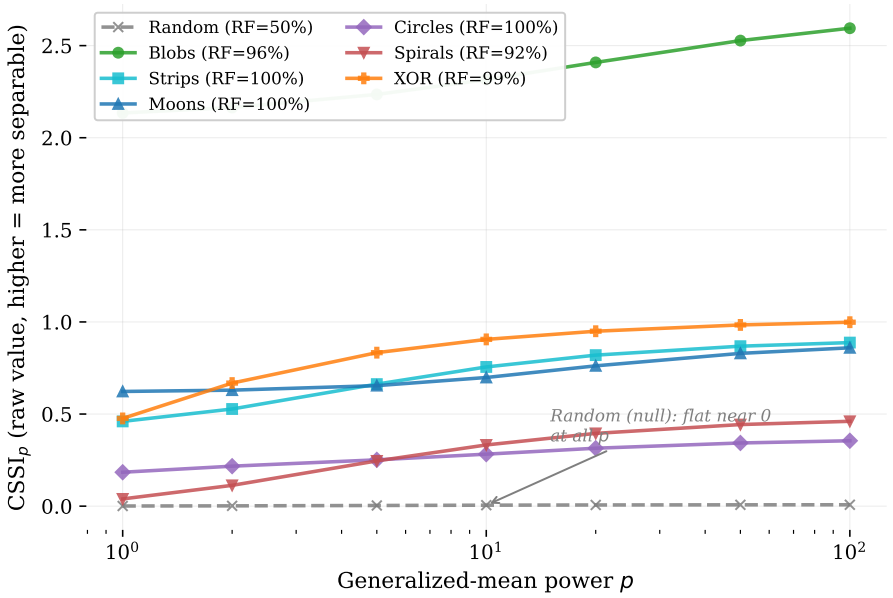


Figure 9. **CSSI_p sensitivity to the generalized-mean power p.** Each curve shows raw CSSI_p as p increases (log-scale x-axis); higher values indicate stronger separability. Higher p focuses the statistic on the hardest quantile position; all separable shapes exhibit monotone increase, while the null case (Random, dashed gray) remains flat near zero at all p.

A.3. Permutation Test: Formal Type-I Error Control

To validate that CSSI functions as a proper test statistic, we conduct a permutation test. Under H_0: positive/negative labels are exchangeable (no class separation). We pool all pairwise distances, randomly reassign labels, and recompute CSSI 2,000 times to construct the null distribution. The p-value is the fraction of permuted values exceeding the observed CSSI (one-sided, right tail: higher CSSI = more separation).

Figure 10 confirms: (a) CSSI has statistical power to detect genuine class separation; (b) it does not hallucinate separation under the null; (c) the permutation framework provides exact finite-sample Type-I error control at any nominal α. This validates the testing interpretation: the same functional used as a training loss (Section 4.3) is also a formally valid hypothesis test statistic.

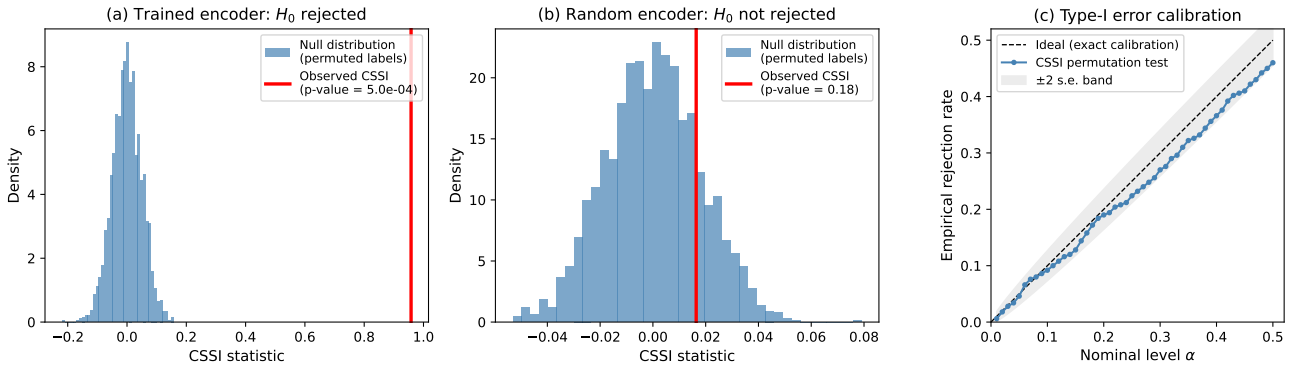


Figure 10. **CSSI permutation test.** (a) Trained encoder: observed CSSI lies far right of the null ($p < 0.001$), correctly detecting separation. (b) Random encoder: observed CSSI falls within null ($p = 0.18$), correctly failing to reject H_0. (c) Calibration: empirical rejection rate vs. nominal α across 500 H_0 runs tracks the diagonal within ± 2 s.e., confirming exact finite-sample Type-I error control.

Consistency. Under H_1 (population $\mathrm{CSSI}_1 := \mathbb{E}[D^-] - \mathbb{E}[D^+] > 0$), bounded distances ($\ell_2$-normalized embeddings give $d \in [0, 4]$) and the law of large numbers yield $\widehat{\mathrm{CSSI}}_1 \to \mathrm{CSSI}_1 > 0$. Under H_0, exchangeability of positive/negative labels forces $\mathbb{E}_{\mathrm{perm}}[\widehat{\mathrm{CSSI}}_1] = 0$ with finite variance, so the permutation null concentrates at zero. Hence the permutation p-value tends to 0 under H_1 (power $\to 1$) and is exactly uniform on $[0, 1]$ under H_0, empirically confirmed by panel (c). The test is consistent against all alternatives where the negative distance distribution has strictly larger mean than the positive, the natural separation condition in contrastive learning.

B. Ablation: Generalized-Mean Power p (Extended)

Results are in Table 4 (Section 4.2). Setup: $p \in \{1, 2, 5, 10, 20, 50, 100, 200\}$ across three datasets with 5 seeds each, holding all other hyperparameters fixed (MLP encoder, d=256, batch size 512, K=6 views, 500 epochs). Downstream: frozen linear probe.

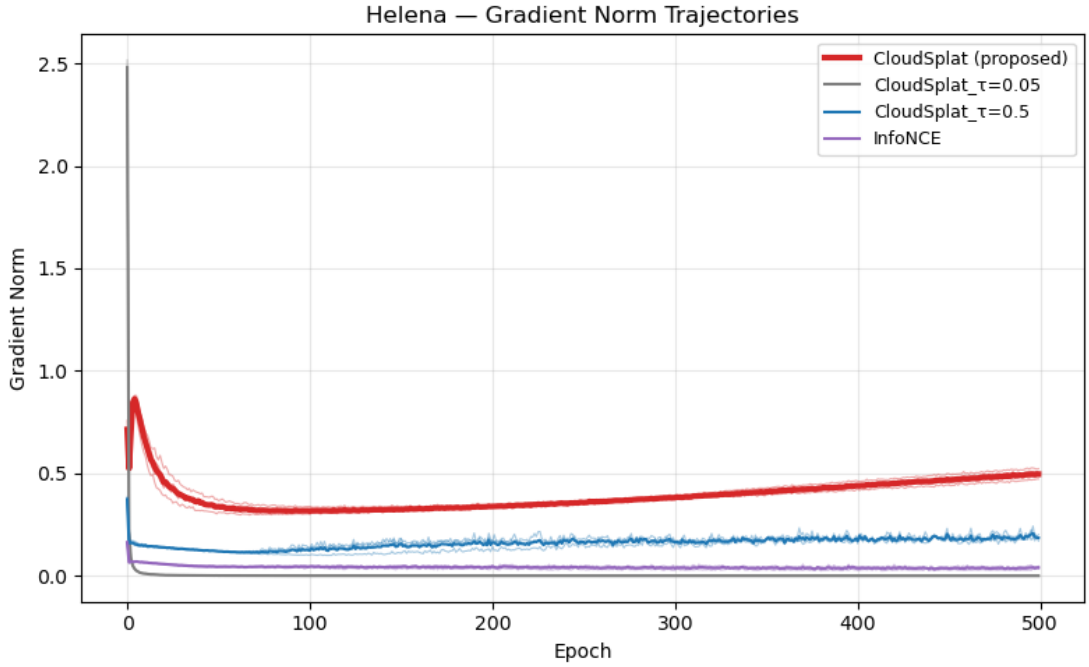
The slight decline at p=200 (-0.5 to -1.9pp vs. optimum) indicates that extremely large p over-focuses on individual outlier positions, reducing gradient diversity. Seed-to-seed variance also decreases with p: on Adult, std drops from 0.015 (p=2) to 0.003 (p=100); on MNIST, from 0.042 to 0.010. The per-dataset optima (p=50 for Adult and MNIST, p=100 for Helena) are consistent with our Bayesian HPO selecting p=100.

C. Training Stability and Collapse Analysis

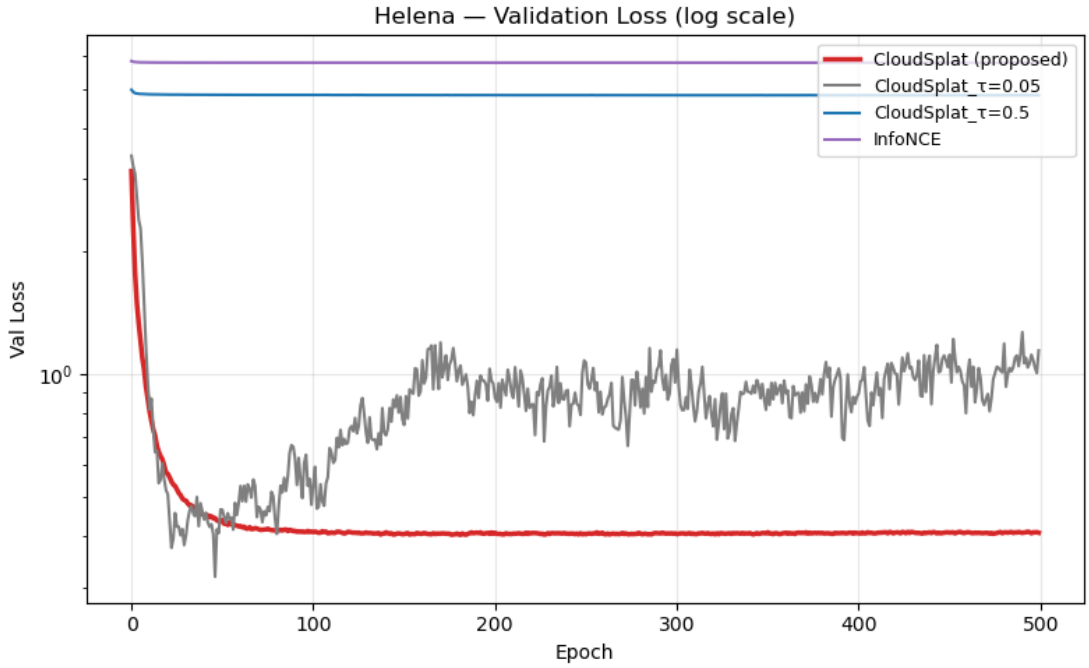
D. SCARF Comparison: Per-Dataset Results

CLOUDSPLAT (frozen encoder) outperforms SCARF (Bahri et al., 2021) (end-to-end fine-tuned) on 7/11 OpenML-CC18 datasets, with notable gains on diabetes ($+7.8\%$), electricity ($+5.2\%$), and car ($+2.5\%$). The main failure is mice_protein (89.8 vs. 98.8), attributable to its small size (n=1,080) and high class count (8 classes) starving the positive distribution.

E. Notation



(a) Gradient norm



(b) Validation loss (log scale)

Figure 11. **CloudSplat maintains gradient signal and loss convergence.** InfoNCE and τ=0.5 produce near-zero gradients; τ=0.05 is unstable.

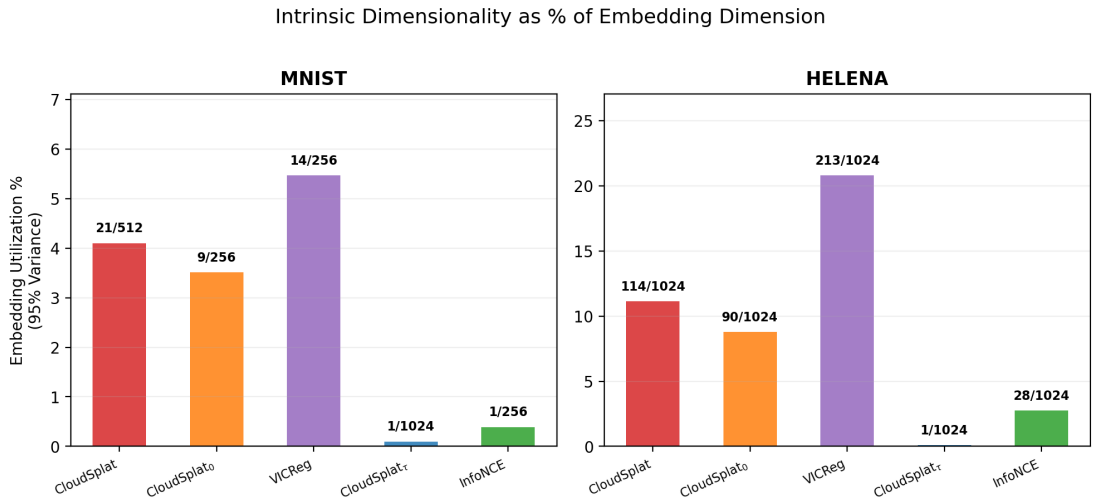


Figure 12. **Collapse analysis:** PCA components for 95% variance as percentage of embedding dimension d. Without VICReg regularization, CLOUDSPLAT$_\tau$ and InfoNCE collapse to 1 dimension. CLOUDSPLAT uses VICReg to prevent collapse (21/512 on MNIST, 114/1024 on Helena) while its distributional contrastive objective ensures the utilized dimensions remain discriminative (Table 3).

Table 5. **CLOUDSPLAT is accurate**: Test accuracy (%) on OpenML-CC18 datasets. CLOUDSPLAT wins on 7/11 completed datasets. Best and 2nd best per dataset. SCARF variants: contrastive (SCARF), autoencoder-based (SCARF AE = SCARF corruption + reconstruction; AE + Gauss. = Gaussian noise + reconstruction; AE clean = no corruption), and discriminative (SCARF disc. = binary clean-vs-corrupt classifier).

ID	Dataset	No pretrain	SCARF	SCARF AE	AE + Gauss.	AE (clean)	SCARF disc.	XGBoost	CLOUDSPLAT ‡
6	letter	94.69	95.75	94.60	94.56	94.36	95.10	88.01	96.08
28	optdigits	97.56	97.88	97.63	97.76	97.44	97.55	97.29	96.44
31	credit_g	73.55	73.86	73.30	72.68	72.92	71.92	76.38	76.00
32	pendigits	98.95	99.25	98.81	98.82	98.89	98.90	98.39	99.00
37	diabetes	75.21	75.36	75.15	75.58	75.57	74.58	76.47	83.12
44	spambase	93.40	94.46	93.53	93.45	93.49	93.55	94.54	95.11
151	electricity	82.77	83.55	82.69	82.67	82.34	83.18	83.27	88.79
1461	connect_4	83.38	82.97	83.29	83.17	83.26	83.17	74.11	83.51
40966	mice_protein	98.44	98.77	98.21	98.46	98.68	97.87	97.05	89.81
40975	car	95.27	97.26	96.86	97.02	96.76	96.70	95.87	99.71
40984	segment	90.56	91.97	90.32	90.46	90.53	90.69	92.66	93.51
CLOUDSPLAT wins		7 / 11 datasets							

‡ CLOUDSPLAT uses frozen encoder + downstream MLP classifier on fixed embeddings. All other methods use end-to-end fine-tuning of encoder + 2-layer MLP head.

Table 6. Symbols and notation used throughout.

Symbol	Description
B	Batch size
K	Number of views (1 clean + $K-1$ corrupted)
d	Input feature dimensionality
D	Embedding dimensionality
$s_{i,j}$	Cosine similarity between embeddings i and j
τ	Temperature (standard InfoNCE)
$\tau_{\text{num}}, \tau_{\text{denom}}$	Decoupled temperatures (positive, negative)
$d^{+}_{(k)}, d^{-}_{(k)}$	k-th sorted positive/negative distance
m	Margin in CLOUDSPLAT loss
p	Generalized mean power
N_{pairs}	Matched cardinality (B or $\binom{K}{2}B$)
ρ	Base corruption rate
σ	Corruption spread across views
η	Noise scale for Gaussian perturbation
$\lambda_{\text{var}}, \lambda_{\text{cov}}$	VICReg regularization weights
F_{+}, F_{-}	CDFs of positive/negative distances
F_{+}^{-1}, F_{-}^{-1}	Quantile functions of positive/negative distances
$\mathcal{W}_p^m$	Penalized Wasserstein functional (Eq. 4)
$\alpha(L)$	Margin-violation fraction

Algorithm 2 MultiCloudSplat: Multi-View Distributional Margin Loss

Require: Batch of latent representations $\mathbf{H} = \{h_i\}_{i=1}^{B} \in \mathbb{R}^{B\times D}$; number of views K; base corruption rate ρ; corruption spread σ; noise scale η; margin m; power p; VICReg weights $\lambda_{\text{var}}, \lambda_{\text{cov}}$

Ensure: Scalar loss $\mathcal{L}_{\text{MultiCSplat}}$

// Step 1: Generate K views (1 clean + $K-1$ corrupted)
$\mathbf{V}^{(0)} \leftarrow \mathbf{H}$ ▷ *Clean view*
for $k = 1\ K-1$ **do**
  if $K = 2$ **then**
    $\alpha_k \leftarrow \rho$ ▷ *Single corrupted view: use base rate*
  else
    $\alpha_k \leftarrow \rho \cdot \left(1 - \sigma + \frac{2\sigma(k-1)}{K-2}\right)$ ▷ *Rates in* $[\rho(1-\sigma),\ \rho(1+\sigma)]$
  end if
  $\mathbf{M}^{(k)} \sim \text{Bernoulli}(\alpha_k)^{B\times D}$ ▷ *Binary corruption mask*
  $\boldsymbol{\epsilon}^{(k)} \sim \mathcal{N}(0, \eta^2\mathbf{I})^{B\times D}$ ▷ *Gaussian noise*
  $\mathbf{V}^{(k)} \leftarrow \mathbf{H} + \mathbf{M}^{(k)} \odot \boldsymbol{\epsilon}^{(k)}$ ▷ *Corrupted view*
end for

// Step 2: Encode all views through shared backbone and projection head
for $k = 0\ K-1$ **do**
  $\mathbf{Z}^{(k)} \leftarrow g_\phi\big(f_\theta(\mathbf{V}^{(k)})\big) \in \mathbb{R}^{B\times D'}$ ▷ *f_θ: encoder, g_ϕ: projection head*
  $\hat{z}_i^{(k)} \leftarrow z_i^{(k)} \ / \ \|z_i^{(k)}\|_2 \quad \forall\, i \in \{1, \ldots, B\}$ ▷ *Row-wise ℓ_2-normalization*
end for

// Step 3: Positive distances over all $\binom{K}{2}$ view pairs
$\mathcal{D}_{\text{pos}} \leftarrow \varnothing$
for $a = 0\ K-1$ **do**
  for $b = a+1\ K-1$ **do**
    for $i = 1\ B$ **do**
      $\mathcal{D}_{\text{pos}} \leftarrow \mathcal{D}_{\text{pos}} \cup \big\{\|\hat{z}_i^{(a)} - \hat{z}_i^{(b)}\|^2\big\}$ ▷ *Same sample, different views*
    end for
  end for
end for
$|\mathcal{D}_{\text{pos}}| = \binom{K}{2} \cdot B$

// Step 4: Negative distances (clean vs. first corrupted view)
$\mathcal{D}_{\text{neg}} \leftarrow \big\{\|\hat{z}_i^{(0)} - \hat{z}_j^{(1)}\|^2 : i \neq j\big\}$ ▷ *$B(B-1)$ cross-sample pairs*
Subsample $|\mathcal{D}_{\text{pos}}|$ elements from $\mathcal{D}_{\text{neg}}$ uniformly ▷ *Match cardinalities for W_p*

// Step 5: Sort-based exact 1D Wasserstein with margin
$\mathbf{d}^+ \leftarrow \text{Sort}(\mathcal{D}_{\text{pos}})$ ▷ *Ascending order*
$\mathbf{d}^- \leftarrow \text{Sort}(\mathcal{D}_{\text{neg}})$ ▷ *Ascending order*
$\mathcal{L}_{\text{CSplat}} \leftarrow \left(\frac{1}{|\mathcal{D}_{\text{pos}}|}\sum_{k=1}^{|\mathcal{D}_{\text{pos}}|}\big[\text{softplus}(\,d_k^+ - d_k^- + m)\big]^p\right)^{1/p}$ ▷ *Generalized mean, W_p norm*

// Step 6: VICReg regularization (averaged over K views, on unnormalized Z)
$\mathcal{L}_{\text{var}} \leftarrow \frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{D'}\sum_{d=1}^{D'}\max\left(0,\ 1 - \sqrt{\text{Var}[\mathbf{Z}_{:,d}^{(k)}] + \epsilon}\right)$
$\mathcal{L}_{\text{cov}} \leftarrow \frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{D'^2}\sum_{d\neq d'}\big[\text{Cov}(\mathbf{Z}^{(k)})_{d,d'}\big]^2$

// Total loss
$\mathcal{L}_{\text{MultiCSplat}} \leftarrow \mathcal{L}_{\text{CSplat}} + \lambda_{\text{var}} \cdot \mathcal{L}_{\text{var}} + \lambda_{\text{cov}} \cdot \mathcal{L}_{\text{cov}}$
$\mathcal{L}_{\text{MultiCSplat}}$