EVOLVING REPLAY HULLS FOR GEOMETRICALLY SUPPORTED CONTINUAL DOMAIN SHIFT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised continual domain shift learning (UCDSL) requires continual adaptation to sequentially arriving unlabeled domains while retaining previously acquired knowledge and generalizing to future domains. However, its reliance on self-training makes continual adaptation vulnerable to unreliable pseudo-labels, whose errors can accumulate across domain stages. We propose a reliable UCDSL framework that estimates pseudo-label confidence from class-wise replay geometry. Replay samples of each class define a convex hull in representation space, and target reliability is measured by relative distances to class-wise hulls. The framework maintains informative replay regions throughout continual adaptation, allowing this confidence criterion to remain effective as new domains are encountered. We theoretically show how the coverage of class-wise replay hulls and sample distances to them relate to classification risk on unseen domains. Empirically, the proposed framework improves adaptation, generalization, and forgetting alleviation across multiple benchmark datasets, while achieving better predictive calibration than existing UCDSL methods.

1 INTRODUCTION

Deep neural networks have achieved remarkable performance across a broad range of applications, largely under the assumption that training and test data are independent and identically distributed (Zhou et al., 2023). In real-world deployment, environmental and acquisition conditions may change over time, exposing a model to a sequence of previously unseen distributions (Truong et al., 2024; Chen et al., 2025). Unsupervised continual domain shift learning (UCDSL) considers this setting, where a model is initially trained on a labeled source domain and subsequently updated on continually arriving unlabeled target domains. At each domain stage, the model is initialized from the previous stage and further updated using only the current unlabeled target data and a small replay memory containing a limited subset of previously observed samples (Liu et al., 2023; Cho et al., 2023; Sun et al., 2025).

Since target annotations are unavailable, UCDSL relies on pseudo-label-based self-training to acquire knowledge from each incoming domain (Lee, 2013; Liang et al., 2020). This dependence makes pseudo-label reliability critical. Unlike conventional domain adaptation, an incorrect pseudo-label in UCDSL does not affect only the current target domain. It can distort the learned representation, influence the supervision generated for subsequent domains, and be repeatedly propagated through replay. Consequently, pseudo-label errors may accumulate and progressively destabilize continual adaptation (Wang et al., 2022).

Most pseudo-labeling approaches estimate reliability using classifier outputs, such as maximum softmax probability or predictive entropy. However, modern neural networks are often poorly calibrated and can assign high confidence to incorrect or out-of-distribution samples (Guo et al., 2017; Hendrycks & Gimpel, 2017; Kim & Lee, 2025). Moreover, classifier-based confidence generally becomes less reliable as the test distribution departs from the training distribution (Ovadia et al., 2019; Wang et al., 2023). We address this problem by assessing target-sample reliability in terms of class-wise coverage, which indicates whether a target representation is supported by class-conditional evidence retained from previous stages. The replay memory provides explicit sample-level evidence for this purpose. We represent the retained evidence of each class using the convex geometry spanned by its replay samples. Specifically, class-c replay samples $\{x_{c,i}^{\mathrm{rep}}\}_{i=1}^{m_c}$ define a class-wise

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.

convex hull

$$\mathcal{H}_c = \mathrm{conv}\{z(x_{c,i}^{\mathrm{rep}})\}_{i=1}^{m_c}, \quad \rho_c(x) = \min_{p \in \mathcal{H}_c} \|z(x) - p\|_2, \tag{1}$$

where $z(x)$ is the representation of a target sample x. The hull residual $\rho_c(x)$ measures whether the sample can be supported by convex combinations of retained class evidence. We convert these residuals into geometry-based confidence scores and use them to weight pseudo-label supervision, thereby emphasizing samples covered by the replay geometry while suppressing geometrically unsupported predictions. Figure 1 illustrates how steeply sloped softmax confidence contours can assign high confidence to shifted samples despite limited support from previously observed class regions, whereas replay-hull confidence decreases as such geometric support weakens.

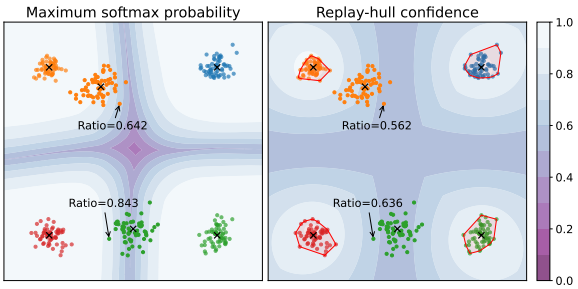


Figure 1: Confidence reliability under domain shift.

However, the effectiveness of this geometric confidence depends on how faithfully the replay memory characterizes the class-wise support region. Although a class prototype provides a compact representation, it reduces retained class variations to a single center and does not explicitly characterize their observed extent (Snell et al., 2017; Lu et al., 2024). As heterogeneous domains accumulate, class representations may occupy structured and extended regions that cannot be adequately described by their mean alone. To preserve such class-wise structure under a limited memory budget, we modify the iCaRL exemplar selection strategy (Rebuffi et al., 2017) to preserve both representative support exemplars and geometrically extensive vertex exemplars. Building on the existing mean-preserving approach, this design retains both the internal structure and geometric extent of each class. We further align the class-wise target geometry with the replay geometry through a hull-matching objective that matches their extent along multiple directions. Together, these components maintain a reliable class-wise support region that evolves continuously across domains.

In this work, we propose a reliable learning framework for UCDSL in which class-wise replay geometry provides confidence estimates for identifying trustworthy pseudo-labeled samples during continual adaptation. This geometric formulation not only enables more stable and effective learning under continual domain shifts, but also provides a basis for theoretically characterizing pseudo-label reliability in terms of unseen domain prediction risk. We empirically validate the proposed framework by evaluating both standard UCDSL performance and the reliability of the proposed confidence estimates using standard calibration metrics.

2 Related Works

Domain Adaptation and Domain Generalization Domain adaptation (DA) transfers knowledge from a labeled source domain to a different target domain, typically using unlabeled target samples to reduce distribution discrepancy (Sun & Saenko, 2016; Ganin et al., 2016; Long et al., 2018; Saito et al., 2018). Source-free domain adaptation further removes the need to retain source data by adapting a pretrained source model directly to the unlabeled target domain (Liang et al., 2020). In contrast, domain generalization (DG) aims to learn representations that generalize to unseen domains without target data during training (Zhou et al., 2023), using strategies such as meta-learning, domain diversification, and robust optimization (Li et al., 2018; Zhou et al., 2021; Cha et al., 2021). However, DA and DG are generally formulated for a fixed target domain or a fixed set of source domains. They therefore do not address sequentially arriving domains, which characterize continual learning scenarios.

Unsupervised Continual Domain Shift Learning UCDSL extends domain shift learning to a non-stationary sequence of unlabeled target domains, requiring models to adapt to the current domain while maintaining performance on both previously observed and unseen domains. RaTP (Liu et al., 2023) introduced this continual setting and proposed RandMix augmentation, Top^2 pseudo-labeling (T2PL), and prototype contrastive alignment as its main components. CoDAG (Cho et al.,

2023) subsequently identified the trade-off between DA and DG as a fundamental issue in this continual setting and addressed it by maintaining separate DA and DG models with complementary interaction. More recently, MPM (Sun et al., 2025) challenged the assumption of a single domain-invariant representation and accordingly allowed multiple prototypes to represent heterogeneous domain characteristics.

Although these methods improve continual adaptation and generalization, their pseudo-labeling mechanisms remain largely dependent on the model's own predictions. RaTP, CoDAG, and MPM adopt different prediction-based criteria for pseudo-labeling, but none explicitly addresses the overconfidence of classifiers under domain shift. As a result, incorrect predictions may be assigned high confidence (Guo et al., 2017; Ovadia et al., 2019), leading to unreliable supervision during continual adaptation. In contrast, our work evaluates pseudo-label reliability using class-conditional evidence accumulated from previously observed domains. We assess whether a target representation is geometrically supported by replay samples of its predicted class, providing an evidence-based reliability criterion that complements classifier confidence.

3 PRELIMINARIES

We adopt the unsupervised continual domain shift learning (UCDSL) setting introduced by RaTP (Liu et al., 2023), which considers a classification task over a sequence of domains encountered one at a time. Let $\mathcal{X} \subseteq \mathbb{R}^m$ denote the input space and $\mathcal{Y} = \{1, \dots, C\}$ the label space, where C is the number of classes. We consider an ordered sequence of $T+1$ domains indexed by $t \in \{0, \dots, T\}$. At the initial stage, the model is trained on the labeled source dataset $\mathcal{D}_0 = \{(x_i^0, y_i^0)\}_{i=1}^{N_0}$, where $x_i^0 \in \mathcal{X}$ and $y_i^0 \in \mathcal{Y}$. At each subsequent stage $t \in \{1, \dots, T\}$, a target dataset $\mathcal{D}_t = \{x_i^t\}_{i=1}^{N_t}$ arrives, where $x_i^t \in \mathcal{X}$, while the corresponding labels are unavailable during training. At target stage t, the model obtained from the preceding stage is updated using only the current target dataset $\mathcal{D}_t$ and a replay memory $\mathcal{M}_{t-1}$ containing samples retained from previously observed domains. The replay memory has a fixed capacity B, such that $|\mathcal{M}_t| \leq B$ for $t = 0, \dots, T$, while the complete datasets from preceding domains are no longer accessible. After learning from $\mathcal{D}_t$, the memory is updated to $\mathcal{M}_t$ and carried forward to the subsequent stage.

The UCDSL setting evaluates a model trained through stage t from three perspectives. Target domain adaptation (TDA) measures adaptation to the current domain $\mathcal{D}_t$. Target domain generalization (TDG) measures generalization to unseen future domains $\{\mathcal{D}_s\}_{s=t+1}^{T}$, while forgetting alleviation (FA) measures knowledge retention on previously observed domains $\{\mathcal{D}_s\}_{s=0}^{t-1}$. These three perspectives emphasize reliable learning throughout the domain sequence.

4 METHODOLOGY

We now introduce our proposed framework, referred to as **T**arget **R**eliability **A**ssessment via **C**onvex **E**vidence (**TRACE**). Figure 2 provides an overview of the proposed training procedure across the source and target stages. At each stage $t \in \{0, \dots, T\}$, we maintain a model composed of an encoder $f_{\theta_t} : \mathcal{X} \to \mathbb{R}^d$ and a classifier $g_\phi : \mathbb{R}^d \to \mathbb{R}^C$. At the source stage, f_{θ_0} and g_ϕ are jointly trained on the labeled source dataset $\mathcal{D}_0$ using supervised cross-entropy. The source-trained classifier is subsequently frozen, providing fixed class decision directions throughout continual adaptation. Accordingly, changes in model predictions across target stages are induced solely by the evolution of encoder representations relative to the fixed classifier.

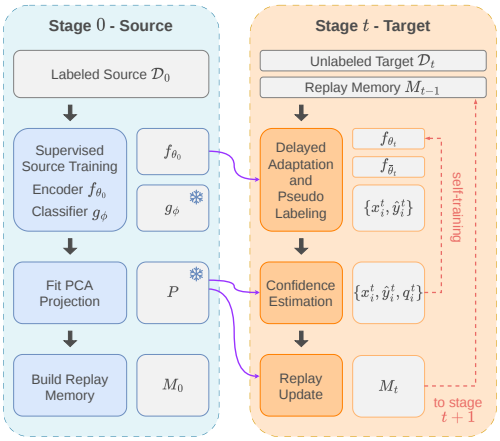


Figure 2: Overall training procedure.

After source training, we also fit a PCA projection $P : \mathbb{R}^d \to \mathbb{R}^r$ to the source representations, providing a lower-dimensional space that reduces

sparsity under the limited replay budget. We then construct the replay memory $\mathcal{M}_0$ by selecting representative source exemplars in this space. The projection P remains fixed throughout continual adaptation, consistent with the frozen classifier g_ϕ. With the source-stage components established, the following subsections detail the reliable adaptation procedure for each target stage.

4.1 Delayed Adaptation for Stable Pseudo-Labeling

Pseudo-labels provide supervision required to adapt the inherited encoder to $\mathcal{D}_t$. However, if the same rapidly updated encoder is used to both generate and learn from pseudo-labels, an initially incorrect label can bias the representation and repeatedly reinforce itself. This feedback loop can lead to confirmation bias. We therefore decouple these roles using an adaptation encoder f_{θ_t} and a reference encoder $f_{\bar{\theta}_t}$. At the beginning of stage t, both encoders are initialized from the reference encoder retained at stage $t-1$. During target adaptation, only the adaptation encoder f_{θ_t} is optimized via gradient descent, whereas the reference encoder $f_{\bar{\theta}_t}$ follows an exponential moving average update

$$\bar{\theta}_t^{(k)} = \alpha \bar{\theta}_t^{(k-1)} + (1-\alpha)\theta_t^{(k)}, \qquad \alpha \in [0, 1). \tag{2}$$

This delayed update suppresses parameter fluctuations and prevents transient changes in the adaptation encoder from immediately altering the reference geometry. Accordingly, the reference encoder gradually incorporates target domain information, favoring representation changes that remain consistent with knowledge accumulated from previous domains.

For input x, the feature representations $z_{\theta_t}(x)$ and $z_{\bar{\theta}_t}(x)$ yield the corresponding class probabilities $p_{\theta_t}(x)$ and $p_{\bar{\theta}_t}(x)$ as

$$\begin{aligned} z_{\theta_t}(x) &= f_{\theta_t}(x), \quad p_{\theta_t}(x) = \sigma(g_\phi(z_{\theta_t}(x))), \\ z_{\bar{\theta}_t}(x) &= f_{\bar{\theta}_t}(x), \quad p_{\bar{\theta}_t}(x) = \sigma(g_\phi(z_{\bar{\theta}_t}(x))), \end{aligned} \tag{3}$$

where σ denotes the softmax function. Within this reference geometry, we apply T2PL (Liu et al., 2023) to obtain pseudo-labels and estimate their reliability using the replay hull confidence.

4.2 Replay Hull Confidence Estimation

The reference encoder provides a stable basis for pseudo-label generation, but its predictive probabilities may remain overconfident under domain shift. We therefore estimate confidence based on whether a sample is geometrically supported by replay exemplars assigned to its pseudo-label class.

When training in stage t, let $A_{t-1,c}$ denote the class-c exemplars stored in the replay memory, where $\mathcal{M}_{t-1} = \bigcup_{c=1}^{C} A_{t-1,c}$. Using the current reference encoder and the fixed PCA projection P, we construct the class-wise replay hull and compute the residual of a target sample x_i^t as

$$\mathcal{H}_{t,c} = \mathrm{conv}\{P(z_{\bar{\theta}_t}(x)) \colon x \in A_{t-1,c}\}, \quad \rho_{t,c}(x_i^t) = \min_{p \in \mathcal{H}_{t,c}} \|P(z_{\bar{\theta}_t}(x_i^t)) - p\|_2. \tag{4}$$

A small residual indicates that the target representation is well supported by convex combinations of class-c replay exemplars. For a target sample x_i^t with pseudo-label $\hat{y}_i^t$, we define its replay hull confidence using the relative distances to the predicted-class hull and its nearest competing hull as

$$d_i^{\mathrm{pred}} = \rho_{t,\hat{y}_i^t}(x_i^t), \quad d_i^{\mathrm{comp}} = \min_{c \neq \hat{y}_i^t} \rho_{t,c}(x_i^t), \quad q_i^t = \frac{d_i^{\mathrm{comp}}}{d_i^{\mathrm{pred}} + d_i^{\mathrm{comp}}}. \tag{5}$$

Accordingly, $q_i^t > 0.5$ when the predicted-class hull is closer than any competing hull, and larger values reflect stronger separation from the competing hulls.

4.3 Objectives for Reliable Self-Training

We optimize the adaptation encoder on the combined dataset $\tilde{\mathcal{D}}_t = \mathcal{D}_t \cup \mathcal{M}_{t-1}$. Target samples are paired with the pseudo-labels and confidence scores computed at stage t, whereas replay samples retain the labels and confidence scores assigned when they were inserted into memory.

Confidence-weighted supervision To reduce the influence of unreliable pseudo-labels, each sample's contribution to the classification loss is scaled by its confidence, giving greater weight to geometrically supported samples. Let $w_t(x)$ denote the confidence associated with sample x. The confidence-weighted classification loss is

$$\mathcal{L}_{\mathrm{ce}} = \frac{\sum_{(x,y)\in\tilde{\mathcal{D}}_t} w_t(x)\,\mathrm{CE}(p_{\theta_t}(x), y)}{\sum_{(x,y)\in\tilde{\mathcal{D}}_t} w_t(x)}, \tag{6}$$

where $\mathrm{CE}(\cdot,\cdot)$ denotes the cross-entropy loss.

Knowledge distillation To mitigate forgetting during target adaptation, the encoder $f_{\bar{\theta}_{t-1}}$ from the previous stage is kept frozen and used as the distillation reference. Its predictive distribution is $p_{\bar{\theta}_{t-1}}(x) = \sigma(g_\phi(f_{\bar{\theta}_{t-1}}(x)))$. The resulting loss is

$$\mathcal{L}_{\mathrm{dist}} = \mathbb{E}_{x\in\tilde{\mathcal{D}}_t}[D_{\mathrm{KL}}(p_{\bar{\theta}_{t-1}}(x)\|p_{\theta_t}(x))], \tag{7}$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence.

Hull matching The hull-matching loss aligns the directional extent of the target representations with the support value of the corresponding replay hull along each sampled direction. For each class c, let $Z^{\mathrm{tgt}}_{t,c} = \{P(z_{\theta_t}(x_i^t))\colon \hat{y}_i^t = c\}$ and $Z^{\mathrm{rep}}_{t,c} = \{\mathrm{sg}(P(z_{\theta_t}(x)))\colon x \in A_{t-1,c}\}$. These sets contain the projected representations of class-c samples from the current target domain and replay memory, respectively. The stop-gradient in $Z^{\mathrm{rep}}_{t,c}$ allows the replay geometry to serve as a fixed anchor. For a set Z and direction v, we define the average of its k largest projections as

$$\mathrm{TopAvg}_k(Z; v) = \frac{1}{k}\sum_{z\in\mathrm{TopK}_k(Z;v)} z^\top v, \tag{8}$$

where $\mathrm{TopK}_k(Z; v)$ returns the k elements of Z having the largest projections onto v. The hull-matching loss is then

$$\mathcal{L}_{\mathrm{hull}} = \mathbb{E}_{c\in\mathcal{C}_t, v\in\Omega}[(\mathrm{TopAvg}_k(Z^{\mathrm{tgt}}_{t,c}; v) - \mathrm{TopAvg}_1(Z^{\mathrm{rep}}_{t,c}; v))^2]. \tag{9}$$

Here, $\Omega = \{\pm v_\ell\}_{\ell=1}^{N_{\mathrm{dir}}}$ is constructed from N_{dir} randomly sampled unit directions and their opposites, while $\mathcal{C}_t$ contains the classes with sufficient target samples accumulated across minibatches. This alignment encourages consistency between the current target representations and the geometric extent captured by class-wise replay samples from previously observed domains. This is intended to reduce forgetting while promoting representations that remain transferable to future domain shifts.

Repulsion To prevent the target representations from prematurely collapsing into a narrow region, we define a repulsion loss over all distinct pairs $i, j \in \{1, \ldots, n\}$ in a target minibatch as

$$\mathcal{L}_{\mathrm{rps}} = \frac{1}{n(n-1)}\sum_{i\neq j}\exp(z_{\theta_t}(x_i^t)^\top z_{\theta_t}(x_j^t)). \tag{10}$$

By penalizing high pairwise similarity without relying on pseudo-labels, this loss encourages representational diversity during early adaptation.

Taken together, the complete objective for target adaptation is

$$\mathcal{L} = \mathcal{L}_{\mathrm{ce}} + \mathcal{L}_{\mathrm{dist}} + \beta\mathcal{L}_{\mathrm{hull}} + \gamma\mathcal{L}_{\mathrm{rps}}. \tag{11}$$

The coefficients β and γ are loss weights, with γ annealed during training.

4.4 Geometry-Preserving Replay Selection

At the end of each stage, we select replay exemplars from the current domain using a modified iCaRL exemplar selection strategy (Rebuffi et al., 2017). The original strategy greedily constructs an exemplar set whose mean approximates the class mean, thereby favoring representative samples. To complement this mean-based criterion with geometric extent preservation, we distinguish vertex exemplars that capture the outer geometry from support exemplars that retain the interior structure of the class distribution.

Let $u_i = P(z_{\bar{\theta}_t}(x_i^t))$ denote a candidate representation and μ_c the mean of the candidates assigned to class c. Given the selected index set I, candidate i is scored by

$$J_\lambda(i|I) = \lambda\|u_i - \mu_c\|_2 - (1-\lambda)\|\mu_{I\cup\{i\}} - \mu_c\|_2, \qquad \lambda \in [0,1]. \tag{12}$$

Here, $\mu_{I\cup\{i\}}$ is the mean after adding candidate i. The second term follows the iCaRL criterion by favoring exemplar sets whose mean remains close to the class mean, whereas the first favors samples farther from the class center.

For each class, we first select vertex exemplars using λ_{v} and then select support exemplars from the remaining candidates using $\lambda_{\mathrm{s}} < \lambda_{\mathrm{v}}$. The selected exemplars are stored in the replay memory $\mathcal{M}_t$ to support knowledge retention during subsequent adaptation.

4.5 Theoretical Justification

The following analysis examines the class-wise replay geometry at a fixed continual-learning stage $t \geq 1$. Let $z(x) = z_{\bar{\theta}_t}(x) \in \mathbb{R}^d$ denote the common representation used for target samples and replay exemplars, including both stored vertices and support samples.

Definition 1 (Class-wise replay hull and residual)**.** For each class $c \in \{1,\dots,C\}$, assume $A_{t-1,c} \neq \varnothing$. We define the class-wise replay hull and the residual of an input x as

$$\mathcal{H}_c \coloneqq \mathrm{conv}\{z(x)\colon x \in A_{t-1,c}\}, \quad \rho_c(x) \coloneqq \min_{p\in\mathcal{H}_c} \|z(x) - p\|_2. \tag{13}$$

The residual $\rho_c(x)$ measures a class-conditional coverage gap. It is zero when the sample lies inside the region spanned by the class-c replay exemplars and increases as the sample moves away from all of their convex combinations. Unlike a distance to a class mean or a single prototype, this residual measures proximity to the set of convex combinations of the retained class-c representations.

Definition 2 (Classifier and pairwise logit margins)**.** Let $g_j\colon \mathbb{R}^d \to \mathbb{R}$ denote the j-th logit of the fixed classifier g_ϕ. The classifier prediction and pairwise logit margin are defined as

$$h(z) \coloneqq \operatorname*{arg\,max}_{j\in\{1,\dots,C\}} g_j(z), \quad M_{c,k}(z) \coloneqq g_c(z) - g_k(z), \quad c \neq k. \tag{14}$$

Assumption 1 (Label-consistent replay hulls and regular logit margins)**.** For each class c, suppose there exist $\kappa_c > 0$ and $L_{c,k} > 0$ for all $k \neq c$ such that

$$|M_{c,k}(z) - M_{c,k}(z')| \leq L_{c,k}\|z - z'\|_2, \qquad \forall z, z' \in \mathbb{R}^d, \tag{15}$$

and

$$M_{c,k}(p) \geq \kappa_c, \qquad \forall p \in \mathcal{H}_c. \tag{16}$$

We denote $L_c \coloneqq \max_{k\neq c} L_{c,k}$.

Assumption 1 requires the class-c replay hull to retain a pairwise logit margin of at least κ_c, while L_c controls how rapidly this margin may change away from the hull. Under our fixed linear head $g_j(z) = \omega_j^\top z$, where $\omega_j \in \mathbb{R}^d$ is the class-j weight vector, both conditions become verifiable. Equation (15) holds with $L_{c,k} = \|\omega_c - \omega_k\|_2$. Since $M_{c,k}$ is affine, equation (16) reduces to a finite check at the anchors regardless of hull size.

Lemma 1 (Residual-based classification certificate)**.** *Under Assumption 1, for any sample x with true class c,*

$$\rho_c(x) < \frac{\kappa_c}{L_c} \Rightarrow h(z(x)) = c. \tag{17}$$

Lemma 1 shows that the ratio κ_c/L_c defines a certified neighborhood around the class-c replay hull. Accordingly, any misclassified sample of class c must lie outside this certified neighborhood.

The following analysis extends the sample-wise certificate to classification risk on an arbitrary unseen domain. Let $\mathbb{P}$ denote the joint distribution over (X, Y) at the current continual-learning stage, with $\mathbb{P}_c$ denoting the conditional distribution of X given $Y = c$. For an arbitrary unseen domain distribution $\mathbb{Q}$ over (X, Y), define the class prior $\pi_c \coloneqq \mathbb{Q}(Y = c)$ and the classification risk as

$$R_{\mathbb{Q}}(h) \coloneqq \mathbb{Q}(h(z(X)) \neq Y). \tag{18}$$

We use $\mathbb{P}_c^z$ and $\mathbb{Q}_c^z$ to denote the class-conditional distributions of $z(X)$ induced by $\mathbb{P}$ and $\mathbb{Q}$, respectively. For the analysis, consider i.i.d. samples $x_1^{(c)}, \dots, x_{n_c}^{(c)} \sim \mathbb{P}_c$ for each class c, drawn independently of model training and replay selection. The empirical mean residual is then defined as $\hat{\rho}_c \coloneqq \frac{1}{n_c}\sum_{i=1}^{n_c} \rho_c(x_i^{(c)})$.

Theorem 1 (Class-wise hull coverage controls unseen domain error)**.** *Suppose Assumption 1 holds and $\|z(x)\|_2 \leq 1$ for all inputs under consideration. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$ over the i.i.d. samples used to compute $\hat{\rho}_c$, we have*

$$R_{\mathbb{Q}}(h) \leq \sum_{c=1}^{C} \pi_c \min\left\{1, \frac{L_c}{\kappa_c}\left(\hat{\rho}_c + W_1(\mathbb{P}_c^z, \mathbb{Q}_c^z) + \sqrt{\frac{2\log(C/\delta)}{n_c}}\right)\right\}. \tag{19}$$

Here, W_1 denotes the 1-Wasserstein distance.

Theorem 1 extends the sample-wise certificate of Lemma 1 to an unseen domain risk bound. The empirical residual $\hat{\rho}_c$ measures the coverage achieved by the current replay geometry, while $W_1(\mathbb{P}_c^z, \mathbb{Q}_c^z)$ accounts for class-conditional representation shift between the current and unseen domains. The final term captures finite-sample uncertainty. The bound becomes tighter with better replay-hull coverage, smaller class-conditional representation shifts, or a larger and more stable classifier margin around the replay hull.

This result motivates enlarging the replay hull as new class-conditional variations are observed. Consider an enlarged hull $\mathcal{H}_c^+ \supseteq \mathcal{H}_c$ that satisfies the label-consistency condition in Assumption 1 with the same margin κ_c. Define

$$\rho_c^+(x) \coloneqq \min_{p \in \mathcal{H}_c^+} \|z(x) - p\|_2, \quad \hat{\rho}_c^+ \coloneqq \frac{1}{n_c}\sum_{i=1}^{n_c} \rho_c^+(x_i^{(c)}). \tag{20}$$

Corollary 1 (Replay hull enlargement)**.** *Under the assumptions of Theorem 1, suppose each replay hull $\mathcal{H}_c$ is enlarged to $\mathcal{H}_c^+ \supseteq \mathcal{H}_c$ while preserving the same classifier margin κ_c. Then $\rho_c^+(x) \leq \rho_c(x)$ for all x, and hence*

$$\hat{\rho}_c^+ \leq \hat{\rho}_c. \tag{21}$$

Consequently, the certified region in (17) *can only expand, and the right-hand side of* (19) *cannot increase.*

These theoretical results motivate the following design choices in TRACE. First, to assess pseudo-label reliability, we use distances to class-wise replay hulls for confidence estimation, grounding each score in the geometric support provided by replay exemplars. Second, we admit only high-confidence target samples into replay memory to promote label-consistent hull enlargement and limit the influence of unreliable labels. Third, we L_2-normalize encoder representations and use a linear classifier without a bias term, which is frozen after source training. The normalization bounds the representation norm, while the frozen linear classifier keeps the logit-margin directions and Lipschitz constants fixed. Finally, we use a fixed PCA projection fitted to source representations to construct compact replay hulls under a limited memory budget. This choice is motivated by the fixed linear classifier, which uses at most $C-1$ independent feature directions to distinguish C classes. By retaining the main source variation, PCA provides a practical approximation of replay geometry in fewer dimensions.

5 Experiments

5.1 Experimental Setup

We compare TRACE with three UCDSL baselines, RaTP (Liu et al., 2023), CoDAG (Cho et al., 2023), and MPM (Sun et al., 2025), following the experimental protocol established by RaTP. In each domain sequence, the first domain serves as the labeled source domain, while the remaining domains are sequentially encountered as unlabeled target domains. We evaluate all methods over the same 10 domain orders using three random seeds, 2022, 2023, and 2024. Further details on the datasets, domain orders, training configurations, and evaluation metrics are provided in Appendix B.

Datasets We conduct experiments on three benchmark datasets, Digits-5 (Ganin & Lempitsky, 2015; Hull, 1994; Lecun et al., 1998; Netzer et al., 2011), PACS (Li et al., 2017), and DomainNet (Peng et al., 2019). Considering label noise and class imbalance, we construct a subset of DomainNet for our experiments (Xie et al., 2022).

Table 1: Performance across datasets.

Dataset	Metric	RaTP	CoDAG	MPM	TRACE
Digits-5	TDA	87.0	87.4	72.7	**89.0**
	TDG	76.1	**77.4**	66.3	77.0
	FA	83.7	84.1	49.0	**84.4**
	All	82.2	83.0	62.7	**83.5**
PACS	TDA	80.1	85.6	82.2	**87.3**
	TDG	66.1	64.1	67.2	**70.0**
	FA	81.5	86.8	82.5	**88.0**
	All	75.9	78.8	77.3	**81.8**
DomainNet	TDA	53.0	**68.7**	57.1	68.2
	TDG	50.5	53.4	47.4	**56.3**
	FA	53.4	**67.3**	55.0	67.1
	All	52.3	63.1	53.2	**63.9**

Table 2: Calibration across datasets.

Dataset	Metric	RaTP	CoDAG	MPM	TRACE
Digits-5	ECE	0.1292	0.1240	0.1442	**0.0992**
	Brier	0.1387	0.1473	0.1447	**0.1305**
	NLL	0.8937	0.4787	1.2522	**0.4120**
	AURC	0.0911	0.0835	**0.0743**	0.0790
PACS	ECE	0.1329	0.2067	0.2188	**0.1105**
	Brier	0.1543	0.2157	0.2188	**0.1420**
	NLL	0.7379	0.6935	2.9757	**0.4250**
	AURC	0.1136	0.1637	0.1100	**0.1080**
DomainNet	ECE	0.2715	0.3349	0.3895	**0.1501**
	Brier	0.2637	0.3295	0.3883	**0.1975**
	NLL	1.0574	1.3628	5.1709	**0.5674**
	AURC	0.3303	0.3359	**0.2648**	0.2913

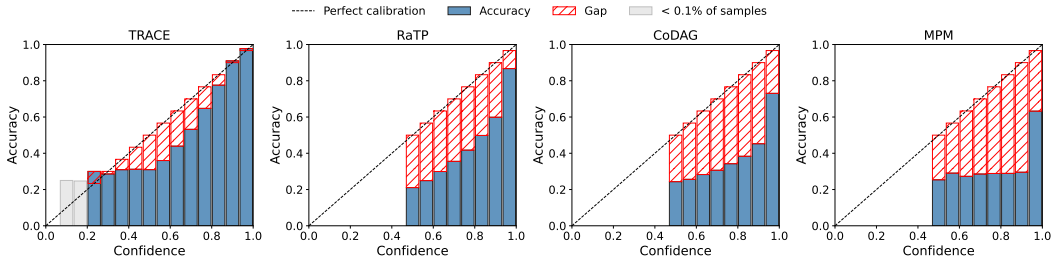


Figure 3: Reliability diagrams on DomainNet, averaged over all 10 domain orders. The diagonal indicates perfect calibration, and the hatched regions show the gaps between confidence and accuracy.

Implementation Details For the source domain, 80% of the data is used for training and the remaining 20% for evaluation. For each target domain, all samples are used for adaptation, while their labels are used only for evaluation. We use DTN as the backbone for Digits-5 (Liang et al., 2020) and a ResNet-50 pretrained on ImageNet for PACS and DomainNet. All models are optimized using SGD with a batch size of 64, and the replay buffer size is fixed to 200 for all datasets.

Evaluation Metrics We report classification accuracy in terms of TDA, TDG, and FA, along with their mean as an overall score. To assess predictive reliability, we use Expected Calibration Error (ECE) (Guo et al., 2017) to evaluate whether prediction confidence reflects empirical accuracy. The Brier score (Brier, 1950) and negative log-likelihood (NLL) complement this assessment by evaluating the quality of predicted class probabilities. We further assess selection ability using the area under the risk-coverage curve (AURC) (Geifman et al., 2019), which summarizes error rates among retained predictions across coverage levels. Lower is better for all four reliability metrics. In all tables, the best and second-best results for each metric are indicated in **bold** and underline, respectively.

5.2 Results

Classification Performance Table 1 summarizes classification accuracy averaged over the 10 domain orders, with the overall score (All) computed as the mean of TDA, TDG, and FA. TRACE achieves the highest overall score on all three datasets, with distinct improvements across the evaluation objectives. When averaging model ranks across all domain orders, TRACE consistently ranks first for every evaluation metric. Notably, TRACE shows stronger TDG performance on the more challenging datasets, particularly PACS and DomainNet. This suggests that accounting for pseudo-label reliability helps prevent excessive model drift under larger domain shifts, thereby preserving more stable generalization to unseen domains. Together, these results indicate that TRACE maintains a favorable balance between adaptation, generalization, and forgetting alleviation throughout continual domain learning. Results for individual domain orders are provided in Appendix C.1.

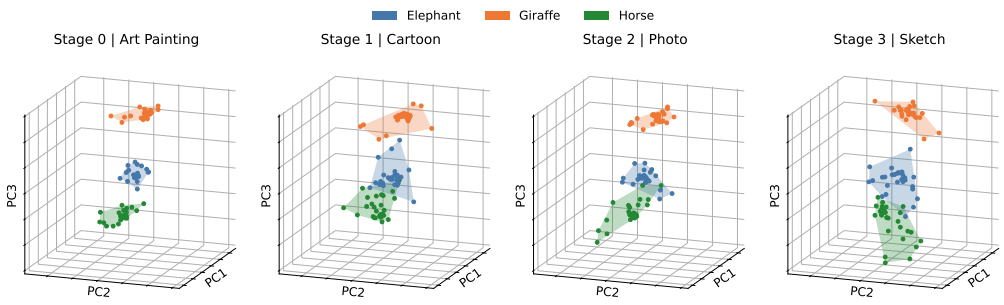


Figure 4: Evolution of class-wise replay hulls under continual domain shifts.

Predictive Reliability Table 2 compares predictive reliability using ECE, Brier score, NLL, and AURC. TRACE achieves the lowest ECE, Brier score, and NLL across all three datasets, indicating consistent improvements in confidence calibration and probabilistic prediction quality. These gains are particularly pronounced on DomainNet, where ECE and NLL decrease from the best baseline values of 0.2715 and 1.0574 to 0.1501 and 0.5674, respectively. The reliability diagrams in Figure 3 provide complementary visual evidence. TRACE exhibits closer agreement between confidence and accuracy, whereas the baselines show larger calibration gaps associated with overconfident predictions. For selection ability, TRACE achieves the lowest AURC on PACS and ranks second on Digits-5 and DomainNet. Together, these results demonstrate that TRACE combines improved classification accuracy with better confidence calibration and competitive selection ability under continual domain shifts. Predictive reliability results for individual domain orders are provided in Appendix C.2.

Expansion of Reliable Regions Figure 4 visualizes the replay geometry after each stage for three selected classes in PACS under domain order 1. Replay exemplars are encoded by the reference encoder saved at each stage and displayed along the first three principal components of the fixed PCA projection. All panels use identical axis limits and viewing angles. The source-stage hulls are relatively compact, while subsequent stages exhibit class-specific changes in their extent and orientation. These evolving geometries illustrate how the hulls represent variation among retained exemplars beyond what a single class center can convey. By using relative distances to these hulls, TRACE assesses target reliability based on the spatial extent of past class evidence rather than class-center proximity alone. Replay memory thus serves not only to retain past examples, but also to provide stage-specific geometric references that guide confidence-weighted adaptation under sequential domain shifts.

Overall, these results suggest that TRACE uses replay geometry to support reliable adaptation while limiting model drift, leading to strong classification performance and well-calibrated confidence under sequential domain shifts. Additional experimental results, including an ablation study and a sensitivity analysis, are provided in Appendix C.

6 Conclusion

In this work, we presented TRACE, a UCDSL framework that treats replay memory not only as a mechanism for retaining past knowledge, but also as geometric evidence for assessing pseudo-label reliability under continual domain shifts. TRACE estimates confidence from relative distances to class-wise replay hulls, providing a geometric criterion for identifying reliable pseudo-labels. To keep this criterion informative, TRACE maintains representative replay regions with sufficient geometric extent as domains continue to arrive. Our theoretical analysis connects replay-hull coverage and sample residuals to unseen-domain classification risk, and shows how label-consistent hull enlargement expands the supported region. Empirically, TRACE achieves strong performance across benchmarks while consistently improving predictive calibration. These findings highlight the importance of explicitly accounting for pseudo-label reliability under sequential domain shifts.

AI USE STATEMENT

We used generative AI tools to polish the writing, improve readability, and check grammar, spelling, and typographical errors. We also used these tools to assist with debugging and completing implementation code. All AI-assisted content was reviewed by the authors. Generated or modified code was verified and tested for correctness by two authors. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with AI assistance.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the source code for TRACE in the supplementary material. Section 5.1 and Appendix B describe the datasets, domain orders, training configurations, and evaluation metrics used in our experiments. Complete proofs of the theoretical results presented in the main text are provided in Appendix A.

REFERENCES

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. In *Advances in Neural Information Processing Systems*, volume 34, pp. 22405–22418. Curran Associates, Inc., 2021.

Bojian Chen, Xinmin Zhang, Changqing Shen, Qi Li, and Zhihuan Song. Couda: Continual unsupervised domain adaptation for industrial fault diagnosis under dynamic working conditions. *IEEE Transactions on Industrial Informatics*, 21(5):4072–4082, 2025. doi: 10.1109/TII.2025.3538135.

Wonguk Cho, Jinha Park, and Taesup Kim. Complementary domain adaptation and generalization for unsupervised continual domain shift learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11408–11418, 2023. doi: 10.1109/ICCV51070.2023.01051.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. In *International Conference on Learning Representations*, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017.

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

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

J.J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. doi: 10.1109/34.291440.

Taewook Kim and Jong-Seok Lee. Gently sloped and extended classification margin for overconfidence relaxation of out-of-distribution samples. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):12854–12865, 2025. doi: 10.1109/TNNLS.2024.3496473.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, pp. 896. Atlanta, 2013.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Deeper, broader and artier domain generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5543–5551, 2017. doi: 10.1109/ICCV.2017.591.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6028–6039. PMLR, 13–18 Jul 2020.

Chenxi Liu, Lixu Wang, Lingjuan Lyu, Chen Sun, Xiao Wang, and Qi Zhu. Deja vu: Continual model generalization for unseen domains. In *The Eleventh International Conference on Learning Representations*, 2023.

Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Haodong Lu, Dong Gong, Shuo Wang, Jason Xue, Lina Yao, and Kristen Moore. Learning with mixture of prototypes for out-of-distribution detection. In *The Twelfth International Conference on Learning Representations*, 2024.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, pp. 4. Granada, 2011.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua 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*, volume 32. Curran Associates, Inc., 2019.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1406–1415, 2019. doi: 10.1109/ICCV.2019.00149.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017. doi: 10.1109/CVPR.2017.587.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, 2018. doi: 10.1109/CVPR.2018.00392.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision – ECCV 2016 Workshops*, pp. 443–450, Cham, 2016. Springer International Publishing. ISBN 978-3-319-49409-8.

Haopeng Sun, Yingwei Zhang, Lumin Xu, Sheng Jin, Ping Luo, Chen Qian, Wentao Liu, and Yiqiang Chen. Unsupervised continual domain shift learning with multi-prototype modeling. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10131–10141, 2025. doi: 10.1109/CVPR52734.2025.00947.

Thanh-Dat Truong, Pierce Helton, Ahmed Moustafa, Jackson David Cothren, and Khoa Luu. Conda: Continual unsupervised domain adaptation learning in visual perception for self-driving cars. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 5642–5650, 2024. doi: 10.1109/CVPRW63382.2024.00573.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, 2008. ISBN 978-3-540-71049-3. doi: 10.1007/978-3-540-71050-9.

Haoxuan Wang, Zhiding Yu, Yisong Yue, Animashree Anandkumar, Anqi Liu, and Junchi Yan. Learning calibrated uncertainties for domain shift: a distributionally robust learning approach. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI '23, 2023. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/162.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7191–7201, 2022. doi: 10.1109/CVPR52688.2022.00706.

Jiangwei Xie, Shipeng Yan, and Xuming He. General incremental learning with domain-aware categorical representations. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14331–14340, 2022. doi: 10.1109/CVPR52688.2022.01395.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *International Conference on Learning Representations*, 2021.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2023. doi: 10.1109/TPAMI.2022.3195549.

A Detailed Proofs

A.1 Proof of Lemma 1

Proof. Fix a sample x with true class c. Since $\mathcal{H}_c$ is the convex hull of finitely many points, it is compact. Hence, there exists

$$p_c^\star \coloneqq \arg\min_{p\in\mathcal{H}_c} \|z(x) - p\|_2, \tag{22}$$

such that

$$\|z(x) - p_c^\star\|_2 = \rho_c(x). \tag{23}$$

For any competing class $k \neq c$, Assumption 1 gives

$$M_{c,k}(z(x)) \geq M_{c,k}(p_c^\star) - L_{c,k}\|z(x) - p_c^\star\|_2 \tag{24}$$

$$\geq \kappa_c - L_{c,k}\rho_c(x) \tag{25}$$

$$\geq \kappa_c - L_c\rho_c(x), \tag{26}$$

where the last inequality uses $L_{c,k} \leq L_c$ and $\rho_c(x) \geq 0$. Therefore, if $\rho_c(x) < \kappa_c/L_c$, then

$$M_{c,k}(z(x)) > 0, \qquad \forall k \neq c. \tag{27}$$

Equivalently, $g_c(z(x)) > g_k(z(x))$ for every $k \neq c$, which implies $h(z(x)) = c$. □

A.2 Proof of Theorem 1

Proof. We first convert the sample-wise certificate into a population risk bound and then estimate the resulting class-conditional residuals using the current-stage samples.

Step 1. From the sample-wise certificate to unseen domain risk By Lemma 1, every classification error on a class-c sample satisfies

$$\{h(z(X)) \neq c\} \subseteq \Big\{\rho_c(X) \geq \frac{\kappa_c}{L_c}\Big\} \qquad \text{on the event } \{Y = c\}. \tag{28}$$

Conditioning on the true class therefore gives

$$R_{\mathbb{Q}}(h) = \sum_{c=1}^{C} \pi_c \mathbb{Q}(h(z(X)) \neq c | Y = c) \tag{29}$$

$$\leq \sum_{c=1}^{C} \pi_c \mathbb{Q}\Big(\rho_c(X) \geq \frac{\kappa_c}{L_c}\Big| Y = c\Big). \tag{30}$$

Since $\rho_c(X) \geq 0$, Markov's inequality yields

$$\mathbb{Q}\Big(\rho_c(X) \geq \frac{\kappa_c}{L_c}\Big| Y = c\Big) \leq \min\Big\{1, \frac{L_c}{\kappa_c}\mathbb{E}_{\mathbb{Q}}[\rho_c(X)\Big| Y = c]\Big\}. \tag{31}$$

Combining (30) and (31),

$$R_{\mathbb{Q}}(h) \leq \sum_{c=1}^{C} \pi_c \min\Big\{1, \frac{L_c}{\kappa_c}\mathbb{E}_{\mathbb{Q}}[\rho_c(X)|Y = c]\Big\}. \tag{32}$$

Step 2. Transfer from the current stage to the unseen domain For a fixed class c and $z' \in \mathbb{R}^d$, define the nearest point in the replay hull and the corresponding distance as

$$p_{z'} \coloneqq \arg\min_{p\in\mathcal{H}_c} \|z' - p\|_2, \quad r_c(z') \coloneqq \|z' - p_{z'}\|_2. \tag{33}$$

The function r_c is 1-Lipschitz. Indeed, for arbitrary $z, z' \in \mathbb{R}^d$,

$$r_c(z) \leq \|z - p_{z'}\|_2 \tag{34}$$

$$\leq \|z - z'\|_2 + \|z' - p_{z'}\|_2 \tag{35}$$

$$= \|z - z'\|_2 + r_c(z'). \tag{36}$$

Exchanging z and z' gives

$$|r_c(z) - r_c(z')| \leq \|z - z'\|_2. \tag{37}$$

Since $\rho_c(X) = r_c(z(X))$, the Kantorovich-Rubinstein duality (Villani, 2008) implies

$$\mathbb{E}_{\mathbb{Q}}[\rho_c(X)|Y = c] \leq \mathbb{E}_{\mathbb{P}}[\rho_c(X)|Y = c] + W_1(\mathbb{P}_c^z, \mathbb{Q}_c^z). \tag{38}$$

Step 3. Finite-sample estimation By the norm bound in Theorem 1, every point $p \in \mathcal{H}_c$ satisfies $\|p\|_2 \le 1$. Hence, for any such p,

$$0 \le \rho_c(x) \le \|z(x) - p\|_2 \le \|z(x)\|_2 + \|p\|_2 \le 2. \tag{39}$$

Since the evaluation samples are independent of model training and replay selection, their residuals $\rho_c(x_i^{(c)})$ are i.i.d. and bounded in $[0, 2]$ conditional on the trained encoder and replay anchors. Applying Hoeffding's inequality (Hoeffding, 1963) then yields, with probability at least $1 - \delta/C$,

$$\mathbb{E}_{\mathbb{P}}[\rho_c(X)|Y = c] \le \hat{\rho}_c + \sqrt{\frac{2\log(C/\delta)}{n_c}}. \tag{40}$$

A union bound makes (40) hold simultaneously for all classes with probability at least $1 - \delta$. On this event, combining (38) and (40) gives

$$\mathbb{E}_{\mathbb{Q}}[\rho_c(X)|Y = c] \le \hat{\rho}_c + W_1(\mathbb{P}_c^z, \mathbb{Q}_c^z) + \sqrt{\frac{2\log(C/\delta)}{n_c}}, \quad \forall c. \tag{41}$$

Finally, since $s \mapsto \min\{1, (L_c/\kappa_c)s\}$ is nondecreasing on $[0, \infty)$, substituting (41) into (32) proves (19). □

A.3 Proof of Corollary 1

Proof. Since $\mathcal{H}_c \subseteq \mathcal{H}_c^+$, every point feasible in the minimization defining $\rho_c(x)$ is also feasible in the minimization defining $\rho_c^+(x)$. Therefore,

$$\rho_c^+(x) = \min_{p \in \mathcal{H}_c^+} \|z(x) - p\|_2 \tag{42}$$

$$\le \min_{p \in \mathcal{H}_c} \|z(x) - p\|_2 = \rho_c(x). \tag{43}$$

Averaging this inequality over the samples from $\mathbb{P}_c$ gives

$$\hat{\rho}_c^+ \le \hat{\rho}_c, \tag{44}$$

which proves (21).

Because the enlarged hull satisfies the same margin condition with κ_c, Lemma 1 also applies to ρ_c^+. Moreover,

$$\left\{x \colon \rho_c(x) < \frac{\kappa_c}{L_c}\right\} \subseteq \left\{x \colon \rho_c^+(x) < \frac{\kappa_c}{L_c}\right\}, \tag{45}$$

so the sample-wise certified region can only expand.

Finally, the Wasserstein term and the sampling term in (19) do not depend on the replay hull. Hence (44) implies, for every class c,

$$\hat{\rho}_c^+ + W_1(\mathbb{P}_c^z, \mathbb{Q}_c^z) + \sqrt{\frac{2\log(C/\delta)}{n_c}} \le \hat{\rho}_c + W_1(\mathbb{P}_c^z, \mathbb{Q}_c^z) + \sqrt{\frac{2\log(C/\delta)}{n_c}}. \tag{46}$$

Since $s \mapsto \min\{1, (L_c/\kappa_c)s\}$ is nondecreasing, replacing $\mathcal{H}_c$ by $\mathcal{H}_c^+$ cannot increase the right-hand side of (19). □

B Additional Experimental Details

This section provides additional details of the experimental setup, including datasets and domain orders, implementation settings, hyperparameter configurations, and the computational environment.

B.1 Datasets and Domain Orders

Digits-5 consists of five domains with 10 digit classes, including MNIST (MT) (Lecun et al., 1998), MNIST-M (MM) (Ganin & Lempitsky, 2015), SVHN (SN) (Netzer et al., 2011), Synthetic Digits (SD) (Ganin & Lempitsky, 2015), and USPS (US) (Hull, 1994). PACS consists of four domains with

Table 3: Domain orders used for the main experiments.

Order	Digits-5	PACS	DomainNet
Order 1	SN→MT→MM→SD→US	A→C→P→S	Re→Pa→In→Cl→Sk→Qu
Order 2	SN→SD→MT→US→MM	A→C→S→P	Cl→In→Pa→Qu→Re→Sk
Order 3	MM→US→MT→SD→SN	A→P→C→S	Cl→Re→In→Qu→Sk→Pa
Order 4	MT→MM→SN→SD→US	C→A→S→P	In→Qu→Cl→Pa→Re→Sk
Order 5	MT→MM→US→SN→SD	C→S→P→A	Pa→Sk→Qu→In→Re→Cl
Order 6	SD→MM→SN→MT→US	P→A→C→S	Qu→Re→Cl→Pa→In→Sk
Order 7	SD→SN→US→MM→MT	P→S→A→C	Qu→Sk→Cl→In→Pa→Re
Order 8	SD→US→MM→SN→MT	P→S→C→A	Sk→In→Pa→Cl→Re→Qu
Order 9	US→MT→SN→MM→SD	S→C→A→P	Sk→Re→Pa→Cl→Qu→In
Order 10	US→SD→SN→MM→MT	S→P→C→A	Sk→Re→Qu→Pa→In→Cl

seven shared classes, including Photo (P), Art Painting (A), Cartoon (C), and Sketch (S) (Li et al., 2017). DomainNet contains six domains, including Quickdraw (Qu), Clipart (Cl), Painting (Pa), Infograph (In), Sketch (Sk), and Real (Re) (Peng et al., 2019). Considering label noise and class imbalance, we use a 10-class subset of DomainNet containing the classes with the largest number of samples across all domains (Xie et al., 2022). We conduct experiments using 10 domain orders for each dataset, where the first domain serves as the labeled source domain and the remaining domains are sequentially encountered as unlabeled target domains. The domain orders used in our experiments are provided in Table 3.

B.2 Implementation Details

Table 4: Dataset-specific training configurations for TRACE.

Hyperparameter	Digits-5	PACS	DomainNet
Backbone	DTN	ResNet-50	ResNet-50
Encoder dimensions	[256, 128]	[512, 256]	[512, 256]
Steps per epoch	800	50	70
Maximum training epochs	40	50	30
Initial learning rate	0.01	0.005	0.005
Confidence threshold τ	0.7	0.8	0.8
PCA dimension r	10	7	10
Hull-matching weight β	1.0	1.0	0.5
Repulsion weight γ	0.2	0.2	0.5
r_{top} (Liu et al., 2023)	1	2	2
r'_{top} (Liu et al., 2023)	10	20	20

Dataset-specific model architectures and training configurations are summarized in Table 4. Unless otherwise specified, the settings described below are shared across all datasets. All experiments are implemented in PyTorch and conducted using an NVIDIA H200 GPU.

Optimization We use SGD with momentum 0.9, weight decay 5×10^{-4}, and a batch size of 64. A polynomial learning-rate scheduler is used with dataset-specific initial learning rates.

Data Augmentation For PACS and DomainNet, we follow the augmentation protocol used by CoDAG and MPM (Cho et al., 2023; Sun et al., 2025), including random resized cropping, horizontal flipping, color jittering, random grayscaling, RandMix (Liu et al., 2023), and Mixup (Zhang et al., 2018). Mixup is applied with probability 0.2, using a Beta distribution with both shape parameters set to 0.2. RandMix and Mixup are applied only to target samples whose confidence exceeds the threshold. Digits-5 is normalized without additional image augmentation.

Target Adaptation The reference encoder is updated using EMA with coefficient $\alpha = 0.99$, and pseudo-labels are refreshed every epoch. Target adaptation is terminated early when the predictions of the reference and adaptation encoders agree on at least 99% of target samples for three consecutive pseudo-label refreshes.

Replay Memory The replay memory has a fixed capacity of 200 samples, equally allocated between vertex and support exemplars. Target samples are considered as candidates for memory update only when their replay-hull confidence exceeds the dataset-specific confidence threshold. The vertex and support selection weights are set to $\lambda_{\mathrm{v}} = 0.8$ and $\lambda_{\mathrm{s}} = 0.0$, respectively, and replay batches contain 64 samples.

Geometry and Hull Matching Convex-hull distances are approximated using 30 Frank–Wolfe iterations. For hull matching, we sample $N_{\mathrm{dir}} = 64$ unit directions and use both their positive and negative directions. For each class c, target samples assigned to that class are accumulated in a class-specific queue across minibatches. Hull matching is evaluated once the queue contains at least 20 samples, after which the corresponding queue is cleared. For a class with n_c^{tgt} accumulated target samples, the number of target samples used to estimate each directional extent is

$$k_c = \max\{1, \mathrm{round}(0.1\, n_c^{\mathrm{tgt}})\}. \tag{47}$$

B.3 Evaluation Metrics

We consider a sequence of $T+1$ domains $\mathcal{D}_0, \ldots, \mathcal{D}_T$, where $\mathcal{D}_0$ is the source domain and $\mathcal{D}_1, \ldots, \mathcal{D}_T$ are sequentially encountered target domains. Reliability metrics are evaluated separately for each target stage, specifically when the initial pseudo-labels used to train the final model for that stage are generated. For RaTP, we evaluate the pseudo-labels obtained from the k-nearest-neighbor procedure in T2PL. For CoDAG, we evaluate the initial pseudo-labels used to train the DA model. For MPM, we evaluate the pseudo-labels generated by the domain model after it has been trained using SHOT and before these pseudo-labels are used to train the MPM model. In each case, the evaluation uses the corresponding pseudo-labels and their associated confidence scores. We first average each reliability metric over the three random seeds within each stage, then equally across target stages within each domain order, and finally across all 10 domain orders. In all tables, the best and second-best results for each metric are indicated in **bold** and underline, respectively.

Classification accuracy For a fixed domain order and random seed, let $\mathcal{E}_s$ denote the evaluation set for domain s. Let h_t^v denote the prediction rule obtained after completing stage t, where $v \in \{\mathrm{DA}, \mathrm{DG}\}$ identifies the model used for adaptation or generalization evaluation. The accuracy on domain s after stage t is

$$A_{t,s}^v = \frac{1}{|\mathcal{E}_s|} \sum_{(x,y)\in\mathcal{E}_s} \mathbf{1}\{h_t^v(x) = y\}. \tag{48}$$

CoDAG uses its DA model for TDA and its DG model for TDG and FA. For the other methods, the same prediction rule is used for both accuracy matrices.

The three classification metrics and their overall score are computed as

$$\mathrm{TDA} = \frac{1}{T+1} \sum_{s=0}^{T} A_{s,s}^{\mathrm{DA}}, \tag{49}$$

$$\mathrm{TDG} = \frac{1}{T} \sum_{s=1}^{T} \left(\frac{1}{s} \sum_{t=0}^{s-1} A_{t,s}^{\mathrm{DG}} \right), \tag{50}$$

$$\mathrm{FA} = \frac{1}{T} \sum_{s=0}^{T-1} \left(\frac{1}{T-s} \sum_{t=s+1}^{T} A_{t,s}^{\mathrm{DG}} \right), \tag{51}$$

$$\mathrm{All} = \frac{\mathrm{TDA} + \mathrm{TDG} + \mathrm{FA}}{3}. \tag{52}$$

TDA averages the diagonal entries, including the source stage. TDG first averages the accuracy on each target domain over all stages before that domain is encountered, and then averages across target domains. FA first averages the accuracy on each previously encountered domain over all subsequent stages, and then averages across domains with at least one subsequent stage.

For each domain order, we first average the accuracy matrices over the three random seeds and then compute the metrics above. The reported results are obtained by averaging the resulting metrics over all 10 domain orders.

Expected calibration error For each target stage t with N_t samples, let $a_i^t = \mathbf{1}\{\hat{y}_i^t = y_i^t\}$ denote the correctness of the pseudo-label assigned to x_i^t, where y_i^t is the ground-truth label used only for evaluation. We divide $[0, 1]$ into $K = 15$ bins of equal width and group samples according to their confidence scores q_i^t. For bin b, define

$$\mathcal{I}_b^t = \Big\{ i \in \{1, \dots, N_t\} \Big| \frac{b-1}{K} < q_i^t \le \frac{b}{K} \Big\}, \quad b = 1, \dots, K, \tag{53}$$

with the first bin also including $q_i^t = 0$. For each nonempty bin, its empirical accuracy and mean confidence are

$$\mathrm{acc}_b^t = \frac{1}{|\mathcal{I}_b^t|} \sum_{i \in \mathcal{I}_b^t} a_i^t, \quad \mathrm{conf}_b^t = \frac{1}{|\mathcal{I}_b^t|} \sum_{i \in \mathcal{I}_b^t} q_i^t. \tag{54}$$

The ECE at stage t is computed as

$$\mathrm{ECE}_t = \sum_{b=1}^{K} \frac{|\mathcal{I}_b^t|}{N_t} |\mathrm{acc}_b^t - \mathrm{conf}_b^t|, \tag{55}$$

where the contribution of each empty bin is defined to be zero.

Brier score We evaluate confidence as a probability estimate of pseudo-label correctness. The Brier score is computed against the binary correctness indicator as

$$\mathrm{Brier}_t = \frac{1}{N_t} \sum_{i=1}^{N_t} (q_i^t - a_i^t)^2. \tag{56}$$

Negative log-likelihood We likewise compute NLL for the binary event that a pseudo-label is correct. For numerical stability, confidence scores are clipped as

$$\tilde{q}_i^t = \min\{1 - \epsilon_{\mathrm{nll}}, \max\{\epsilon_{\mathrm{nll}}, q_i^t\}\}, \quad \epsilon_{\mathrm{nll}} = 10^{-7}. \tag{57}$$

Using natural logarithms, the NLL is

$$\mathrm{NLL}_t = -\frac{1}{N_t} \sum_{i=1}^{N_t} [a_i^t \log \tilde{q}_i^t + (1 - a_i^t) \log(1 - \tilde{q}_i^t)]. \tag{58}$$

Thus, both Brier score and NLL assess confidence estimates against pseudo-label correctness rather than evaluating the full probability distribution over classes.

Area under the risk-coverage curve To evaluate selective prediction, we sort samples by decreasing confidence so that $q_{(1)}^t \ge \cdots \ge q_{(N_t)}^t$. Let $a_{(j)}^t$ denote the correctness indicator of the sample at rank j. Retaining the first k predictions gives coverage and risk

$$\mathrm{coverage}_{t,k} = \frac{k}{N_t}, \quad R_{t,k} = \frac{1}{k} \sum_{j=1}^{k} (1 - a_{(j)}^t). \tag{59}$$

The implementation computes AURC as the arithmetic mean of these risks over all retained sample counts,

$$\mathrm{AURC}_t = \frac{1}{N_t} \sum_{k=1}^{N_t} R_{t,k}. \tag{60}$$

C Additional Experimental Results

This section provides detailed experimental results that complement the aggregate comparisons reported in the main text. Section C.1 reports classification performance for each of the 10 domain orders on Digits-5, PACS, and DomainNet, including TDA, TDG, FA, and their overall average. Section C.2 presents the corresponding order-wise predictive reliability results in terms of ECE, Brier score, NLL, and AURC. These detailed results provide a more complete view of the behavior of each method across different domain-shift trajectories beyond the averages reported in the main text. Finally, Section C.3 presents an ablation study that incrementally introduces the main components of TRACE to examine their individual contributions.

C.1 CLASSIFICATION PERFORMANCE

Table 5: Performance comparison on Digits-5.

Metric	Orders	RaTP	CoDAG	MPM	TRACE	Metric	Orders	RaTP	CoDAG	MPM	TRACE
TDA	Order 1	87.9	**94.6**	90.6	91.1	TDG	Order 1	76.6	**80.4**	74.1	78.7
	Order 2	91.3	**94.5**	63.4	93.4		Order 2	80.8	**82.8**	68.0	81.8
	Order 3	85.3	86.8	82.8	**88.8**		Order 3	75.8	77.1	67.8	**78.2**
	Order 4	**84.2**	78.9	73.8	82.1		Order 4	**71.3**	71.0	62.2	71.2
	Order 5	83.3	78.0	69.7	**87.6**		Order 5	71.3	71.6	61.5	**73.3**
	Order 6	90.0	89.7	77.1	**90.7**		Order 6	81.5	**82.7**	71.1	82.0
	Order 7	88.9	90.0	62.2	**91.1**		Order 7	80.3	**81.7**	58.2	80.6
	Order 8	89.4	89.2	73.4	**90.6**		Order 8	80.7	**82.3**	63.1	81.0
	Order 9	84.1	84.2	66.2	**87.7**		Order 9	71.9	**72.2**	67.3	72.0
	Order 10	85.7	**88.1**	67.9	87.1		Order 10	70.6	**72.5**	69.5	71.1
	Avg.	87.0	87.4	72.7	**89.0**		Avg.	76.1	**77.4**	66.3	77.0
FA	Order 1	82.1	**86.0**	70.7	83.6	ALL	Order 1	82.2	**87.0**	78.5	84.4
	Order 2	87.2	**90.0**	31.3	87.8		Order 2	86.4	**89.1**	54.2	87.7
	Order 3	88.4	**90.1**	72.0	89.8		Order 3	83.2	84.7	74.2	**85.6**
	Order 4	**80.3**	76.5	47.6	77.4		Order 4	**78.6**	75.5	61.2	76.9
	Order 5	81.9	80.2	58.9	**85.5**		Order 5	78.9	76.6	63.4	**82.1**
	Order 6	84.4	**85.0**	50.0	83.7		Order 6	85.3	**85.8**	66.1	85.5
	Order 7	84.0	**85.9**	39.5	85.6		Order 7	84.4	**85.9**	53.3	85.8
	Order 8	83.5	**83.9**	43.8	82.3		Order 8	84.5	**85.1**	60.1	84.7
	Order 9	83.2	82.6	47.1	**85.4**		Order 9	79.7	79.7	60.2	**81.7**
	Order 10	81.5	81.1	29.4	**83.2**		Order 10	79.3	**80.6**	55.6	80.5
	Avg.	83.7	84.1	49.0	**84.4**		Avg.	82.2	83.0	62.7	**83.5**

Table 6: Performance comparison on PACS.

Metric	Orders	RaTP	CoDAG	MPM	TRACE	Metric	Orders	RaTP	CoDAG	MPM	TRACE
TDA	Order 1	82.6	85.9	86.7	**88.1**	TDG	Order 1	73.4	69.8	**80.2**	77.4
	Order 2	81.0	85.4	85.4	**88.6**		Order 2	72.4	68.9	**80.3**	77.4
	Order 3	82.3	86.3	84.7	**88.7**		Order 3	71.9	67.8	74.0	**77.5**
	Order 4	84.6	88.4	83.5	**92.0**		Order 4	80.8	80.0	82.5	**85.6**
	Order 5	81.0	88.3	84.6	**90.4**		Order 5	77.2	78.6	74.9	**82.6**
	Order 6	77.4	81.4	78.6	**84.6**		Order 6	54.7	49.8	51.5	**55.5**
	Order 7	68.8	**80.2**	79.8	79.9		Order 7	48.1	49.9	**56.0**	53.2
	Order 8	68.7	**80.5**	78.9	78.9		Order 8	46.5	49.1	**56.0**	52.5
	Order 9	87.1	90.0	85.5	**91.0**		Order 9	69.7	65.5	65.1	**71.7**
	Order 10	87.7	89.7	74.0	**91.2**		Order 10	66.4	61.1	51.9	**66.7**
	Avg.	80.1	85.6	82.2	**87.3**		Avg.	66.1	64.1	67.2	**70.0**
FA	Order 1	87.4	88.7	85.1	**90.8**	ALL	Order 1	81.1	81.5	84.0	**85.4**
	Order 2	77.1	84.2	80.8	**85.4**		Order 2	76.8	79.5	82.1	**83.8**
	Order 3	86.6	89.2	83.9	**91.4**		Order 3	80.3	81.1	80.9	**85.9**
	Order 4	81.4	88.2	82.4	**89.9**		Order 4	82.3	85.5	82.8	**89.2**
	Order 5	81.5	90.2	85.4	**91.4**		Order 5	79.9	85.7	81.6	**88.1**
	Order 6	83.6	83.3	81.8	**87.4**		Order 6	71.9	71.5	70.6	**75.8**
	Order 7	74.7	**86.4**	83.6	83.9		Order 7	63.9	72.1	**73.1**	72.3
	Order 8	68.5	**83.0**	80.4	79.9		Order 8	61.2	70.9	**71.8**	70.5
	Order 9	85.5	88.6	83.5	**88.6**		Order 9	80.8	81.3	78.0	**83.8**
	Order 10	88.8	86.7	78.3	**91.6**		Order 10	81.0	79.2	68.0	**83.2**
	Avg.	81.5	86.8	82.5	**88.0**		Avg.	75.9	78.8	77.3	**81.8**

Table 7: Performance comparison on DomainNet.

Metric	Orders	RaTP	CoDAG	MPM	TRACE	Metric	Orders	RaTP	CoDAG	MPM	TRACE
TDA	Order 1	47.8	69.8	58.6	**70.5**	TDG	Order 1	44.7	50.9	46.9	**53.2**
	Order 2	58.8	73.6	61.8	**74.1**		Order 2	54.6	56.0	49.3	**61.7**
	Order 3	57.3	**74.1**	59.3	72.8		Order 3	54.4	58.0	47.2	**62.2**
	Order 4	47.7	58.7	56.0	**64.9**		Order 4	50.5	52.0	52.6	**58.5**
	Order 5	53.8	**72.9**	58.6	72.1		Order 5	51.2	54.0	49.8	**56.5**
	Order 6	43.6	**59.4**	51.0	51.3		Order 6	32.8	36.9	35.4	**37.4**
	Order 7	42.8	**53.9**	51.4	52.4		Order 7	34.3	37.6	**37.8**	37.7
	Order 8	58.6	**75.3**	59.5	74.6		Order 8	59.8	63.4	52.1	**65.4**
	Order 9	58.9	75.7	60.3	**76.1**		Order 9	62.3	63.6	52.4	**66.1**
	Order 10	61.3	**73.6**	54.7	73.2		Order 10	60.6	62.0	50.8	**64.4**
	Avg.	53.0	**68.7**	57.1	68.2		Avg.	50.5	53.4	47.4	**56.3**
FA	Order 1	51.5	69.7	57.7	**70.2**	ALL	Order 1	48.0	63.5	54.4	**64.6**
	Order 2	59.1	70.0	60.2	**72.9**		Order 2	57.5	66.5	57.1	**69.6**
	Order 3	55.8	69.6	54.7	**70.2**		Order 3	55.8	67.2	53.7	**68.4**
	Order 4	45.5	59.5	55.7	**64.1**		Order 4	47.9	56.7	54.8	**62.5**
	Order 5	52.4	67.7	52.6	**68.6**		Order 5	52.5	64.9	53.7	**65.7**
	Order 6	44.5	**59.8**	51.4	52.4		Order 6	40.3	**52.0**	45.9	47.0
	Order 7	42.3	**52.9**	48.6	49.2		Order 7	39.8	**48.1**	45.9	46.4
	Order 8	57.4	73.1	55.9	**73.4**		Order 8	58.6	70.6	55.8	**71.1**
	Order 9	65.3	**81.2**	63.8	80.8		Order 9	62.1	73.5	58.8	**74.4**
	Order 10	60.4	69.4	49.7	**69.4**		Order 10	60.8	68.3	51.7	**69.0**
	Avg.	53.4	**67.3**	55.0	67.1		Avg.	52.3	63.1	53.2	**63.9**

C.2 PREDICTIVE RELIABILITY

Table 8: Calibration comparison on Digits-5.

Metric	Orders	RaTP	CoDAG	MPM	TRACE	Metric	Orders	RaTP	CoDAG	MPM	TRACE
ECE	Order 1	0.1229	0.0873	**0.0815**	0.1132	Brier	Order 1	0.1337	0.1209	**0.0824**	0.1297
	Order 2	0.0851	**0.0557**	0.0572	0.1373		Order 2	0.0923	0.0863	**0.0582**	0.1060
	Order 3	0.1321	0.1281	0.1446	**0.0918**		Order 3	0.1385	0.1473	0.1446	**0.1185**
	Order 4	0.1550	0.1973	0.2437	**0.0717**		Order 4	0.1620	0.2090	0.2433	**0.1501**
	Order 5	0.1557	0.1890	0.2026	**0.0956**		Order 5	0.1668	0.2018	0.2023	**0.1321**
	Order 6	0.1097	**0.0888**	0.1158	0.0999		Order 6	0.1195	0.1167	**0.1164**	0.1204
	Order 7	0.1060	**0.0866**	0.1483	0.1141		Order 7	0.1161	0.1142	0.1492	**0.1108**
	Order 8	0.1096	**0.0910**	0.1187	0.1035		Order 8	0.1201	0.1228	**0.1193**	0.1276
	Order 9	0.1553	0.1634	0.1846	**0.0818**		Order 9	0.1652	0.1795	0.1849	**0.1549**
	Order 10	0.1609	0.1525	0.1453	**0.0834**		Order 10	0.1726	0.1748	**0.1464**	0.1548
	Avg.	0.1292	0.1240	0.1442	**0.0992**		Avg.	0.1387	0.1473	0.1447	**0.1305**
NLL	Order 1	1.0381	**0.3922**	0.6584	0.4131	AURC	Order 1	0.0841	0.0519	**0.0261**	0.0595
	Order 2	0.7149	**0.2769**	0.4534	0.3590		Order 2	0.0534	0.0299	**0.0155**	0.0350
	Order 3	0.9474	0.4887	1.1764	**0.3800**		Order 3	0.0996	0.0924	**0.0742**	0.0771
	Order 4	1.0643	0.7059	2.4154	**0.4553**		Order 4	**0.1171**	0.1444	0.1761	0.1218
	Order 5	0.9886	0.6572	1.8137	**0.4106**		Order 5	0.1238	0.1389	0.1265	**0.0876**
	Order 6	0.7463	**0.3730**	0.9573	0.3885		Order 6	0.0714	0.0552	**0.0518**	0.0646
	Order 7	0.8212	0.3779	1.3878	**0.3658**		Order 7	0.0709	0.0520	0.0675	**0.0490**
	Order 8	0.7412	**0.3874**	0.9564	0.4069		Order 8	0.0716	0.0594	**0.0526**	0.0694
	Order 9	0.8724	0.5618	1.5373	**0.4692**		Order 9	0.1064	0.1111	**0.0967**	0.1152
	Order 10	1.0022	0.5663	1.1656	**0.4714**		Order 10	0.1122	0.0997	**0.0564**	0.1111
	Avg.	0.8937	0.4787	1.2522	**0.4120**		Avg.	0.0911	0.0835	**0.0743**	0.0790

Table 9: Calibration comparison on PACS.

Metric	Orders	RaTP	CoDAG	MPM	TRACE	Metric	Orders	RaTP	CoDAG	MPM	TRACE
ECE	Order 1	**0.1057**	0.1846	0.1937	0.1299	Brier	Order 1	**0.1274**	0.1887	0.1937	0.1352
	Order 2	**0.1107**	0.1926	0.2029	0.1358		Order 2	**0.1340**	0.1975	0.2029	0.1418
	Order 3	**0.1050**	0.2070	0.1867	0.1200		Order 3	**0.1229**	0.2031	0.1868	0.1290
	Order 4	**0.0891**	0.1050	0.1852	0.1444		Order 4	**0.1133**	0.1349	0.1853	0.1280
	Order 5	0.1245	**0.1033**	0.1758	0.1211		Order 5	0.1433	0.1393	0.1758	**0.1252**
	Order 6	0.1569	0.3138	0.2722	**0.0950**		Order 6	0.1795	0.3081	0.2723	**0.1713**
	Order 7	0.2357	0.2996	0.3093	**0.0901**		Order 7	0.2498	0.2946	0.3092	**0.1942**
	Order 8	0.2240	0.3113	0.3111	**0.0944**		Order 8	0.2407	0.3026	0.3111	**0.1969**
	Order 9	0.1002	0.1453	0.1604	**0.0744**		Order 9	0.1202	0.1662	0.1602	**0.0928**
	Order 10	**0.0770**	0.2043	0.1904	0.0998		Order 10	0.1119	0.2226	0.1904	**0.1061**
	Avg.	0.1329	0.2067	0.2188	**0.1105**		Avg.	0.1543	0.2157	0.2188	**0.1420**
NLL	Order 1	0.6210	0.6015	2.5602	**0.4048**	AURC	Order 1	0.0828	0.1299	0.1077	**0.0824**
	Order 2	0.6592	0.6340	2.7120	**0.4245**		Order 2	**0.0867**	0.1359	0.1039	0.0924
	Order 3	0.6012	0.6343	2.4363	**0.3832**		Order 3	**0.0791**	0.1481	0.0947	0.0879
	Order 4	0.5933	0.4346	2.4591	**0.3862**		Order 4	**0.0666**	0.0736	0.1020	0.0895
	Order 5	0.8926	0.4784	2.3343	**0.3837**		Order 5	0.1047	**0.0800**	0.0807	0.0843
	Order 6	0.7598	0.9999	3.7679	**0.4971**		Order 6	**0.1216**	0.2529	0.1465	0.1327
	Order 7	1.0659	0.8963	4.3020	**0.5586**		Order 7	0.2367	0.2605	**0.1573**	0.2144
	Order 8	0.9480	0.9263	4.2515	**0.5663**		Order 8	0.2359	0.2782	**0.1578**	0.2236
	Order 9	0.7197	0.5803	2.2403	**0.3077**		Order 9	0.0696	0.1159	0.0565	**0.0398**
	Order 10	0.5182	0.7490	2.6933	**0.3380**		Order 10	0.0519	0.1621	0.0931	**0.0328**
	Avg.	0.7379	0.6935	2.9757	**0.4250**		Avg.	0.1136	0.1637	0.1100	**0.1080**

Table 10: Calibration comparison on DomainNet.

Metric	Orders	RaTP	CoDAG	MPM	TRACE	Metric	Orders	RaTP	CoDAG	MPM	TRACE
ECE	Order 1	0.3370	0.3685	0.4522	**0.1373**	Brier	Order 1	0.3052	0.3541	0.4509	**0.1969**
	Order 2	0.2339	0.3141	0.3367	**0.1270**		Order 2	0.2413	0.3102	0.3359	**0.1830**
	Order 3	0.2344	0.3011	0.3257	**0.1318**		Order 3	0.2308	0.2980	0.3247	**0.1905**
	Order 4	0.3074	0.3166	0.3674	**0.1270**		Order 4	0.2925	0.3060	0.3662	**0.1763**
	Order 5	0.2539	0.3444	0.4014	**0.1429**		Order 5	0.2422	0.3311	0.3999	**0.1867**
	Order 6	0.3617	0.4423	0.4666	**0.2326**		Order 6	0.3388	0.4339	0.4652	**0.2826**
	Order 7	0.3685	0.4525	0.5195	**0.2381**		Order 7	0.3503	0.4393	0.5175	**0.2688**
	Order 8	0.2099	0.2696	0.3336	**0.1104**		Order 8	0.2188	0.2695	0.3331	**0.1584**
	Order 9	0.1957	0.2545	0.3485	**0.1244**		Order 9	0.1967	0.2618	0.3478	**0.1557**
	Order 10	0.2123	0.2854	0.3432	**0.1292**		Order 10	0.2202	0.2912	0.3420	**0.1762**
	Avg.	0.2715	0.3349	0.3895	**0.1501**		Avg.	0.2637	0.3295	0.3883	**0.1975**
NLL	Order 1	1.4107	1.6008	6.0384	**0.5626**	AURC	Order 1	0.4206	0.3365	0.3380	**0.2641**
	Order 2	0.7608	1.1662	4.4178	**0.5298**		Order 2	0.2395	0.3034	**0.2053**	0.2444
	Order 3	0.7118	1.1627	4.1707	**0.5511**		Order 3	0.2771	0.3089	**0.1973**	0.2589
	Order 4	1.3068	1.0839	4.9848	**0.5172**		Order 4	0.3707	0.3336	**0.2502**	0.2785
	Order 5	0.8529	1.1928	5.2895	**0.5379**		Order 5	0.3333	0.3573	**0.2771**	0.2771
	Order 6	1.5981	1.9825	6.3133	**0.7864**		Order 6	0.4697	0.4478	**0.3261**	0.4532
	Order 7	1.7400	2.0394	6.8512	**0.7425**		Order 7	0.5108	0.4660	**0.3947**	0.4298
	Order 8	0.7400	1.0975	4.5055	**0.4700**		Order 8	0.2194	0.2653	**0.2041**	0.2220
	Order 9	0.6424	1.0364	4.6625	**0.4608**		Order 9	**0.2143**	0.2534	0.2343	0.2255
	Order 10	0.8107	1.2658	4.4748	**0.5159**		Order 10	0.2479	0.2863	**0.2211**	0.2599
	Avg.	1.0574	1.3628	5.1709	**0.5674**		Avg.	0.3303	0.3359	**0.2648**	0.2913

C.3 Ablation Study

Table 11: Incremental ablation of TRACE components.

Configuration	All	TDA	TDG	FA
Baseline	80.68	82.97	76.53	82.55
+ Delayed adaptation	81.14 (+0.46)	83.38 (+0.41)	76.70 (+0.18)	83.33 (+0.78)
+ Replay hull confidence	82.96 (+1.82)	87.92 (+4.54)	76.87 (+0.17)	84.09 (+0.76)
+ Vertex/support replay	83.04 (+0.08)	87.92 (+0.00)	76.89 (+0.02)	84.29 (+0.21)
+ Hull matching	83.34 (+0.30)	88.82 (+0.90)	76.95 (+0.06)	84.24 (-0.06)
+ Repulsion (full TRACE)	83.44 (+0.10)	88.89 (+0.07)	76.98 (+0.03)	84.45 (+0.21)

Table 12: Order-wise results of the incremental ablation study.

Order	Base	+DA	+Conf	+Vert	+Hull	+Rep
1	80.48	81.13	85.04	83.37	84.25	84.43
2	83.98	84.56	87.77	87.35	87.74	87.77
3	82.76	83.01	85.30	85.32	85.65	85.47
4	77.10	76.86	77.68	77.18	76.42	76.78
5	79.46	79.44	81.68	81.36	81.92	82.31
6	83.97	84.59	85.37	85.66	85.49	85.48
7	83.61	84.52	84.38	85.79	85.54	85.80
8	82.49	83.39	84.11	84.39	84.58	84.54
9	77.25	77.61	78.75	80.22	81.39	81.43
10	75.72	76.27	79.53	79.72	80.39	80.38
Mean	80.68	81.14	82.96	83.04	83.34	83.44

We conduct an incremental ablation study on Digits-5 using all 10 domain orders and three random seeds. Starting from a self-training baseline with an iCaRL replay buffer, we sequentially add delayed adaptation (DA), replay hull confidence (Conf), vertex/support replay selection (Vert), hull matching (Hull), and repulsion (Rep). In Table 11, values in parentheses denote the improvement over the preceding configuration. Table 12 reports the corresponding order-wise overall scores.

All components provide positive average gains in the overall score, improving it from 80.68 for the baseline to 83.44 for the full TRACE model. The largest gain comes from replay hull confidence, which improves the overall score by 1.82 points and TDA by 4.54 points. This result directly supports the main motivation of TRACE. By weighting pseudo-label supervision according to class-wise replay support, the model is less affected by geometrically unsupported predictions during adaptation, leading to substantially better performance on the current target domain. Delayed adaptation improves the overall score by 0.46 points and yields a notable gain in FA of 0.78 points, consistent with the role of the EMA reference encoder in reducing rapid model drift. Vertex/support replay selection, hull matching, and repulsion provide additional complementary gains. Overall, the ablation results show that replay-hull confidence is the main contributor to improved target adaptation, while delayed adaptation and the replay-geometry components further support knowledge retention and stable continual learning.

C.4 Sensitivity Analysis

Sensitivity to the main hyperparameters is evaluated on Digits-5, with results averaged over all domain orders. This analysis uses a single random seed. Specifically, β and γ control the contributions of the hull-matching and repulsion losses, respectively. The confidence threshold τ determines which target samples are treated as sufficiently reliable during adaptation. Figure 5 reports TDA, TDG, FA, and their overall average for different combinations of these parameters.

The results suggest that performance is comparatively more sensitive to τ than to β or γ. Across all three evaluation objectives, $\tau = 0.8$ consistently yields higher performance than $\tau = 0.5$, highlighting the importance of selecting sufficiently reliable samples based on confidence. This observation is

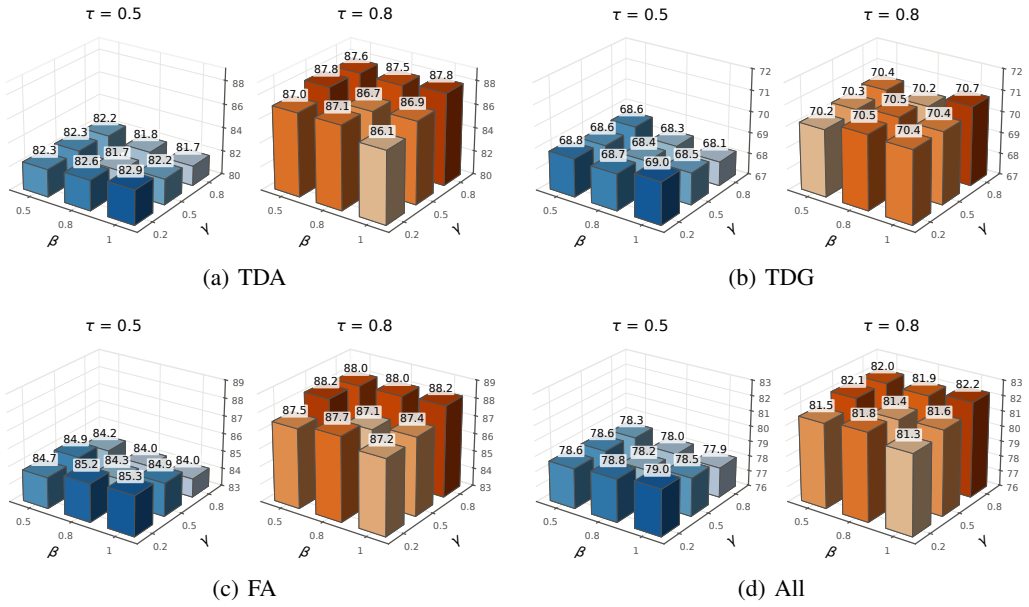


Figure 5: Sensitivity analysis of the hull-matching weight β, repulsion weight γ, and confidence threshold τ for (a) TDA, (b) TDG, (c) FA, and (d) overall performance.

consistent with our central motivation that confidence reliability is critical for stable continual adaptation. In contrast, varying β and γ within the evaluated ranges produces relatively small changes under a fixed τ.

C.5 Replay-Hull Residual and Prediction Error

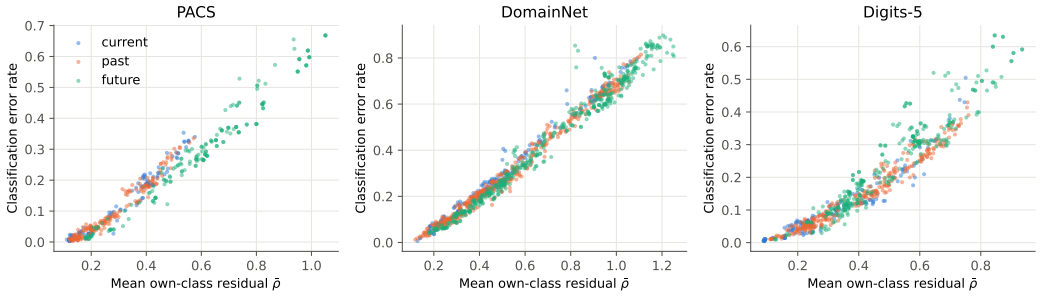


Figure 6: Relationship between mean own-class replay-hull residual and classification error.

Figure 6 examines whether the replay-hull residual reflects prediction difficulty across continual stages. For each evaluation domain, we compute the mean distance of samples to their ground-truth class replay hull and compare it with the corresponding classification error. Each point represents one domain evaluated at a particular stage and is categorized as current, past, or future. Results are collected over all 10 domain orders and three random seeds. Across PACS, DomainNet, and Digits-5, larger residuals are consistently associated with higher error rates. Future domains exhibit greater dispersion, while still following the same overall relationship between residual and error. This suggests that distance from the accumulated class-wise replay geometry provides a useful indicator of prediction difficulty across adaptation, generalization, and forgetting alleviation.

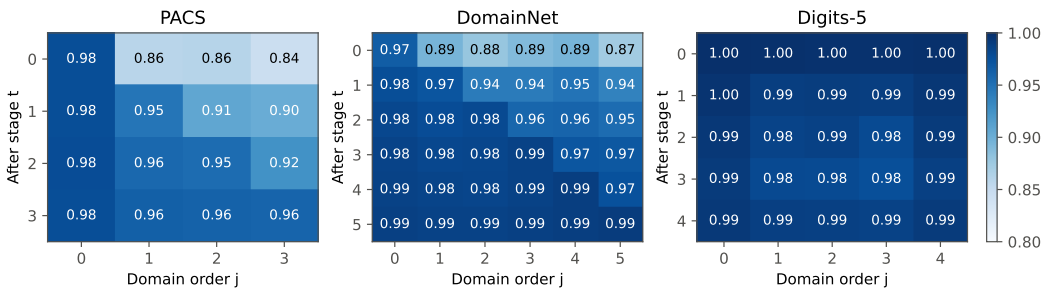


Figure 7: Fraction of representation variance captured by the fixed source PCA subspace.

C.6 Variance Preservation in the Fixed PCA Subspace

We further examine whether the fixed PCA space constructed at the source stage remains representative as the encoder evolves. For each stage and evaluation domain, we measure the fraction of representation variance preserved after projection onto the fixed source PCA basis. The results are averaged over all 10 domain orders and three random seeds. In Figure 7, the diagonal entries correspond to the current domain at each stage, while the lower and upper triangular regions represent previously observed and future domains, respectively. Most of the representation variance is retained throughout continual learning, with particularly high preservation on Digits-5 and progressively increasing coverage on PACS and DomainNet. Notably, the upper triangular region also maintains relatively high retained variance, indicating that the fixed source PCA space captures meaningful representation structure even for domains that have not yet been observed. Together, these results suggest that the fixed source PCA space remains broadly representative across current, past, and future domains.