UNSUPERVISED DOMAIN SHIFT DETECTION VIA DISCREPANCY BETWEEN POINTWISE AND PROTOTYPE-REFINED PSEUDO-LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study unsupervised domain-shift detection for image classifiers in the realistic setting where target labels are unavailable. We propose a simple detector based on the discrepancy between two label assignments on unlabeled target data: the classifier's pointwise pseudo-labels and prototype-refined pseudo-labels obtained through batch-level prototype refinement in feature space. Our key observation is that these two labelings largely agree when the source classifier remains well aligned with the target domain, but diverge under harmful shift, when the source decision boundary becomes mismatched with the target structure. This leads to a shift detection score that can be computed directly from a deployed model, without training an auxiliary detector or explicitly estimating source–target divergence. Experiments on standard visual domain-shift benchmarks show that the proposed score accurately detects batch-level shift and tracks accuracy degradation better than confidence-based and distribution-based baselines. These results suggest that pseudo-label self-consistency provides a simple and effective criterion for detecting performance-relevant distribution shift. Our code is publicly available: `https://anonymous.4open.science/r/PLSC-50BF/`.

1 INTRODUCTION

Modern image classifiers are often deployed on data whose distribution differs from the source domain on which the model was trained. Changes in style, acquisition conditions, background statistics, or population composition can substantially degrade predictive accuracy. Domain shift is a common problem, and the resulting performance degradation can be costly in many real-world ML applications, including computer vision, medical imaging, autonomous driving, remote sensing, speech/audio, and industrial monitoring (see e.g. (Roschewitz et al., 2025; Guan et al., 2025; Frenkel et al., 2026)). In many realistic deployments target labels are unavailable, making it difficult to determine when such degradation is likely to occur. This motivates the problem of unsupervised domain-shift detection: given only unlabeled target data and a trained source model, determine whether the target domain has shifted in a way that is likely to harm predictive performance.

A large body of prior work detects domain shift by measuring discrepancies between source and target distributions, either in input space or in learned feature representations. Representative approaches include kernel-based two-sample tests, density-ratio estimation, and feature-space divergence measures (Gretton et al., 2012; Lipton et al., 2018; Rabanser et al., 2019). While effective at identifying changes in the marginal distribution $p(x)$, these methods may fail to capture performance-relevant shifts that compromise the model's learned decision boundaries, which are typically the primary cause of performance degradation (Rabanser et al., 2019). As a result, distribution-level detectors may signal a substantial shift even when model accuracy is preserved, or fail to detect shifts that critically affect predictions (Rabanser et al., 2019). An alternative strategy is to rely on model-based signals such as prediction confidence or predictive uncertainty (Hendrycks & Gimpel, 2017; Ovadia et al., 2019). However, deep neural networks are often poorly calibrated under distribution shift and may remain highly confident even when making incorrect predictions (Ovadia et al., 2019). As a result, uncertainty-based signals may not reliably indicate when a model is failing on a new domain.

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.

In this work, we take a self-consistency perspective on domain-shift detection. Rather than asking only whether the target distribution differs from the source distribution, we ask whether the source model remains self-consistent on the target data. Our method compares two target-side label assignments that encode different assumptions about the data. The first is the model's standard pseudo-label, which reflects the source-trained decision boundary. The second is a prototype-refined label, obtained by structure-aware refinement in target feature space, for example using centroid-based class refinement (Liang et al., 2020). In SFDA, refined pseudo-labels are typically used as supervision for target-model adaptation; in our approach, the refinement-induced change from pseudo-labels to prototype-refined pseudo-labels is itself the diagnostic signal. When the source model transfers well to the target domain, pseudo-labels and prototype-refined pseudo-labels tend to agree. A harmful domain shift induces misalignment between the source decision boundary and the class structure of the target domain. Under such shifts, structure-aware refinement can partially correct systematic pseudo-label errors, increasing disagreement between the two labelings. We therefore use the discrepancy between pseudo-labels and prototype-refined pseudo-labels as a direct signal of model–data mismatch. Unlike conventional shift detectors that ask whether the target distribution differs from the source distribution, our detector asks whether the target-domain geometry implies a different decision rule from the one encoded by the source classifier. This viewpoint offers two advantages. First, it targets shifts that are relevant to downstream performance rather than merely generic distributional change. Second, it is simple to deploy: it requires only a trained source model, a held-out unlabeled source reference set, and an unlabeled target batch, without target labels, model retraining, or an auxiliary detector.

We evaluated the proposed approach on standard visual domain-shift benchmarks spanning both real-world and synthetic shifts. We compared against representative distribution-based and confidence-based baselines, including two-sample tests in feature and prediction space. Across datasets, the proposed discrepancy score provides strong shift detection performance and correlates more closely with accuracy degradation than competing methods. These results suggest that pseudo-label self-consistency is an effective criterion for detecting performance-relevant domain shift in unlabeled target domains.

Our contributions are threefold. First, we introduce a new diagnostic use of pseudo-label refinement: rather than using prototype-refined pseudo-labels to adapt the target model as in SFDA, we use the discrepancy between the source classifier's pointwise predictions and the refined assignments as a measure of model–data misalignment. Second, we turn this refinement-induced discrepancy into a statistically calibrated batch-level shift detector using a source–target permutation test. Third, we show on standard visual domain-shift benchmarks that our algorithm achieves strong detection performance and better tracks accuracy degradation than representative distribution-, confidence-, and classifier-based baselines.

2 Related Work

Domain shift detection has been studied from several closely related perspectives, most notably out-of-distribution (OOD) detection, statistical shift testing, and domain adaptation. In the OOD literature, a common strategy is to assign each test sample a score based on the model's predictive confidence. Early work showed that maximum softmax probability already provides a surprisingly strong baseline for detecting misclassified and out-of-distribution examples (Hendrycks & Gimpel, 2017). Subsequent methods improved this paradigm using temperature scaling and input perturbations (Liang et al., 2018), as well as energy-based scores derived from the classifier logits (Liu et al., 2020). While effective, these approaches are primarily designed for *sample-level* OOD detection rather than for detecting whether an entire target domain has shifted relative to the source.

A complementary line of work formulates shift detection as a statistical testing problem between source and target samples. In this view, the goal is to determine whether a batch of target examples is drawn from the same distribution as the training data. A particularly influential study in this direction is that of Rabanser et al. (2019), which systematically compared methods for dataset shift detection and showed that low-dimensional summaries based on model outputs can already provide strong detection performance. This perspective is especially relevant in practical settings where one wishes to detect domain-level changes without requiring access to target labels.

Another line of work detects shift by training a domain classifier to distinguish reference samples from test samples (Jang et al., 2022) and using the accuracy of this classifier as a proxy for measuring

distances between distributions. A major drawback of this approach is its high computational cost: for every test set, a new domain classifier must be trained, which is highly impractical in continuous monitoring scenarios.

A related line of work addresses unsupervised accuracy estimation under distribution shift. Difference of Confidences (DoC) estimates the change in accuracy from the difference between the model's average confidence on the source and target domains, while Average Thresholded Confidence (ATC) learns, using labeled source data, a confidence threshold whose exceedance rate is used to estimate target accuracy (Guillory et al., 2021; Garg et al., 2022). More recently, MaNo derives an accuracy-related score from a data-dependently normalized matrix of target logits, without requiring target labels (Xie et al., 2024). These methods are related to our objective but address a different problem: they aim to produce a numerical estimate of target-domain accuracy, whereas our goal is to perform a statistical test of whether classifier–prototype alignment has significantly deteriorated relative to a source reference distribution. Moreover, unlike ATC and DoC, our approach does not require labeled source validation data or knowledge of source accuracy at the monitoring stage; it uses only an unlabeled source reference set and an unlabeled target batch.

Existing methods for shift detection typically measure distributional discrepancy or rely on confidence-based surrogates, while source-free adaptation methods use pseudo-label refinement to improve target performance. In source-free domain adaptation, prototype refinement is typically used to generate improved pseudo-labels for updating the target model; the discrepancy introduced by the refinement itself is not treated as a diagnostic quantity. In contrast, our method uses the discrepancy between pseudo-labels and prototype-based pseudo-labels itself as a shift signal, thereby targeting model–data misalignment that is directly relevant to downstream performance.

3 Pseudo-Label Self-Consistency

Let g_s be a classifier trained on a labeled source domain, and let $\mathcal{X}_t = \{x_i\}_{i=1}^n$ be an unlabeled batch of target samples observed at deployment time. Our goal is to determine whether the target batch exhibits a domain shift that is also harmful, namely a shift that makes the source classifier substantially less accurate on the target data. Unlike standard domain shift detectors that measure only distributional discrepancy between source and target samples, we seek a signal that is more directly tied to model-data mismatch and potential performance degradation. Because target labels are unavailable at deployment, this degradation cannot be observed directly. Our method compares two label assignments on the target batch. The first is the classifier's standard pseudo-label, computed independently for each sample. The second is a pseudo-label obtained by refining these assignments using the global structure of the target batch in feature space. Our central hypothesis is that these two labelings remain largely consistent when the source classifier is well aligned with the target domain, and diverge under harmful shift.

Pseudo-Labels (PL). We write the source model as $g_s = h \circ f$, where $f(x) \in \mathbb{R}^d$ is the feature representation produced by the backbone and $h(f(x)) = p(x) \in \Delta^{K-1}$ is the vector of predicted class probabilities over K classes. We denote the k-th predicted class probability by $p_k(x) = p(y = k \mid x)$. The standard pseudo-label of a sample x is $y_{\text{PL}}(x) = \arg\max_k p_k(x)$.

Prototype-Refined Pseudo-Labels (PPL). For each class $k \in \{1, \ldots, K\}$, define a soft class prototype by a probability-weighted average of target features:

$$c_k = \frac{\sum_{i=1}^n p_k(x_i)\, f(x_i)}{\sum_{i=1}^n p_k(x_i)}.$$

These prototypes summarize the class-conditional structure induced by the source model on the target batch. Following standard feature extraction, the penultimate feature vectors $f(x)$ are used in their raw, unnormalized form. Each sample is then reassigned according to the prototype to which its feature representation is most similar. Using Euclidean distance, the refined pseudo-label is defined as

$$y_{\text{PPL}}(x_i) = \arg\min_k \|f(x_i) - c_k\|.$$

The main distinction between PL and PPL is that $y_{\text{PL}}(x_i)$ is a pointwise decision based only on x_i, whereas $y_{\text{PPL}}(x_i)$ depends on the full target batch through the prototypes $\{c_k\}_{k=1}^K$. The PPL procedure

is inspired by the prototype-refinement step in SHOT (Liang et al., 2020), which was introduced in source-free domain adaptation to reduce pseudo-label noise in the adaptation objective. In SHOT, the motivation was to denoise pseudo-labels used to update the target network via gradient descent. We use the refinement step for a different purpose: to measure the disagreement between PL and PPL.

Pseudo-label self-consistency as a shift signal. The PL-PPL agreement can be viewed as measuring the consistency between the decision rule encoded by the fixed source classifier and the class structure induced by the observed feature geometry. When the source model remains well aligned with the data, these two views tend to agree: the classifier predictions and the prototype-based assignments yield similar labels. A harmful domain shift can disrupt this classifier-prototype alignment, causing groups of samples to be systematically assigned differently by the two rules. In particular, when the source decision boundary becomes misaligned with the target feature structure, PPL can exploit the target-batch geometry to partially correct structured pseudo-label errors, thereby increasing PL-PPL disagreement. Consequently, the discrepancy between PL and PPL provides a direct signal of model-data misalignment. This differs from prior SFDA methods, which use PPL primarily as an intermediate representation for adaptation; here, the refinement-induced change itself is treated as a diagnostic quantity for harmful domain shift. Appendix B provides an additional geometric interpretation of this behavior through neural collapse.

Hypothesis test for harmful shift. We next quantify the refinement-induced change through the PL–PPL agreement rate. High agreement indicates that the source decision rule remains compatible with the target feature geometry, whereas reduced agreement indicates model–data misalignment. To determine whether the target agreement is abnormally low, we compare it to a reference agreement computed on held-out source validation data.

Let $\mathcal{X}_s = \{x_i^s\}_{i=1}^n$ be an unlabeled source validation set. Applying the same PL-PPL construction to $\mathcal{X}_s$, we define the source-domain reference agreement

$$A(\mathcal{X}_s) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{y_{\text{PL}}(x_i^s) = y_{\text{PPL}}(x_i^s)\}\,. \tag{1}$$

This quantity provides an empirical reference for PL–PPL agreement under the source distribution. For the target batch $\mathcal{X}_t = \{x_i^t\}_{i=1}^n$, we define the target-domain agreement

$$A(\mathcal{X}_t) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\big\{y_{\text{PL}}(x_i^t) = y_{\text{PPL}}(x_i^t)\big\}\,. \tag{2}$$

We test whether the target agreement is significantly lower than the source reference agreement. We form the directional test statistic:

$$T = A(\mathcal{X}_s) - A(\mathcal{X}_t). \tag{3}$$

A large positive value means that PL–PPL agreement is substantially lower in the target domain. We define the hypotheses as H_0: the source and target domains are exchangeable ($P_s = P_t$), and H_1: the target domain exhibits a reduction in PL–PPL self-consistency relative to the source domain, indicating a potentially harmful model–data mismatch.

We evaluate T using a Monte-Carlo permutation test conditional on the pooled sample. Under H_0, we create a pooled set $X_{\text{pool}} = \mathcal{X}_s \cup \mathcal{X}_t$ and randomly reassign the observations into two pseudo-groups of sizes n for B permutations. Crucially, for each permutation b, we recompute the group-specific probability-weighted prototypes and PPLs from scratch to automatically reproduce the dependence induced by prototype estimation. This yields the permuted agreement rates $A_s^{(b)}$ and $A_t^{(b)}$, forming the permuted statistic $T_b = A_s^{(b)} - A_t^{(b)}$.

The resulting one-sided Monte Carlo p-value evaluates the upper tail of this permutation distribution:

$$p\text{-value} = \frac{1 + \sum_{b=1}^{B} \mathbf{1}\{T_b \geq T\}}{B+1}.$$

The $+1$ correction avoids reporting a zero p-value merely because no sampled permutations exceeded the observed statistic. A significant p-value means the observed reduction in self-consistency is unusually large relative to random chance under distributional equality. We flag a potentially harmful shift whenever this p-value falls below a predefined threshold α (e.g., $\alpha = 0.05$). We denote the

Algorithm 1 Pseudo-Label Self-Consistency (PLSC) Permutation Test

Input: Source data $\mathcal{X}_s$, target data $\mathcal{X}_t$, model g_s, number of permutations B.
Output: p-value for a reduction in PL–PPL self-consistency.

1: Extract feature representations $f(x)$ and class probabilities $p_k(x)$ using g_s for all $x \in \mathcal{X}_s \cup \mathcal{X}_t$.
2: Compute observed agreement score: $T = A(\mathcal{X}_s) - A(\mathcal{X}_t)$.
3: For $b = 1 \dots B$, randomly split $\mathcal{X}_s \cup \mathcal{X}_t$ into pseudo-groups $\mathcal{X}_s^{(b)}$ and $\mathcal{X}_t^{(b)}$, and compute the permuted statistic:

$$T_b = A(\mathcal{X}_s^{(b)}) - A(\mathcal{X}_t^{(b)}).$$

4: Compute the p-value using a Monte-Carlo permutation distribution:

$$p\text{-value} = \frac{1 + \sum_{b=1}^{B} \mathbf{1}(T_b \geq T)}{B + 1}.$$

5: Flag a potentially harmful shift if p-value $< \alpha$.

proposed algorithm Pseudo-Label Self-Consistency (PLSC). The full procedure is summarized in Algorithm 1.

PLSC has three practical advantages. First, it combines prediction-space and feature-space information: PL is derived from classifier outputs, whereas PPL depends on the geometry of the full target batch. Second, it targets performance-relevant shift rather than generic marginal drift. Third, it is simple to deploy: it requires only a trained source model, a held-out unlabeled source reference set, and an unlabeled target batch, without target labels, model retraining, or an auxiliary detector.

From a computational perspective, the additional cost beyond a forward pass through the source model is modest. Once features and predicted probabilities are available, prototype construction is linear in batch size, and PPL assignment requires only comparisons to K class prototypes. This makes PLSC suitable for batch-level monitoring in deployment settings.

4 EXPERIMENTS

In this section, we compare our domain shift detection procedure to existing methods using standard publicly available domain shift benchmarks.

Datasets. We evaluated our method on several widely used domain shift benchmarks, which are commonly used for evaluating domain adaptation methods: Office-Home (Venkateswara et al., 2017), VisDA (Peng et al., 2017), and DomainNet (Peng et al., 2019). The Office-Home dataset consists of four domains, each containing 65 categories. These domains include: Art (A), which contains artistic depictions such as sketches and paintings; Clipart (C), composed of clipart-style images; Product (P), featuring object images with minimal or no background; and Real-World (R), which includes images captured using standard cameras. VisDA is a large-scale synthetic-to-real benchmark for domain adaptation, comprising over 280,000 images spanning 12 categories. DomainNet is an extensive multi-domain dataset covering common object classes. Due to annotation noise, we follow prior work (Zhang et al., 2023; Diamant et al., 2024) and use the 126-class subset, referred to as DomainNet126. It contains four domains: Clipart (C), Product (P), Real (R), and Sketch (S). The Office-31 dataset (Saenko et al., 2010), which contains 31 object categories across three domains: Amazon (A), DSLR (D), and Webcam (W). These categories include common office items such as keyboards, file cabinets, and laptops.

Compared Domain-Shift Detection Methods. We compare PLSC with representative baselines from three families of shift-detection methods: kernel two-sample testing, one-dimensional distributional testing, and classifier-based two-sample testing. Maximum Mean Discrepancy (MMD) (Kore et al., 2024; Dreiseitl, 2022) is a non-parametric metric for measuring the distance between two probability distributions based on their samples. It quantifies the difference between the mean embeddings of the source and target distributions in a reproducing kernel Hilbert space (RKHS), where larger values indicate greater divergence. Following Gretton et al. (2012) we assessed statistical significance using a permutation test. After computing the observed MMD, we repeatedly shuffled and repartitioned the pooled data to obtain surrogate MMD values. The p-value was estimated as $p = \frac{k+1}{N+1}$, where k counts

permutations with MMD greater than or equal to the observed value. The Kolmogorov-Smirnov (KS) test (Merkow et al., 2023; Poenaru-Olaru et al., 2024; Ginart et al., 2022) is a non-parametric statistical test used to compare two one-dimensional continuous distributions by measuring the maximum deviation between their cumulative distribution functions. It produces a p-value to determine whether the distributions differ significantly. Since both feature representations and soft prediction scores are multi-dimensional, we apply the KS test with Bonferroni correction over the resulting p-values to obtain a final shift score. The Classifier Two-Sample Test (C2ST) (Koch et al., 2024; Guan et al., 2025) operates by training a classifier to distinguish between samples from two distributions. The classifier's accuracy is compared to the expected accuracy under random guessing (i.e., chance level) using a binomial test, allowing the method to produce a p-value. In our experiments, we use this p-value (with $p_{\text{null}} = 0.5$) as the decision criterion for detecting distribution shift. For each baseline method, we evaluate performance under four configurations: (1) the model's feature representation, (2) the model's soft predictions, (3) the confidence scores, and (4) the entropy of the soft predictions. The latter three correspond to black-box shift-detection (BBSD) signals derived from model outputs (Rabanser et al., 2019; Lipton et al., 2018).

Implementation Details. In our experiments, all baseline methods were compared with the proposed PLSC approach across standard domain adaptation benchmarks. For each dataset, both no-shift and shift scenarios were considered. The source data was first partitioned into three independent train/validation splits, and a separate model was trained for each split. For each such split, the validation portion was further randomly divided 20 times into two disjoint subsets, forming the source-train (reference) and source-test (in-domain test) sets. For each of these configurations, a target-test set was sampled from the target domain to represent shifted data. This procedure resulted in a total of 60 triplets of (source-train, source-test, target-test) per source–target pair, enabling repeated and statistically robust evaluation. The evaluation set size ranged from a few hundred to several thousand samples, depending on the dataset and the severity of the shift. A ResNet architecture was used as the underlying model (ResNet-101 for VisDA and ResNet-50 for all other datasets), and intermediate feature representations extracted from this network were used within PLSC to construct PPLs. For C2ST, the classifier was trained using 5-fold cross-validation, with results averaged across the folds. For high-dimensional representations (e.g., features and softmax outputs), the KS test was applied with a Bonferroni correction over the resulting p-values. Since all methods produced p-values, a uniform decision threshold of $p = 0.05$ was used throughout. Given precomputed model logits and features, all statistical tests were executed efficiently on CPU (in our case, an M4 chip). In practice, PLSC exhibited negligible computational overhead, typically requiring well under one second per evaluation.

Evaluation Measures. We evaluate each method using two complementary metrics. First, we report the F1-score at the fixed detection threshold, which measures the balance between false positives and false negatives in the binary shift-detection task. Second, we report the absolute Spearman correlation between each method's shift score and the model accuracy computed using ground-truth labels across both source-test and target-test batches. This metric evaluates whether the shift score tracks downstream performance degradation. All results are averaged over 60 sampled (source-train, source-test, target-test) triplets for each domain shift.

Table 1: F1-score (%) achieved by different shift detection methods across various datasets.

Method	DomainNet126	OfficeHome	VisDA	Office31	Avg.
KS features	98.0 (1.0)	98.5 (1.1)	97.6 (0.0)	98.4 (1.5)	**98.1 (1.2)**
KS softmax	98.4 (1.0)	97.9 (1.7)	**100.0 (0.0)**	97.6 (2.1)	98.5 (1.6)
KS confidence	97.7 (1.6)	97.5 (1.7)	98.4 (0.0)	97.8 (0.8)	97.9 (1.4)
KS entropy	96.8 (1.4)	97.8 (1.2)	98.4 (0.0)	97.2 (1.9)	97.5 (1.5)
MMD features	82.4 (15.7)	81.9 (12.9)	66.7 (0.0)	75.6 (10.8)	76.7 (13.1)
MMD softmax	97.0 (1.5)	98.5 (0.9)	**100.0 (0.0)**	97.8 (1.0)	98.3 (1.1)
MMD confidence	91.0 (20.4)	85.1 (30.2)	95.2 (0.0)	97.4 (0.8)	92.2 (17.1)
MMD entropy	87.1 (32.7)	81.1 (37.9)	95.2 (0.0)	97.0 (2.1)	90.1 (24.2)
C2ST features	95.8 (1.0)	95.7 (1.7)	96.8 (0.0)	96.2 (1.9)	96.1 (1.5)
C2ST softmax	94.9 (1.6)	94.9 (2.7)	96.0 (0.0)	89.4 (4.4)	93.8 (2.9)
C2ST confidence	96.7 (2.0)	97.7 (1.3)	96.8 (0.0)	97.8 (1.4)	97.2 (1.5)
C2ST entropy	96.4 (1.0)	97.2 (0.9)	98.4 (0.0)	96.8 (2.1)	97.2 (1.3)
PLSC	**98.9 (1.0)**	**98.7 (1.2)**	**100.0 (0.0)**	**98.4 (1.3)**	**99.0 (1.2)**

Table 2: Absolute Spearman correlation between p-values and ground-truth model accuracy (%) for different shift detection methods across multiple datasets.

Method	DomainNet126	OfficeHome	VisDA	Office31	Avg.
KS features	80.4	79.0	79.9	84.6	81.0
KS softmax	76.6	86.1	81.2	84.6	82.1
KS confidence	69.7	83.7	77.2	81.1	77.9
KS entropy	72.2	83.7	77.6	81.6	78.8
MMD features	58.9	59.7	2.0	64.2	46.2
MMD softmax	79.1	82.8	85.2	82.1	82.3
MMD confidence	80.9	73.1	73.4	77.3	76.2
MMD entropy	76.7	67.4	74.4	77.8	74.1
C2ST features	82.0	83.8	80.2	68.6	78.6
C2ST softmax	81.5	**85.0**	**86.4**	79.3	83.1
C2ST confidence	82.0	83.2	77.0	74.3	79.1
C2ST entropy	80.8	83.2	77.0	74.1	78.8
PLSC	**85.4**	83.2	81.3	**88.2**	**84.5**

Results. Table 1 reports F1-scores for shift detection under a fixed decision threshold. Several baselines achieve strong performance, including KS, as well as several C2ST variants. However, PLSC obtains the highest overall performance, with an average F1-score of 99.0%, while maintaining near-perfect results across datasets. In contrast, several competing methods exhibit substantially higher variability. MMD-based methods show inconsistent behavior: while MMD on softmax predictions remains competitive, MMD on features and confidence scores degrades substantially and exhibits high variance. Overall, PLSC provides the most consistent and robust threshold-based detection performance without dataset-specific threshold tuning. Table 2 presents the absolute Spearman correlation between the produced p-values and the ground-truth model accuracy. For this analysis, only the three original source splits (each associated with a separately trained model) were used, without the additional resampling procedure. PLSC achieves the highest average correlation (84.5%), indicating that its scores are most strongly aligned with accuracy degradation. The closest competing method is C2ST applied to softmax outputs. This finding supports the central premise of PLSC: rather than detecting arbitrary distributional differences, it focuses on shifts that are predictive of downstream performance degradation, i.e., practically harmful domain shifts.

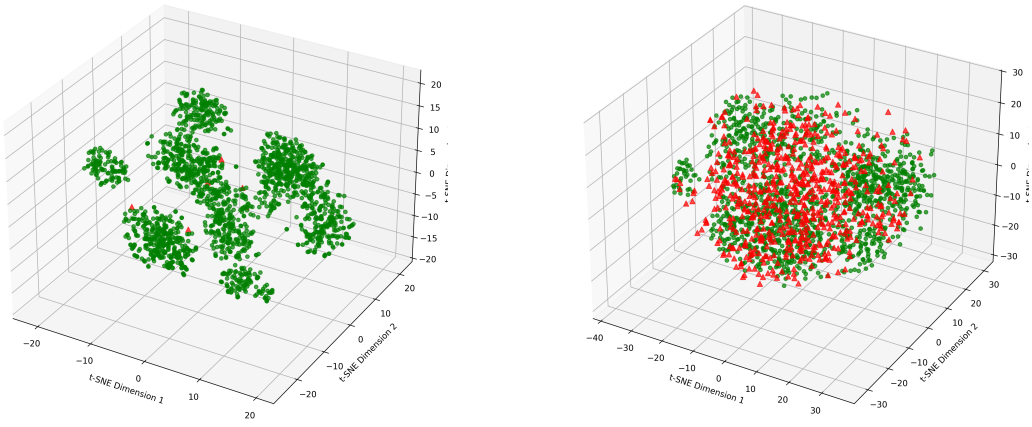
While Tables 1 and 2 establish the overall detection performance and accuracy alignment of PLSC across real domain shifts, we next use a controlled experiment to directly examine its central intended advantage: distinguishing performance-relevant shift from distributional change that does not harm the deployed model.

Controlled shift severity. Following the controlled-shift protocol of (Rabanser et al., 2019), we systematically analyze detector behavior under gradually increasing synthetic perturbations, rather than relying solely on fixed domain pairs. Specifically, we construct target datasets from the A, P, and R domains of OfficeHome by injecting Gaussian noise at varying intensities into the input images. For each noise level, we generate 20 datasets. We evaluate the proposed PLSC method against KS, MMD, and C2ST baselines, applied over feature representations, softmax outputs, confidence scores, and prediction entropy. We measure the average p-value for each noise level across the 20 datasets and examine its correlation with accuracy degradation. The results for domain R are reported in Table 3, with results for domains A and P deferred to the appendix.

The main observation from Table 3 is that PLSC responds to performance-relevant shift rather than merely to changes in the input distribution. At the lowest noise level, where accuracy is essentially unchanged, the mean PLSC p-value remains above the 5% threshold, whereas feature-based KS and C2ST already produce mean p-values below it. As corruption substantially degrades accuracy, the mean PLSC p-value falls below 5% at 9.1% degradation and remains low thereafter. This behavior illustrates the central advantage of PLSC: because it measures the consistency between the classifier's predictions and the geometry of the target data, its statistical signal reflects deterioration in model–data alignment. Generic distributional tests, by contrast, may either react to harmless covariate changes or remain insensitive to harmful ones, depending on the representation being tested. Although some prediction-based baselines detect smaller degradations earlier, PLSC provides a stable and directly model-relevant indication of harmful shift.

Table 3: Mean p-value (%) under varying Gaussian noise levels on the OfficeHome R domain, averaged over 20 trials.

Noise Level	0.015	0.02	0.03	0.04	0.05	0.075	0.1	0.125	0.15	0.175
Accuracy Degradation	-0.1	0.5	1.6	2.5	3.9	9.1	15.2	23.9	35.6	47.3
KS features	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KS softmax	64.0	6.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KS confidence	28.0	7.6	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KS entropy	26.8	6.9	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MMD features	100.0	100.0	100.0	100.0	99.5	48.9	7.4	5.3	1.9	1.0
MMD softmax	18.8	9.7	2.9	1.0	0.9	0.9	0.9	0.9	0.9	0.9
MMD confidence	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	88.2	94.6
MMD entropy	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
C2ST features	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2ST softmax	100.0	100.0	100.0	99.9	99.9	99.7	4.6	0.0	0.0	0.0
C2ST confidence	19.4	4.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2ST entropy	17.4	12.0	3.6	0.1	0.0	0.0	0.0	0.0	0.0	0.0
PLSC	33.3	26.1	13.0	14.6	7.1	0.2	0.1	0.1	0.1	0.1



(a) VisDA Source Domain

(b) VisDA Target Domain

Figure 1: Agreement between PL and PPL values. Green indicates agreement, while red indicates disagreement. The 3D points were produced by applying t-SNE to feature embeddings extracted by a model trained on the source-domain training data.

Visualization of the PL–PPL agreement. Our method relies on the observation that, in the source domain, PPLs closely align with the model's pseudo-labels, since incorporating global geometric structure offers little additional benefit when no shift is present. Under a harmful domain shift, however, this agreement decreases substantially. To illustrate this behavior, we apply t-SNE to image features from the VisDA dataset (Peng et al., 2017) and project them into three dimensions to visualize the consistency between PLs and PPLs. Figure 1 shows the samples for which PL and PPL agree (green) and those for which they differ (red). We show the same analysis for both the source and target domains. We can first see that, in the source domain, the classes are clearly separated, whereas in the target domain they are more mixed. Figure 1 also shows that PPLs and PLs are largely consistent in the source domain, whereas substantial disagreement emerges in the shifted target domain. This gap highlights the sensitivity of PL–PPL agreement as an indicator of a harmful distribution shift. Figure 2 shows the histogram of the differences in agreement percentages between the source and target domains across all the evaluated source-target pairs. We observe that the PL-PPL difference is consistently lower in the source domain than in the target domain.

Ablation study. We conducted an ablation study under the same protocol as the controlled shift-severity experiment to evaluate two key components of PLSC: (i) a hard-clustering variant in which prototypes are computed from discrete pseudo-label assignments rather than soft class probabilities, and (ii) the use of cosine distance instead of Euclidean distance. As shown in Table 4, the standard PLSC configuration (utilizing soft probabilities and Euclidean distance) is the most sensitive and

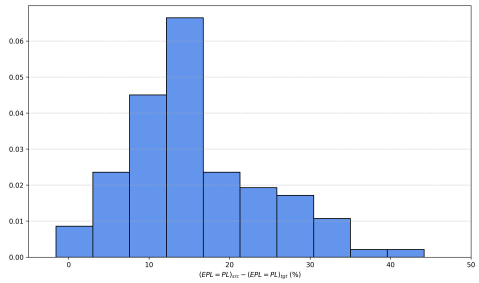


Figure 2: A density histogram displaying the difference in agreement percentages between the source and target domains across various datasets.

stable, consistently producing lower p-values that rapidly cross the significance threshold as accuracy degradation increases. In contrast, relying on hard clustering significantly degrades performance; it fails to confidently detect severe shifts until accuracy degradation reaches extreme levels (27.1%) and exhibits severe instability, with p-values fluctuating inconsistently across moderate noise levels. Similarly, substituting Euclidean distance with cosine distance reduces the test's sensitivity. While the cosine variant eventually detects heavy shifts, it consistently trails behind the standard PLSC method in identifying mild to moderate accuracy degradation.

Table 4: Average p-value ($\%$) for controlled shift detection on the OfficeHome domain A.

Noise Level	0.015	0.02	0.03	0.04	0.05	0.075	0.1	0.125	0.15	0.175
Accuracy Degradation	1.1	1.1	1.3	2.1	3.1	7.1	11.2	16.4	22.1	27.1
PLSC hard clustering	67.6	67.2	61.1	56.0	41.8	55.0	62.5	43.4	17.1	4.6
PLSC cosine	48.5	47.3	58.2	54.6	37.5	21.8	8.8	1.9	0.3	0.1
PLSC	29.0	28.6	26.4	16.6	4.3	0.9	0.3	0.1	0.1	0.1

5 Limitations

Our study has several limitations. First, although PLSC is evaluated on standard visual domain-shift benchmarks, all experiments are conducted in image-classification settings, and additional evaluation is needed to assess its effectiveness in other modalities and tasks. Second, the method relies on the quality of the source model's feature representation and on the assumption that target-domain class structure remains meaningful in this space. Finally, PLSC may miss harmful shifts when both PL and PPL fail in the same way, and it should therefore be viewed as a practical monitoring signal rather than a guarantee of target-domain reliability.

6 Conclusion

We presented an unsupervised method for harmful domain-shift detection based on the discrepancy between pointwise and prototype-refined pseudo-labels on unlabeled target data. The key idea is that agreement between pointwise pseudo-labels and batch-refined pseudo-labels indicates that the source classifier remains aligned with the target-domain structure, whereas growing disagreement reveals model–data mismatch and potential performance degradation. This turns pseudo-label refinement from an adaptation mechanism into a diagnostic tool for model reliability under distribution shift. The resulting detector requires no target labels at test time, and no additional supervision. More broadly, our results suggest that internal inconsistencies in unlabeled target predictions can help determine not only how to adapt a model, but also when adaptation is necessary.

AI use statement

In this work, we used generative AI tools in the preparation of this manuscript solely for language editing: identifying typographical errors and improving grammatical and stylistic clarity of the text. AI tools were not used to generate research ideas, formulate hypotheses, design the methodology or

experiments, perform data analysis, interpret results, or produce any part of the code or numerical results reported in this paper. All scientific content, claims, and conclusions are the authors' own.

REFERENCES

Idit Diamant, Amir Rosenfeld, Idan Achituve, Jacob Goldberger, and Arnon Netzer. De-confusing pseudo-labels in source-free domain adaptation. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2024.

Stephan Dreiseitl. A comparison of covariate shift detection methods on medical datasets. In *International Conference on Computer Aided Systems Theory*, pp. 487–495. Springer, 2022.

Lior Frenkel, Shlomo E. Chazan, and Jacob Goldberger. Detection of actionable domain shifts in speech enhancement systems by tracking prediction uncertainty. *Journal on Audio, Speech, and Music Processing*, 2026.

Saurabh Garg, Sivaraman Balakrishnan, Zachary C. Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations (ICLR)*, 2022.

Tony Ginart et al. MLDemon: Deployment monitoring for machine learning systems. In *International Conference on Artificial Intelligence and Statistics*, pp. 3962–3997, 2022.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

Hao Guan, David Bates, and Li Zhou. Keeping medical AI healthy and trustworthy: A review of detection and correction methods for system degradation. *IEEE Transactions on Biomedical Engineering*, 2025.

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*, 2017.

Sooyong Jang, Sangdon Park, Insup Lee, and Osbert Bastani. Sequential covariate shift detection using classifier two-sample tests. In *International Conference on Machine Learning (ICML)*, 2022.

Lisa M. Koch et al. Distribution shift detection for the postmarket surveillance of medical AI algorithms: a retrospective simulation study. *npj Digital Medicine*, 7:1–11, 2024.

Ali Kore et al. Empirical data drift detection experiments on real-world medical imaging data. *Nature Communications*, 15(1):1–11, 2024.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Zachary C. Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*, 2018.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jameson Merkow et al. Chexstray: a real-time multi-modal monitoring workflow for medical imaging AI. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 326–336. Springer, 2023.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020.

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2019.

Lorena Poenaru-Olaru et al. McUDI: Model-centric unsupervised degradation indicator for failure prediction AIOps solutions. *arXiv preprint arXiv:2401.14093*, 2024.

Stephan Rabanser, Stephan Günnemann, and Zachary C. Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Mélanie Roschewitz, Raghav Mehta, Charles Jones, and Ben Glocker. Automatic dataset shift identification to support safe deployment of medical imaging AI. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2025.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2010.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Renchunzi Xie, Ambroise Odonnat, Vasilii Feofanov, Weijian Deng, Jianfeng Zhang, and Bo An. MaNo: Exploiting matrix norm for unsupervised accuracy estimation under distribution shifts. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Wenyu Zhang, Li Shen, and Chuan-Sheng Foo. Rethinking the role of pre-trained networks in source-free domain adaptation. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2023.

A Appendix

A.1 Additional Controlled shift severity

Tables 5 and 6 show the results we got on domain A, P respectively, with the same settings used for the Controlled shift severity section in the main paper.

Table 5: Average p-value under varying Gaussian noise levels on the OfficeHome domain A.

Noise Level	0.015	0.02	0.03	0.04	0.05	0.075	0.1	0.125	0.15	0.175
Accuracy Degradation	1.1	1.1	1.3	2.1	3.1	7.1	11.2	16.4	22.1	27.1
KS features	88.2	12.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KS softmax	100.0	100.0	99.3	57.2	4.7	0.0	0.0	0.0	0.0	0.0
KS confidence	94.7	90.8	80.9	50.6	29.7	1.1	0.0	0.0	0.0	0.0
KS entropy	96.0	91.3	78.5	52.3	33.1	1.5	0.0	0.0	0.0	0.0
MMD features	100.0	100.0	93.1	50.0	49.7	37.3	30.1	32.3	33.0	32.9
MMD softmax	54.0	47.8	36.3	27.4	10.1	1.1	0.9	0.9	0.9	0.9
MMD confidence	89.3	89.0	97.0	97.0	100.0	100.0	100.0	100.0	100.0	18.2
MMD entropy	71.9	97.6	97.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0
C2ST features	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2ST softmax	100.0	100.0	100.0	99.9	99.9	99.9	93.7	4.2	0.0	0.0
C2ST confidence	53.8	42.6	31.6	21.6	9.8	2.0	0.0	0.0	0.0	0.0
C2ST entropy	78.8	79.4	58.4	24.6	9.2	1.2	0.0	0.0	0.0	0.0
PLSC	29.0	28.6	26.4	16.6	4.3	0.9	0.3	0.1	0.1	0.1

Table 6: Average p-value under varying Gaussian noise levels on the OfficeHome P domain

Noise Level	0.015	0.02	0.03	0.04	0.05	0.075	0.1	0.125	0.15	0.175
Accuracy Degradation	0.1	0.3	1.2	2.6	4.1	8.0	12.2	17.9	24.7	34.5
KS features	0.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KS softmax	97.0	34.1	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KS confidence	56.5	22.1	1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KS entropy	63.7	28.8	1.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MMD features	100.0	66.2	28.0	31.0	34.5	39.9	45.2	46.7	45.7	46.7
MMD softmax	24.6	11.2	1.4	1.0	0.9	0.9	0.9	0.9	0.9	0.9
MMD confidence	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	0.9
MMD entropy	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	1.4
C2ST features	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
C2ST softmax	100.0	100.0	100.0	100.0	99.9	99.3	2.4	0.0	0.0	0.0
C2ST confidence	26.7	12.3	1.9	0.1	0.0	0.0	0.0	0.0	0.0	0.0
C2ST entropy	31.2	12.9	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PLSC	62.5	60.1	57.8	29.0	15.5	0.4	0.1	0.1	0.1	0.1

Sample size and class imbalance sensitivity. We evaluated (1) the impact of sample size on detection accuracy, and (2) robustness to class imbalance. All experiments followed the controlled shift severity protocol on the R domain of OfficeHome.

For sample size, we randomly sampled K instances (without replacement) from both the source and target datasets and applied PLSC on these subsets (denoted PLSC K).

For class imbalance, we simulated a natural long-tail shift by exponentially under-sampling the data. Classes were sorted by decreasing frequency, and for each class index $i \in \{0, \ldots, C-1\}$, the number of samples was defined as:

$$N_i = \max(1, \lfloor N_{\text{max}} \cdot \mu^i \rfloor), \tag{4}$$

where N_{max} denoted the size of the largest class and $\mu = 0.953$ ensured a single-sample tail.

The results, presented in Table 7, illustrate how PLSC behaves under data scarcity and class imbalance.

Sample Size Robustness: As expected, statistical power depends heavily on sample size. We observe two key limitations when data is scarce. First, below $K = 300$, PLSC fails to consistently detect mild distribution shifts (e.g., at 9.1% accuracy degradation, PLSC 100 and 200 yield non-significant average p-values of 25.5 and 13.1, respectively). Second, for smaller subsets (typically below 600 examples), the strict monotonicity of the p-values breaks down in the low-noise regime, a slight increase in shift severity does not always result in a more extreme p-value. Subsets of $K \geq 600$ offer a much smoother and more reliable progression.

Class Imbalance Resilience: Under the simulated long-tail distribution (PLSC imbalanced), the method maintains its sensitivity. Following the exponential sampling, this imbalanced dataset retains exactly 405 examples. Notably, its detection power (e.g., p-value$= 2.0$ at 9.1% degradation) is comparable to, or even slightly stronger than, uniform subsets of similar sizes (such as PLSC 400 and PLSC 500, which yield p-value$= 9.8$ and p-value$= 3.0$ respectively). This confirms that PLSC's performance does not degrade under severe class imbalance; the probability-weighted prototypes ensure the test remains robust and primarily governed by the total number of available samples rather than the uniformity of the class distribution.

Table 7: Average p-value (%) of PLSC under varying sample sizes ($K = 100$ to 800) and simulated class imbalance (long-tail) across controlled noise levels on the OfficeHome R domain.

Noise Level	0.015	0.02	0.03	0.04	0.05	0.075	0.1	0.125	0.15	0.175
Accuracy Degradation	-0.1	0.5	1.6	2.5	3.9	9.1	15.2	23.9	35.6	47.3
PLSC imbalanced	61.1	83.7	41.8	14.7	12.6	2.0	0.1	0.1	0.1	0.1
PLSC 100	53.9	62.1	62.3	52.1	53.9	25.5	4.8	0.6	0.3	0.1
PLSC 200	53.5	46.4	51.8	41.5	32.7	13.1	1.6	0.1	0.1	0.1
PLSC 300	59.4	46.8	38.4	38.7	27.8	4.6	0.1	0.1	0.1	0.1
PLSC 400	56.3	52.7	45.2	48.9	29.1	9.8	0.2	0.1	0.1	0.1
PLSC 500	58.9	60.9	51.6	44.5	26.9	3.0	0.1	0.1	0.1	0.1
PLSC 600	73.3	70.7	66.8	57.5	30.7	6.0	0.1	0.1	0.1	0.1
PLSC 700	66.4	54.6	56.4	53.1	25.8	3.3	0.1	0.1	0.1	0.1
PLSC 800	45.7	34.4	31.5	34.2	19.6	0.5	0.1	0.1	0.1	0.1

B NEURAL-COLLAPSE INTERPRETATION.

The consistency between PL and PPL on the source domain can be interpreted through the geometry of neural collapse (NC) (Papyan et al., 2020). In particular, the neural-collapse property NC3 states that, near neural collapse, the weight vector of each class in the final linear classifier becomes aligned with the corresponding centered class-mean feature vector. Hence, the classifier geometry and the class-prototype geometry become approximately equivalent. The main difference between the class means used in NC3 and the prototypes used in PPL is that, in NC3, the class means are computed using the true labels, whereas in PPL, the class prototypes are estimated from unlabeled data using soft assignments. NC4 further states that, under the neural-collapse geometry, the linear classification rule approaches nearest-class-center classification. This suggests that classifier-based and prototype-based assignments should tend to agree on source-like data, provided that the soft PPL prototypes are good approximations to the underlying class centers. This is consistent with the high PL–PPL agreement observed on the source domain. We note that classical NC results describe behavior on the training set at zero training error. We are computing PL-PPL on a held-out unlabeled source reference set. (Papyan et al., 2020) also empirically observed the NC4 nearest-class-center behavior on held-out test examples, showing increasing agreement between the network classifier and a nearest-class-center rule based on training class means. Under domain shift, however, the classifier weights remain tied to the source-domain geometry, while the target feature clusters and their estimated prototypes may shift. A harmful domain shift can therefore break the alignment between the fixed classifier and the target-domain prototype structure, leading to increased PL–PPL disagreement. From this perspective, PLSC can be interpreted as measuring the breakdown of classifier–prototype alignment induced by domain shift. This connection is intended as a geometric interpretation rather than as a formal consequence of neural collapse, since the PPL prototypes are estimated from unlabeled soft assignments rather than from true class labels.