Domain-Conditioned Class Imbalance: Rethinking Class Balance in Multi-Domain Learning

Yongkun Deng[1] **Nan Yang**[1,2*] **Xiatong Guo**[1] **Zhiyong Wang**[1] **Dong Yuan**[1*]
[1]Faculty of Engineering, The University of Sydney
Darlington, NSW 2008, Australia
[2]School of Science, Engineering and Digital Technologies, University of Southern Queensland
Darling Heights, QLD 4350, Australia
{yongkun.deng, n.yang, zhiyong.wang, dong.yuan}@sydney.edu.au
xguo0045@uni.sydney.edu.au nan.yang@unisq.edu.au

Abstract

Unlabeled image collections often span several visual domains, and the same class can be common in one domain and rare in another. Balanced class totals and balanced domain sizes, the common checks in dataset construction, do not rule this out: the allocation across domain–class pairs can remain highly skewed. We call this regime Domain-Conditioned Class Imbalance (DCI). Standard corrections are derived from global class counts and apply the same per-class adjustment in every domain. Under DCI the class priors differ across domains, and we show that no single domain-independent additive adjustment can match all of them at once. We introduce HyperIDAC, which estimates a soft domain prior for each image and uses a hypernetwork to generate an instance-specific, domain-conditioned classifier. A complementary multi-level pseudo-labeling module combines teacher predictions, label history, semantic fallback, and consistency learning to limit error reinforcement in sparse domain–class pairs. For controlled evaluation, we fix the class and domain marginals and vary only the joint domain–class allocation, which isolates the effect of DCI. Across four multi-domain benchmarks under original, long-tailed, and DCI regimes, HyperIDAC consistently outperforms the compared methods. These gains indicate that correcting class imbalance in multi-domain data requires conditioning on domain context rather than a single global adjustment.

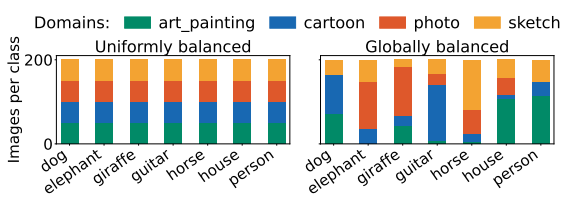
1 Introduction

Modern visual recognition systems increasingly operate on large uncurated image collections rather than carefully curated benchmarks. These collections are gathered passively, exhibit pronounced class imbalance, and, in many cases, are almost entirely *unlabeled*, since obtaining exhaustive annotations at scale is costly and slow. This reality contrasts sharply with the setting assumed by most deep learning successes, where training data are densely labeled and roughly class balanced. A rich body of work on class imbalance tackles this issue in labeled datasets through resampling [1–3], reweighting [4–7], and related techniques [8–14]. All of these approaches, however, rely on access to ground truth labels to estimate class frequencies, define loss weights, or calibrate decision boundaries. When data are unlabeled and heavily imbalanced, these quantities become hidden, and imbalance turns into a silent but persistent source of bias in representation learning and downstream prediction. Beyond this, a more fundamental challenge arises from the multi-domain nature of real-world data.

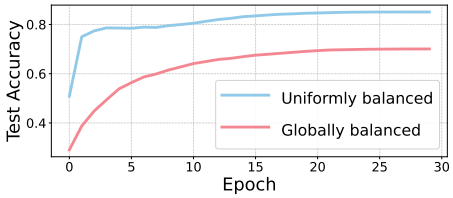
In many real-world applications, data do not arise from a single domain but from a collection of visual domains with distinct appearance statistics and, critically, different class frequencies. For instance,

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Class–domain distributions for two sampling schemes with the same per-domain and per-class totals: **uniformly balanced** (left, equal images per class in every domain) and **globally balanced** (right, equal totals per class but uneven allocation across domains).



(b) Performance gap between **Uniformly balanced** (blue) and **Globally balanced** (red) settings on the PACS dataset.

Figure 1: **Marginal balance can hide Domain-Conditioned Class Imbalance.**

daytime images typically contain clear illumination and many vehicles and pedestrians, whereas nighttime images are dominated by low light, strong contrasts, and a larger share of traffic lights or reflective objects. If class frequencies were independent of domain, auditing global class totals would be sufficient. However, when class frequencies vary with domain context, class imbalance is no longer a single global property of the dataset: a class that appears frequent when aggregated over all domains may be rare in specific domains, and conversely a globally rare class may dominate a particular domain. This makes dataset construction and auditing subtle: marginal balancing is a reasonable and common practice for making multi-domain datasets suitable for training, but total samples per class and per domain do not determine how samples are allocated across domain–class pairs. Directly auditing the full domain–class table is often infeasible in unlabeled or weakly labeled data, where class labels are unavailable at collection time. Even when labels are later obtained for curation, enforcing balanced counts for every domain–class cell can discard many valid samples from naturally frequent cells. Methods that correct only global class imbalance therefore struggle to adapt, since they ignore how class frequencies vary across domains.

To empirically demonstrate the impact of class imbalance in a multi-domain setting, we conduct a controlled study on the PACS dataset [15] with 7 classes across 4 domains. As shown in Fig. 1, we compare two data regimes that share identical global class distributions (200 samples per class) and identical per domain sample sizes (350 samples per domain), differing only in how samples are allocated across domain–class combinations: (1) **Uniformly balanced**, where each class has exactly 50 samples in each domain, and (2) **Globally balanced**, where class counts are balanced globally and per-domain totals remain constant, but individual domain-class combinations vary significantly (e.g., the class "horse" has 118 samples in one domain but only 6 in another). Training the same model under these two settings leads to a marked performance gap: accuracy drops from 85.06% in the Uniformly balanced case to 70.05% in the Globally balanced one, despite identical global histograms. This gap indicates that in multi-domain learning, performance is governed not only by global class counts but also by how classes are distributed across domains.

We call this regime *Domain-Conditioned Class Imbalance* (DCI): the class and domain marginals are balanced, yet the joint domain–class allocation is skewed. The balanced marginals in Fig. 1 serve as a diagnostic control: by removing global long-tailedness and unequal domain sizes, they make the effect of skewed joint domain–class allocation directly measurable. The formal definition is given in Section 3.1. The gap in Fig. 1 reflects a general limitation of corrections based on global class statistics. Under balanced class totals they treat every class alike, and more generally they apply one adjustment to every domain while the class-prior term of the decision differs from domain to domain, so no single domain-independent additive correction can match the class priors of every domain (Proposition 1 in Appendix B). For a learner without labels, this limitation shows up in three ways. (1) **Domain-specific imbalance patterns**: each domain has its own class distribution, so the required correction differs from domain to domain and no global reweighting or resampling can serve all domains at once. (2) **Extreme data scarcity**: some domain–class combinations (e.g., rare classes in low-resource domains) have very few samples, which makes it difficult to learn reliable decision boundaries for these cases with standard empirical risk minimization. (3) **Distribution blind spots**: in unlabeled settings, the learner cannot directly observe which domain–class combinations are underrepresented, yet it must discover and compensate for these blind spots to avoid collapsing predictions toward dominant patterns.

In this work, we propose **Hypernetwork-based Imbalanced Domain Adaptive Classifier (HyperIDAC)**, a framework designed around the failure mode exposed above. If imbalance is hidden in domain–class pairs, then a single global correction is insufficient, and a domain-agnostic classifier has no explicit mechanism to target the affected domain–class pairs. HyperIDAC therefore follows three design principles. First, it estimates domain context for each unlabeled sample by combining visual features with lightweight zero-shot domain predictions, producing a domain-aware prior. Second, it uses this prior to generate sample-conditioned classifier parameters and domain-conditioned class embeddings, allowing decision boundaries to vary with domain context rather than applying one global correction to all samples. Third, it uses multi-level pseudo-label management to prevent scarce domain–class pairs from being suppressed by early noisy predictions: high-confidence samples use teacher or historical labels, uncertain samples can fall back to CLIP semantic scores, and very low-confidence samples are trained through consistency. The first two principles form *Dynamic Domain Learning* (DDL), which learns domain-conditioned decision boundaries, and the third is *Multi-Level Pseudo Label Management* (PLM), which stabilizes supervision for sparse domain–class pairs. Together, these components turn the hidden conditional skew of DCI into an explicit conditioning signal for classifier adaptation. The contributions of this work can be summarized as:

- We identify and formalize *Domain-Conditioned Class Imbalance* (DCI), a hidden conditional imbalance in unlabeled multi-domain learning where class and domain marginals may appear balanced but the joint domain–class allocation remains skewed, creating challenges that go beyond conventional single-domain class imbalance.
- We propose **HyperIDAC**, a unified framework that addresses DCI through domain-aware priors, sample-conditioned classifier generation, and semantics-guided pseudo labeling, allowing classifier adaptation to depend on domain context while better supporting under-represented domain–class pairs.
- Experiments on four multi-domain benchmarks under original, long-tailed, and DCI regimes show that HyperIDAC consistently improves over strong unsupervised baselines, demonstrating the importance of domain-aware classifier adaptation under conditional class imbalance.

2 Related Work

Long-tailed recognition. Class imbalance in visual recognition has been addressed through resampling [1–3], reweighting and margin/logit calibration [4–7, 16–18, 12, 13, 19], decoupled classifier training [20–22], stronger pretraining or task-specific designs [23, 24], and self-supervised or label-efficient strategies [25, 26]. These approaches typically treat class imbalance as a global property of the training set and correct it without modeling domain context. However, many real-world multi-domain applications exhibit class imbalance that varies across domains, calling for domain-conditioned rather than purely global treatments. Closest to this direction, MDLT [27] studies supervised multi-domain long-tailed learning over domain–class pairs. DCI instead focuses on unlabeled, marginally balanced data where joint skew remains hidden after class/domain balancing.

Class-imbalanced domain adaptation. Unsupervised domain adaptation transfers knowledge from a labeled source domain to an unlabeled target domain [28–30], commonly through feature alignment [31–34], class-conditional alignment [30, 35], pseudo-labeling or self-training [36, 37], and source-free or test-time adaptation [38, 39]. Class-imbalanced domain adaptation further studies source–target label-prior mismatch [40–44], but it mainly estimates a target-level correction or adapts over a source–target pair. HyperIDAC instead learns from multiple unlabeled domains simultaneously, where each domain can require a different class correction and no single target prior is sufficient.

Vision-language priors and hypernetwork-based adaptation. CLIP [45] provides zero-shot semantic priors and has been adapted with residual adapters [46], cache-based few-shot tuning [47], and weight interpolation [48], but the resulting classifier is typically fixed or domain-agnostic. Hypernetworks generate adaptive parameters [49] and have been used in meta-learning [50, 51], continual learning [52, 53], neural architecture search [54, 55], and domain-related generative adaptation [56, 57]. Generated or conditioned parameters also appear in few-shot learning and domain generalization. HyperShot [58] generates classifiers from labeled support sets, SPG [59] and L2C [60] generate prompts with generators trained on labeled source domains, and bi-level meta-learning [61] learns domain-specific subspaces. These methods rely on labels or known domain groupings and do not model class priors, and we do not claim such primitives as new. Unlike prior CLIP adaptation

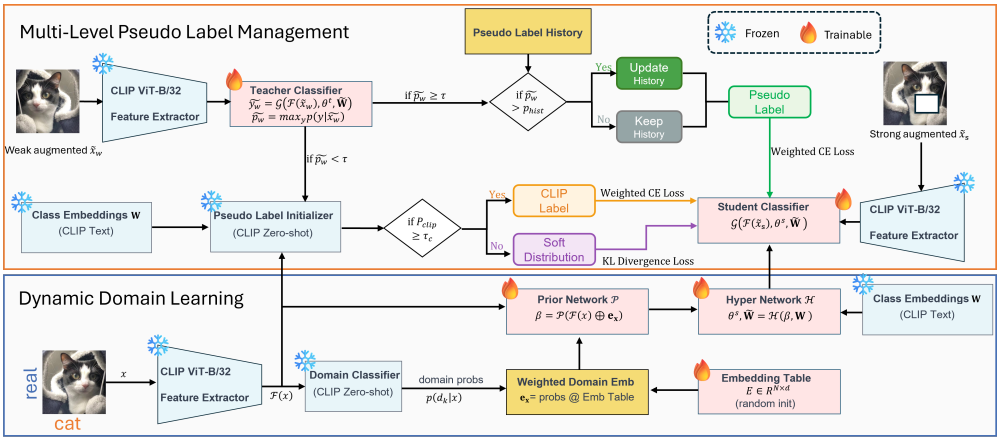


Figure 2: **Overview of the HyperIDAC framework.** (Top) Multi-Level Pseudo Label Management in a teacher–student setup: high-confidence samples use teacher or historical pseudo-labels, uncertain samples fall back to multi-template CLIP zero-shot predictions, and very low-confidence samples are trained with KL consistency instead of being discarded. (Bottom) Dynamic Domain Learning builds a domain prior from image features and zero-shot domain predictions, and uses a hypernetwork to turn this prior into instance-specific classifier weights and domain-aware class embeddings for the student classifier. The teacher is updated via EMA.

or generative hypernetwork methods, HyperIDAC couples CLIP-derived semantic/domain priors with hypernetwork-generated classifier parameters to adapt decision boundaries under unlabeled multi-domain class imbalance.

A more detailed discussion of related work is provided in Appendix C.

3 Method

3.1 Problem Formulation

Domain-Conditioned Class Imbalance. Given an unlabeled multi-domain dataset with domains $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$ and classes $\mathcal{C} = \{1, \ldots, C\}$, let $\mathcal{D}_i^c$ denote the set of samples belonging to class c in domain $\mathcal{D}_i$. A common dataset construction or auditing protocol is to balance marginal statistics, such as the total number of samples per domain and the total number of samples per class. However, in multi-domain data, these two marginals do not determine the joint domain–class allocation. We define *Domain-Conditioned Class Imbalance* (DCI) as the failure mode where a dataset appears balanced under these marginal checks but remains imbalanced after conditioning on domain. Formally, DCI occurs when: (1) each domain contains an equal number of samples, i.e., $|\mathcal{D}_i| = |\mathcal{D}_{i'}|$ for all i, i'; (2) the global class distribution is balanced, i.e., $\sum_{i=1}^{K} |\mathcal{D}_i^c| = \sum_{i=1}^{K} |\mathcal{D}_i^{c'}|$ for all c, c'; but (3) the per-domain class distributions vary significantly, i.e., $|\mathcal{D}_i^c|$ differs substantially across domains i for the same class c. Conditions (1) and (2) correspond to the marginal balance criteria commonly used during dataset construction, while Condition (3) exposes the remaining skew in the joint domain–class allocation. Our goal is to learn a robust classifier under this hidden conditional imbalance using only pseudo labels and zero-shot supervision during training.

3.2 Overview

HyperIDAC consists of three interacting components (Fig. 2). (1) A **domain-aware prior network** that fuses CLIP image features with zero-shot domain predictions to produce domain-conditioned priors. (2) A **hypernetwork-based dynamic classifier** that generates instance-specific classifier parameters using these priors, enabling domain-adaptive decision boundaries. (3) A **multi-level pseudo labeling module** that integrates teacher predictions, historical pseudo labels, and CLIP zero-shot scores to obtain reliable supervision for both majority and minority classes. Components (1) and (2) form Dynamic Domain Learning (DDL), and component (3) is Multi-Level Pseudo Label

Management (PLM). Together, these components allow the student network to be trained solely from pseudo labels and zero-shot information in the presence of domain-varying class imbalance.

3.3 Domain-Aware Prior Network

Our motivation experiment (Fig. 1) demonstrates that Domain-Conditioned Class Imbalance causes about a 15 percentage-point accuracy drop even under controlled global and per-domain balance. Unlike prior methods that enforce domain-invariant representations through adversarial training, we argue for explicit domain conditioning: the classifier should adapt based on which domain a sample comes from. Hence, we introduce a domain-aware prior network to explicitly model and leverage domain characteristics, enabling the classifier to handle domain-specific patterns while addressing class imbalance.

CLIP Feature Encoder. We leverage a pre-trained CLIP model [45] as our feature extractor $\mathcal{F} : \mathcal{X} \to \mathbb{R}^D$. The CLIP encoder is kept frozen to preserve its rich semantic representations from large-scale vision-language pre-training. For an input image x, we denote the visual feature as $\mathcal{F}(x)$.

Domain Classifier. To explicitly model domain information, we employ a zero-shot domain classifier based on CLIP's text encoder. For each domain k, we construct textual prompts using domain-specific descriptors (*e.g.*, "a clipart image"). The detailed templates used are provided in Appendix I.1. Let $\mathcal{T} = \{t_1, \ldots, t_K\}$ denote the text embeddings for K domains. With temperature parameter τ_d, the domain probability distribution is computed as:

$$p(d_k|x) = \frac{\exp(\langle \mathcal{F}(x), t_k \rangle / \tau_d)}{\sum_{j=1}^{K} \exp(\langle \mathcal{F}(x), t_j \rangle / \tau_d)}. \tag{1}$$

Although we use lightweight textual domain descriptors to obtain a soft posterior $p(d_k \mid x)$, HyperIDAC does not rely on predefined domain names or a fixed domain taxonomy. When domain descriptors are unavailable, we replace the prompt-based domain classifier with unsupervised latent domain discovery via either fixed-K K-means or an elbow-based variant that automatically selects K, and use the resulting cluster assignments as domain priors. Empirically, these cluster-based variants remain competitive across imbalance regimes and outperform the strongest main-table baseline in most settings. DomainNet results are reported in Table 3, and Office-Home results in Appendix I.5.

Domain Embedding. The domain probabilities $p(d_k|x)$ indicate which domains a sample is likely to come from, but do not encode the critical information for handling Domain-Conditioned Class Imbalance: *how class distributions differ across domains*. To capture these domain-class relationships, we introduce a learnable domain embedding table $\mathbf{E} \in \mathbb{R}^{K \times d_e}$, where d_e is the embedding dimension and the k-th row $\mathbf{E}_k$ represents domain k. This embedding is learned jointly with the classifier and is intended to capture both low-level visual characteristics and domain-specific class-frequency patterns that are important under DCI. The domain-weighted embedding for a sample is computed as:

$$\mathbf{e}_x = \sum_{k=1}^{K} p(d_k|x)\mathbf{E}_k, \tag{2}$$

which provides a soft, differentiable encoding of domain information.

Prior Network. To bridge domain information with visual features, we introduce a Prior Network $\mathcal{P}$ that fuses image features and domain embeddings into a unified prior representation:

$$\beta = \mathcal{P}(\mathcal{F}(x) \oplus \mathbf{e}_x) \tag{3}$$

where $\oplus$ denotes concatenation. The prior vector $\beta \in \mathbb{R}^{d_\beta}$ encodes both semantic content and domain-specific characteristics, serving as the control signal for dynamic parameter generation.

3.4 Hypernetwork and Dynamic Classifier

Standard linear classifiers use shared parameters across all samples, limiting their ability to adapt to domain-specific and class-specific instances. Inspired by hypernetworks [49, 62], we generate instance-specific classifier parameters conditioned on the prior β. Appendix B gives the motivation and relates this choice to feature-space alternatives.

Hypernetwork Architecture. Let $\mathcal{F}(x) \in \mathbb{R}^D$ be the frozen CLIP image feature and C the number of classes. For each instance, a hypernetwork $\mathcal{H}$ generates classifier parameters $\theta = \mathcal{H}(\beta)$, where $\theta \in \mathbb{R}^{d_\theta}$ contains all weights and biases. Since the classifier is linear, $d_\theta = C \cdot D + C$. In practice $\mathcal{H}$ is implemented as a single linear layer from $\mathbb{R}^{d_\beta}$ to $\mathbb{R}^{d_\theta}$. We then reshape θ into weight matrix $W_\beta \in \mathbb{R}^{C \times D}$ and bias vector $b_\beta \in \mathbb{R}^C$ and compute the base logits $\mathbf{z}_{\text{base}} = W_\beta \mathcal{F}(x) + b_\beta$.

Attribute Adapter. To further leverage class semantics, we introduce an Attribute Adapter A_ξ that adjusts class name embeddings based on the domain prior. Let $\mathbf{W} \in \mathbb{R}^{C \times D}$ be the CLIP text embeddings for all class names. We use a single refinement module that takes β as input to produce a control vector $g \in \mathbb{R}^D$ and modulates $\mathbf{W}$ into refined, domain-conditioned embeddings $\tilde{\mathbf{W}} \in \mathbb{R}^{C \times D}$:

$$g = \mathrm{ReLU}(\mathrm{LN}(\Phi\beta)), \tag{4}$$

$$\tilde{\mathbf{W}} = \mathrm{ReLU}((\mathbf{W}\Psi) \odot g), \tag{5}$$

where Φ, Ψ are learnable matrices and $\odot$ denotes element-wise multiplication with broadcasting over the class dimension so that g is shared across all C rows of $\mathbf{W}$. The adapter thus produces domain-conditioned class embeddings $\tilde{\mathbf{W}}$ used in the semantic branch.

Dynamic Classifier. Given feature $\mathcal{F}(x)$, the dynamic linear classifier applies the instance-specific parameters (W_β, b_β) produced by the hypernetwork to obtain the base logits $\mathbf{z}_{\text{base}}$. While this data-driven pathway learns discriminative boundaries from training samples, it suffers from unreliable estimates for minority classes with limited data. To address this, we introduce a complementary *semantic pathway* that leverages CLIP's pre-trained class embeddings as a stable prior. Specifically, we compute cosine similarity for each class $c \in \{1, \ldots, C\}$:

$$\mathbf{z}_{\text{cos}}[c] = \frac{\mathcal{F}(x)^\top \tilde{\mathbf{W}}_c}{\|\mathcal{F}(x)\| \|\tilde{\mathbf{W}}_c\|}, \tag{6}$$

where $\tilde{\mathbf{W}}_c$ is the domain-conditioned embedding for class c. Stacking these scores yields the semantic logit vector $\mathbf{z}_{\text{cos}} \in \mathbb{R}^C$. This semantic score provides reliable signals for rare classes, as CLIP embeddings offer equal semantic coverage across all classes regardless of training data frequency. The final prediction combines both pathways:

$$\mathbf{z} = \mathbf{z}_{\text{base}} + \lambda_{\text{cos}} \mathbf{z}_{\text{cos}}, \quad \lambda_{\text{cos}} = \mathrm{std}(\mathbf{z}_{\text{base}}) / \mathrm{std}(\mathbf{z}_{\text{cos}}), \tag{7}$$

where λ_{cos} performs scale matching between the data-driven logits and the semantic logits. When the base logits have a larger dispersion, the semantic logits are correspondingly amplified so that CLIP-derived class prototypes remain numerically competitive with the learned classifier. Conversely, when the base logits have a smaller scale, this normalization prevents the semantic branch from dominating solely due to magnitude mismatch.

3.5 Multi-Level Pseudo Label Management (PLM)

Teacher-Student Framework. Following Mean Teacher [63], we maintain student and teacher copies of the trainable adaptation modules. Let Θ^s and Θ^t denote the student and EMA teacher module parameters, respectively, including the domain embedding table, prior network, hypernetwork, and attribute adapter. The teacher is updated by exponential moving average:

$$\Theta^t \leftarrow \alpha_{\text{EMA}} \Theta^t + (1 - \alpha_{\text{EMA}}) \Theta^s, \tag{8}$$

where $\alpha_{\text{EMA}} = 0.99$. The teacher operates on weakly-augmented images $\tilde{x}_w$ for stable pseudo-labels, while the student trains on strongly-augmented images $\tilde{x}_s$.

Historical Pseudo-Label Tracking. To mitigate label noise from fluctuating teacher predictions, we maintain a per-sample historical buffer $\{(y^i, p^i)\}_{i=1}^N$, where y^i is the pseudo-label and p^i is the confidence for sample i, N is the total number of samples. For each sample, we update the history only when the current teacher prediction exceeds both a threshold τ and the historical confidence:

$$(y^i, p^i) \leftarrow \begin{cases} (\tilde{y}_w^i, \tilde{p}_w^i) & \text{if } \tilde{p}_w^i \geq \tau \text{ and } \tilde{p}_w^i > p^i \\ (y^i, p^i) & \text{otherwise,} \end{cases} \tag{9}$$

where $(\tilde{y}_w^i, \tilde{p}_w^i)$ are the teacher's current prediction and confidence. For the weak view $\tilde{x}_w^i$, the EMA teacher modules generate an instance-specific classifier head $\theta^{i,t}$ and adapted class embeddings $\tilde{\mathbf{W}}^i$. We write the prediction compactly as

$$(\tilde{y}_w^i, \tilde{p}_w^i) = \mathcal{G}\Big(\mathcal{F}(\tilde{x}_w^i), \theta^{i,t}, \tilde{\mathbf{W}}^i\Big), \tag{10}$$

where $\mathcal{G}$ returns the $\arg\max$ pseudo-label and the maximum softmax probability as confidence.

CLIP Fallback. For low-confidence samples ($\tilde{p}_w^i < \tau$) that are disproportionately from minority classes, we leverage CLIP's zero-shot classification with multiple prompt templates (Appendix I.2). The aggregated CLIP probability for class y given sample x is:

$$p_{\text{CLIP}}(y \mid x) = \frac{1}{M} \sum_{j=1}^{M} \text{softmax}_{c \in \mathcal{C}} \left(\frac{\langle F(x), t_{j,c} \rangle}{\tau_d} \right)_y, \tag{11}$$

where M is the number of templates, $F(x)$ is the CLIP image feature, $t_{j,c}$ is the text embedding of class c under template j, and τ_d is the temperature. We use the confidence $p_{\text{CLIP}}(x) = \max_y(p_{\text{CLIP}}(y|x))$ and pseudo-label $y_{\text{CLIP}} = \arg\max_y(p_{\text{CLIP}}(y|x))$.

3.6 Training Objective

Our training objective combines cross-entropy loss for high-confidence samples and KL divergence for low-confidence samples.

Sample Partitioning. For each sample, we choose its training strategy based on confidence. Let $\mathcal{B}$ be a training batch. We partition samples into three sets:

1. **Teacher-confident** ($\mathcal{I}_T$): $\{i : \tilde{p}_w^i \geq \tau\}$
2. **CLIP-confident** ($\mathcal{I}_C$): $\{i : \tilde{p}_w^i < \tau, p_{\text{CLIP}}(i) \geq \tau_c\}$
3. **Low-confident** ($\mathcal{I}_L$): $\{i : \tilde{p}_w^i < \tau, p_{\text{CLIP}}(i) < \tau_c\}$

where τ is the confidence threshold (linearly annealed from a lower to a higher value over training) and τ_c is the CLIP confidence threshold.

Weighted Cross-Entropy Loss. For high-confidence samples ($\mathcal{I}_T \cup \mathcal{I}_C$):

$$\mathcal{L}_{\text{CE}} = \frac{\sum_{i \in \mathcal{I}_T \cup \mathcal{I}_C} w_i \ell_{\text{CE}}(\mathbf{z}_s^i, y^i)}{\sum_{i \in \mathcal{I}_T \cup \mathcal{I}_C} w_i}, \tag{12}$$

with weights:

$$w_i = \begin{cases} \max\left(0.1, \left(\frac{p^i - \tau}{1 - \tau}\right)^2\right) & i \in \mathcal{I}_T \\ w_{\text{CLIP}}(t) \cdot p_{\text{CLIP}}(i) & i \in \mathcal{I}_C, \end{cases} \tag{13}$$

where $\mathbf{z}_s^i$ is the output logits of the student classifier, y^i is the corresponding pseudo label and $w_{\text{CLIP}}(t)$ is a time-decaying trust factor that decreases with epochs to reflect CLIP's diminishing reliability.

KL Divergence Loss. For low-confidence samples ($\mathcal{I}_L$), we enforce consistency with the teacher's temperature-scaled distribution ($T = 1.2$):

$$\mathcal{L}_{\text{KL}} = \frac{1}{|\mathcal{I}_L|} \sum_{i \in \mathcal{I}_L} D_{\text{KL}}(\text{softmax}(\mathbf{z}_t^i / T) \| \text{softmax}(\mathbf{z}_s^i)), \tag{14}$$

where $D_{\text{KL}}(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence, $\mathbf{z}_s^i$ and $\mathbf{z}_t^i$ denote the output logits of the student and teacher classifiers, respectively.

Total Loss.

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}, \tag{15}$$

where λ_{KL} is the weight of the KL loss. All hyperparameter values are provided in Appendix D.

Table 1: Average accuracy (%) of different methods on four datasets under three regimes. The best results are **bolded**, and the second-best are underlined. Each experiment was repeated three times, and mean values are reported. †ConMix + linear head (adapted). ‡Gran-LT (benchmark-only).

Method	Venue	VLCS			PACS			Office-Home			DomainNet		
		Original	*LTD*	*DCI*	*Original*	*LTD*	*DCI*	*Original*	*LTD*	*DCI*	*Original*	*LTD*	*DCI*
WVN [64]	IEEE Access'20	81.55	79.44	77.56	92.59	83.91	83.14	79.77	63.71	73.53	52.08	48.63	69.66
ImOOD [65]	NeurIPS'24	80.30	66.05	69.90	90.66	49.75	73.75	76.59	45.26	67.30	42.20	32.31	65.76
ConMix† [66]	ICLR'25	81.31	79.96	72.95	91.69	88.91	91.26	78.68	74.19	73.76	45.58	46.34	65.42
LOS [67]	ICLR'25	80.85	67.24	71.18	90.67	52.25	75.94	76.70	45.32	67.46	51.91	32.52	65.52
TAET [68]	CVPR'25	74.51	41.94	64.65	90.75	40.91	77.20	69.94	31.19	62.18	49.38	26.31	63.44
Gran-LT‡ [69]	CVPR'25	79.28	76.90	71.10	90.17	67.61	72.68	77.01	70.28	67.91	51.82	33.06	65.79
ARITH [70]	CVPR'25	79.21	81.17	78.40	90.67	89.39	87.20	59.93	62.47	63.95	43.54	43.95	55.65
HyperIDAC	—	**82.03**	**82.61**	**81.60**	**95.00**	**93.60**	**93.08**	**81.44**	**83.33**	**78.69**	**53.87**	**64.96**	**72.98**

4 Experiments

To evaluate the effectiveness of our method, we perform extensive experiments on four popular multi-domain datasets with three different imbalanced settings.

4.1 Experimental Settings

Dataset. We evaluate on four standard multi-domain benchmarks. Office-Home [71] contains about 15,500 images from 65 classes across four domains (Real World, Clipart, Product, Art). DomainNet [72] has roughly 0.6M images from 345 classes in six domains (Clipart, Infograph, Painting, Quickdraw, Real, Sketch). PACS [15] includes 9,991 images from 7 classes across four style domains (Photo, Art Painting, Cartoon, Sketch), and VLCS [73] has 10,729 images from 5 classes across VOC2007, LabelMe, Caltech101 and SUN09.

Dataset settings. We evaluate three regimes: (1) *Original.* Datasets are used as released. DomainNet is severely imbalanced, Office-Home moderately, while PACS and VLCS are nearly balanced. (2) *Long-Tailed Distribution (LTD).* We downsample to 100/50/7/5 classes with imbalance ratio $\mathrm{IR} = n_{\max}/n_{\min}$ of 50/20/10/5 for DomainNet/Office-Home/PACS/VLCS. (3) *Domain-Conditioned Class Imbalance (DCI).* We use 40/65/7/5 classes. Each domain has equal total samples and classes are globally balanced, but per-domain distributions are deliberately skewed. Sample sizes and construction details are provided in Appendix D, and the corresponding dataset distributions are visualized in Figs. 3 and 4.

Baseline. To investigate the impact of Domain-Conditioned Class Imbalance, we compare HyperIDAC with methods from both domain generalization/adaptation and long-tailed recognition. From the former, we include ARITH [70], which meta-learns domain-balanced parameters via arithmetic aggregation of domain-specific optima. From the latter, we consider WVN [64], ImOOD [65], ConMix [66], LOS [67], TAET [68], and Gran-LT [69], covering representative strategies such as classifier calibration, posterior regularization for imbalanced OOD detection, contrastive mixup for deep clustering, label over-smoothing, adversarial equalization, and category extrapolation for long-tailed recognition. A detailed introduction to these baselines is provided in Appendix D.

Implementation details. For a fair comparison, we use the same network architecture, optimizer, and learning rate schedule, and train HyperIDAC and all baselines for 30 epochs. A pretrained frozen CLIP model is adopted as the backbone for feature extraction, and a single linear classifier is attached on top for every method. For baselines that require ground-truth labels for supervision, we generate training labels using CLIP zero-shot classification. For HyperIDAC, the domain prior network is implemented as a two-layer MLP. The hypernetwork is a single linear layer that maps the prior vector β to the weights of the dynamic classifier, and we use an attribute adapter (another single linear layer) to refine the class embeddings. We set the confidence threshold τ from 0.8 to 0.9, the CLIP confidence threshold τ_c to 0.5, and the KL regularization weight λ_{KL} to 0.5. For ConMix [66], whose original protocol evaluates representations by K-means clustering with permutation matching, we append a linear classifier trained on detached features with the shared CLIP pseudo-labels, because the benchmark requires fixed semantic class predictions. For Gran-LT [69], ground-truth labels are replaced by the same CLIP pseudo-labels, as for all label-supervised baselines, and its external retrieval component is disabled because the benchmark provides no external data source, which may

Table 2: Ablation study results under three imbalance regimes on DomainNet.

Method	Original	LTD	DCI
CLIP	50.09	32.28	60.03
CLIP+DDL	52.32	61.53	70.01
CLIP+PLM	50.98	41.16	67.06
HyperIDAC	53.87	64.96	72.98

Table 3: Latent domain discovery on DomainNet without domain names.

Method	Original	LTD	DCI
K-means (fixed K)	53.47	61.53	71.78
K-means (elbow K)	53.58	61.20	70.93
HyperIDAC	53.87	64.96	72.98
Best main-table baseline	52.08	48.63	69.66

reduce its performance. Both rows are marked in Table 1, the adaptations are summarized in Table 6, and all other baselines run their original training procedures unchanged.

4.2 Main Results

We evaluate **HyperIDAC** against state-of-the-art methods on four benchmarks under three data regimes: Original, Long-Tailed Distribution (LTD), and Domain-Conditioned Class Imbalance (DCI). Overall, HyperIDAC attains the best mean accuracy on *every* dataset and setting in Table 1.

Original. On the original datasets, HyperIDAC achieves the best performance across all benchmarks, with particularly notable improvements on PACS and DomainNet. Compared to the strongest main-table baseline WVN, our approach shows gains of 2.41 percentage points (pp) on PACS and 1.79 pp on DomainNet.

Long-Tailed (LTD). Under severe class imbalance, HyperIDAC consistently ranks first across all datasets, improving over the best main-table baseline by up to +16.33 pp on DomainNet and +9.14 pp on Office-Home, which highlights strong robustness to heavy-tail skew in multi-domain learning.

Domain-Conditioned Class Imbalance (DCI). In the most challenging regime where per-domain class distributions differ while global counts are balanced, HyperIDAC again obtains the highest accuracy on all datasets (Tab. 1), with margins up to +4.93 pp on Office-Home and over +3 pp on both DomainNet (+3.32 pp) and VLCS (+3.20 pp), validating domain-conditioned calibration.

Discussion. LTD and DCI reveal the limits of methods that assume a single global class prior. HyperIDAC overcomes this by *conditioning the classifier on a domain prior*, which disentangles and compensates for per-domain sampling bias and delivers strong accuracy. Across all datasets and regimes, HyperIDAC ranks first for every dataset–setting pair, with improvements ranging from +0.48 pp to +16.33 pp and averaging about +4 pp over the strongest main-table baseline. These results indicate optimal performance under LTD/DCI requires a unified treatment of domain heterogeneity and class imbalance, which HyperIDAC provides via a domain-aware classifier.

Why LTD/DCI can exceed "Original". Accuracy under LTD and DCI can be higher than under the Original setting, e.g., DomainNet 53.87% $\rightarrow$ 64.96%/72.98%. A plausible explanation is task complexity: Original evaluates on all classes (e.g., DomainNet 345), whereas LTD/DCI use reduced subsets (100 and 40 for DomainNet), which lowers the label-space complexity despite the added imbalance. Cross-setting numbers are therefore not directly comparable. The meaningful signal is the within-setting gap, where HyperIDAC ranks first across all datasets and regimes. In Appendix E.2 and Section 4.4, we tighten the regime with stricter subsets and harder imbalance, and **HyperIDAC exhibits even larger margins in most cases**, showing that our method scales favorably as difficulty increases.

4.3 Ablation Study

Table 2 decomposes HyperIDAC into two main components on DomainNet: Dynamic Domain Learning (DDL) and Multi-Level Pseudo Label Management (PLM). Starting from a CLIP backbone with a single linear classifier baseline, DDL already brings clear gains in all three regimes, especially under heavy imbalance. In the long-tailed setting (LTD), accuracy jumps from 32.28% to 61.53% (+29.25 pp), and in the DCI setting from 60.03% to 70.01% (+9.98 pp), showing that conditioning the classifier on domain-aware priors is crucial for correcting domain-specific sampling bias. PLM alone offers more moderate but consistent improvements over CLIP (e.g., 41.16% on LTD and 67.06% on DCI), indicating that multi-level pseudo labels and CLIP fallback reduce confirmation

Table 4: Accuracy (%) on PACS under increasing DCI severity. Smaller α is more severe. UB: uniform balance.

α	UB	1.0	0.5	0.3
WVN	90.03	88.23	83.64	76.85
LOS	84.54	81.33	59.19	56.67
HyperIDAC	94.81	93.98	92.94	90.16

Table 5: Accuracy (%) on ImageNet-LT (1,000 classes), mean over three seeds. Full results in Appendix E.5. Markers as in Table 1.

Method	Top-1	Many	Medium	Few
WVN	41.04	59.78	33.00	16.30
ISFDA	35.38	37.87	34.70	30.77
ConMix†	39.56	55.20	32.89	18.73
LOS	45.71	62.40	40.44	17.01
HyperIDAC	**55.06**	**62.45**	**52.80**	**42.13**

bias, particularly for underrepresented domain–class pairs. Combining both modules in the full HyperIDAC framework yields the best performance in every regime, reaching 53.87%, 64.96%, and 72.98% on Original, LTD, and DCI respectively, and further improving over the strongest single-component variant (Dynamic Domain Learning) by +3.43 pp and +2.97 pp on LTD and DCI. These results confirm that domain-aware dynamic classifiers and robust pseudo-label management are complementary and jointly necessary to address class imbalance in multi-domain learning. Table 3 further examines whether predefined domain names are necessary. Replacing prompt-based domain descriptors with unsupervised K-means clusters remains competitive on DomainNet and still outperforms the strongest main-table baseline in all three regimes. This indicates that HyperIDAC does not critically rely on exact domain names or a fixed prompt taxonomy. The appendix also compares HyperIDAC with ISFDA and BoDA, two baselines from class-imbalanced domain adaptation and multi-domain long-tailed recognition, under the same protocol (Appendix E.1). Further ablations are in the appendix: hyperparameter analysis in Appendix G (accuracy varies by at most 2.21 pp across the sweeps), architecture sensitivity in Appendix H, and full latent-domain discovery results on Office-Home in Appendix I.5.

4.4 DCI Severity and Scalability

DCI severity. We vary DCI severity on PACS by sampling domain-wise class proportions from a Dirichlet distribution with concentration α, while the number of samples per domain and the global class totals stay fixed (Appendix E.3). From uniform balance (UB) to $\alpha = 0.3$, HyperIDAC drops by 4.65 pp, compared with 13.18 pp for WVN and 27.87 pp for LOS, so its margin over WVN grows from 4.78 to 13.31 pp as the joint allocation becomes more skewed (Table 4).

Scaling to 1,000 classes. On ImageNet-LT [74] with all 1,000 classes (single domain, $K = 1$), HyperIDAC reaches 55.06% Top-1, 9.35 pp above LOS, the next-best method, and 25.12 pp higher on few-shot classes, while the two are tied on many-shot classes (Table 5). The unfactorized generator produces all $1000 \times (512 + 1)$ classifier coefficients for every sample and trains on a single 24-GB GPU.

5 Conclusion

In this work, we studied class imbalance in unlabeled multi-domain learning and identified Domain-Conditioned Class Imbalance (DCI), where performance is strongly affected by how classes are allocated across domains even when global class and domain totals are balanced. To address this regime, we proposed HyperIDAC, a hypernetwork-based classifier that conditions on domain-aware priors and uses multi-level pseudo-labeling to better support minority classes. Experiments on four multi-domain benchmarks under three regimes showed consistent gains over strong baselines. These results indicate that resolving multi-domain class imbalance requires explicit domain conditioning rather than purely global corrections, and we hope that the DCI setting and HyperIDAC will serve as a useful basis for future work on realistic multi-domain learning.

Acknowledgments and Disclosure of Funding

This work is partly supported by the Australian Research Council Linkage Project LP220200893 and Discovery Project DP250103783. The authors declare no competing interests.

References

[1] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[2] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[3] Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, 2003.

[4] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[6] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[7] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 735–744, 2021.

[8] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10795–10816, 2023.

[9] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6897–6907, 2022.

[10] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

[11] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[12] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[13] Junjiao Tian, Yen-Cheng Liu, Nathaniel Glaser, Yen-Chang Hsu, and Zsolt Kira. Posterior re-calibration for imbalanced datasets. *Advances in neural information processing systems*, 33:8101–8113, 2020.

[14] Tong Wu, Ziwei Liu, Qingqiu Huang, Yu Wang, and Dahua Lin. Adversarial robustness under long-tailed distribution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8659–8668, 2021.

[15] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[16] Jiawei Ren, Cunjun Yu, shunan sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4175–4186. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/2ba61cc3a8f44143e1f2f13b2b729ab3-Paper.pdf`.

[17] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pages 4334–4343. PMLR, 2018.

[18] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[19] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6626–6636, 2021.

[20] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020.

[21] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021.

[22] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020.

[23] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018.

[24] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *European conference on computer vision*, pages 728–744. Springer, 2020.

[25] Hong Liu, Jeff Z. HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. URL `https://openreview.net/forum?id=vUz4JPRLpGx`.

[26] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19290–19301. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/e025b6279c1b88d3ec0eca6fcb6e6280-Paper.pdf`.

[27] Yuzhe Yang, Hao Wang, and Dina Katabi. On multi-domain long-tailed recognition, imbalanced domain generalization and beyond. In *European conference on computer vision*, pages 57–75. Springer, 2022.

[28] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[29] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[30] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018.

[31] 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.

[32] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[33] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

[34] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[35] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4893–4902, 2019.

[36] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6028–6039. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/liang20a.html`.

[37] Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. Source-free domain adaptation via distribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7212–7222, June 2022.

[38] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

[39] Shiqi Yang, yaxing wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29393–29405. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f5deaeeae1538fb6c45901d524ee2f98-Paper.pdf.

[40] Tzu Ming Harry Hsu, Wei Yu Chen, Cheng-An Hou, Yao-Hung Hubert Tsai, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Unsupervised domain adaptation with imbalanced cross-domain data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4121–4129, 2015.

[41] Shuhan Tan, Xingchao Peng, and Kate Saenko. Class-imbalanced domain adaptation: An empirical odyssey. In *European Conference on Computer Vision*, pages 585–602. Springer, 2020.

[42] Xinhao Li, Jingjing Li, Lei Zhu, Guoqing Wang, and Zi Huang. Imbalanced source-free domain adaptation. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 3330–3339, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386517. doi: 10.1145/3474085.3475487. URL https://doi.org/10.1145/3474085.3475487.

[43] Sunghyun Park, Seunghan Yang, Jaegul Choo, and Sungrack Yun. Label shift adapter for test-time adaptation under covariate and label shifts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16421–16431, 2023.

[44] Korawat Tanwisuth, Xinjie Fan, Huangjie Zheng, Shujian Zhang, Hao Zhang, Bo Chen, and Mingyuan Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:17194–17208, 2021.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

[46] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

[47] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pages 493–510. Springer, 2022.

[48] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022.

[49] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

[50] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. *Advances in neural information processing systems*, 29, 2016.

[51] Woojin Cho, Kookjin Lee, Donsub Rim, and Noseong Park. Hypernetwork-based meta-learning for low-rank physics-informed neural networks. *Advances in Neural Information Processing Systems*, 36: 11219–11231, 2023.

[52] Johannes von Oswald, Christian Henning, Benjamin F. Grewe, and João Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1906.00695.

[53] Hamed Hemati, Vincenzo Lomonaco, Davide Bacciu, and Damian Borth. Partial hypernetworks for continual learning. In Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup, editors, *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pages 318–336. PMLR, 22–25 Aug 2023. URL https://proceedings.mlr.press/v232/hemati23a.html.

[54] Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkgW0oA9FX.

[55] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Smash: one-shot model architecture search through hypernetworks. In *International Conference on Learning Representations*, 2018.

[56] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022.

[57] Aibek Alanov, Vadim Titov, and Dmitry P Vetrov. Hyperdomainnet: Universal domain adaptation for generative adversarial networks. *Advances in Neural Information Processing Systems*, 35:29414–29426, 2022.

[58] Marcin Sendera, Marcin Przewięźlikowski, Konrad Karanowski, Maciej Zięba, Jacek Tabor, and Przemysław Spurek. Hypershot: Few-shot learning by kernel hypernetworks. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2468–2477, 2023.

[59] Shuanghao Bai, Yuedi Zhang, Wanqi Zhou, Zhirong Luan, and Badong Chen. Soft prompt generation for domain generalization. In *European Conference on Computer Vision*, 2024.

[60] Zhixiang Chi, Li Gu, Huan Liu, Ziqiang Wang, Yanan Wu, Yang Wang, and Konstantinos Plataniotis. Learning to adapt frozen CLIP for few-shot test-time domain adaptation. In *International Conference on Learning Representations*, pages 66359–66380, 2025.

[61] Xiaorong Qin, Xinhang Song, and Shuqiang Jiang. Bi-level meta-learning for few-shot domain generalization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15900–15910, 2023.

[62] Qingsheng Wang, Lingqiao Liu, Chenchen Jing, Hao Chen, Guoqiang Liang, Peng Wang, and Chunhua Shen. Learning conditional attributes for compositional zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11197–11206, 2023.

[63] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[64] Byungju Kim and Junmo Kim. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8:81674–81685, 2020.

[65] Kai Liu, Zhihang Fu, Sheng Jin, Chao Chen, Ze Chen, Rongxin Jiang, Fan Zhou, Yaowu Chen, and Jieping Ye. Rethinking out-of-distribution detection on imbalanced data distribution. *Advances in Neural Information Processing Systems*, 37:109152–109176, 2024.

[66] Zhixin Li and Yuheng Jia. Conmix: Contrastive mixup at representation level for long-tailed deep clustering. In *The Thirteenth International Conference on Learning Representations*, 2025.

[67] Siyu Sun, Han Lu, Jiangtong Li, Yichen Xie, Tianjiao Li, Xiaokang Yang, Liqing Zhang, and Junchi Yan. Rethinking classifier re-training in long-tailed recognition: Label over-smooth can balance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0eKp3AdiVO.

[68] Wang Yu-Hang, Junkang Guo, Aolei Liu, Kaihao Wang, Zaitong Wu, Zhenyu Liu, Wenfei Yin, and Jian Liu. Taet: Two-stage adversarial equalization training on long-tailed distributions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15476–15485, June 2025.

[69] Shizhen Zhao, Xin Wen, Jiahui Liu, Chuofan Ma, Chunfeng Yuan, and Xiaojuan Qi. Learning from neighbors: Category extrapolation for long-tail learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30483–30492, June 2025.

[70] Xiran Wang, Jian Zhang, Lei Qi, and Yinghuan Shi. Balanced direction from multifarious choices: Arithmetic meta-learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 30577–30587, June 2025.

[71] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

[72] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019.

[73] Chen Fang, Ye Xu, and Daniel N. Rockmore. Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[74] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2532–2541, 2019.

[75] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

A Limitations and Broader Impact

Limitations. This work focuses on visual multi-domain classification benchmarks and controlled imbalance regimes. Although the Original, LTD, and DCI settings allow us to isolate the interaction between domain heterogeneity and class imbalance, they are built on benchmark-style domains that are given or discoverable as groups. When domains vary continuously or domain boundaries are highly ambiguous, additional mechanisms for finer-grained domain discovery or continual adaptation may be needed. Finally, scaling to substantially larger label spaces or streaming data requires further efficiency and reliability analysis.

Broader Impact. This work highlights domain-conditioned class imbalance (DCI) as a common but often overlooked issue in multi-domain visual recognition. In practice, dataset construction often balances only aggregate class counts or domain counts, but such marginal balance can still hide severe domain-conditioned imbalance; ignoring DCI may therefore lead to overly optimistic evaluation and degraded performance on domain-specific tail classes. By formalizing DCI and providing a benchmark protocol and adaptation method, this work encourages DCI-aware dataset construction, fine-grained robustness evaluation, and analysis beyond average accuracy. We do not expect direct negative societal impacts comparable to high-risk generative systems, as this work is a benchmark study built on public visual datasets and does not introduce new personal data or generate content.

B A Formal Observation on DCI

Let $\pi_d(y) = p(y \mid d)$ denote the class prior in domain d. By Bayes' rule, the log-odds between two classes y and y' for a sample x from domain d decompose as

$$\log \frac{p(y \mid x, d)}{p(y' \mid x, d)} = \log \frac{p(x \mid y, d)}{p(x \mid y', d)} + \log \frac{\pi_d(y)}{\pi_d(y')}. \tag{16}$$

A global prior correction, such as logit adjustment with one class prior, adds a single vector $b \in \mathbb{R}^C$ to the logits of every sample, whatever its domain. The question is whether one such vector can reproduce the class-prior term of Eq. (16) in every domain.

Proposition 1. *Assume $\pi_d(y) > 0$ for every domain d and class y. If a vector $b \in \mathbb{R}^C$ satisfies*

$$b_y - b_{y'} = \log \frac{\pi_d(y)}{\pi_d(y')} \qquad \textit{for every } d,\ y,\ y', \tag{17}$$

then $\pi_d = \pi_{d'}$ for all domains d and d'.

Proof. Fix a reference class y_0. By Eq. (17), $\pi_d(y) = \pi_d(y_0) \exp(b_y - b_{y_0})$ for every class y. Summing over y and using $\sum_y \pi_d(y) = 1$ gives

$$\pi_d(y_0) = \Big(\sum_y \exp(b_y - b_{y_0}) \Big)^{-1}, \tag{18}$$

which does not depend on d. Hence $\pi_d(y) = \pi_d(y_0) \exp(b_y - b_{y_0})$ does not depend on d either. □

Under DCI the within-domain class distributions differ across domains (Section 3.1), so by Proposition 1 no single domain-independent additive vector reproduces the class-prior term of Eq. (16) in every domain. If a class has zero prior in one domain and a positive prior in another, the log-ratio in Eq. (17) is infinite in the first domain, and no finite b can match it. Corrections computed from global class statistics do even less under DCI, where the global class counts are balanced by definition (Section 3.1): class-frequency reweighting or resampling gives every class the same factor, and global logit adjustment adds the same offset to every class. Neither changes the relative treatment of classes.

Scope. Proposition 1 concerns exact representation of the domain-specific prior term, not accuracy. It motivates a correction that can vary with domain context, which HyperIDAC provides through the generated bias b_β in $\mathbf{z}_{\text{base}} = W_\beta \mathcal{F}(x) + b_\beta$ (Section 3.4). When $p(x \mid y, d)$ also differs across domains, the likelihood-ratio term in Eq. (16) can vary with d as well, so a bias shift alone need not suffice, and the generated weights W_β let the feature-dependent part of the logits adapt too. A domain-conditioned feature transformation is a valid alternative. For example, feature-wise modulation followed by a shared linear classifier can be interpreted as a constrained form of context-dependent classifier weights. We generate (W_β, b_β) because this directly parameterizes both context-dependent class offsets and decision-boundary orientations. We do not claim that a hypernetwork is the only way to provide this capacity.

C Related Work

Long-tailed Recognition. Class imbalance, where minority classes are underrepresented, has been widely studied [8–11, 5]. Existing methods broadly fall into resampling and reweighting. Resampling [1, 2, 23, 24, 3] adjusts the data distribution by oversampling minority classes or undersampling majority ones (e.g., SMOTE [1] synthesizes new minority samples by interpolation). Reweighting [4–6, 16, 17] instead operates at the loss level, such as Focal Loss [4] and Class-Balanced Loss [5], while LDAM [18] introduces label-distribution-aware margins. Other lines include decoupled representation/classifier training [20–22], logit adjustment for prior-shift calibration [13, 19, 16], and self-supervised strategies that exploit unlabeled data [25, 26]. These approaches typically treat class imbalance as a global property of the training set and correct it without modeling domain context. However, many real-world multi-domain applications exhibit class imbalance that varies across domains, calling for domain-conditioned rather than purely global treatments.

Class-imbalanced Domain Adaptation. Unsupervised domain adaptation (UDA) [28–30] transfers knowledge from a labeled source domain to an unlabeled target domain, primarily by learning domain-invariant representations via feature alignment [31–34]. Beyond classical alignment, recent work explores class-conditional alignment [30, 35], pseudo-label/self-training [36, 37], and source-free or test-time adaptation [38, 39] to handle complex domain shifts. Most UDA methods assume identical class priors across source and target, whereas in realistic scenarios target domains often exhibit different class frequency distributions. Class-imbalanced Domain Adaptation (CDA) [41, 42, 40] addresses this mismatch. Early work [40] proposed CCSL to preserve class structure under differing class ratios or even numbers of classes. Tan et al. [41] systematically analyzed CDA, showed that many UDA methods degrade under joint domain and label shifts, and introduced COAL, which jointly aligns feature and label distributions to reduce majority-class bias. Subsequent studies incorporate reweighting and curriculum-style strategies; for instance, Li et al. [42] proposed a source-free method that adjusts classifier thresholds and progressively alleviates imbalance without source data. Park et al. [43] proposed a Label Shift Adapter that feeds an estimated target label distribution into a small adapter network to update classifier parameters at test time. PCT [44] further improves UDA under class-imbalance and source-private settings, but still follows the source–target adaptation paradigm. However, real-world systems often operate over multiple domains **simultaneously**, where classes can exhibit distinct imbalance patterns in each domain, calling for domain-conditioned, instance-specific corrections.

Vision-Language Models. Vision-language models, particularly CLIP [45], have revolutionized transfer learning by learning joint image–text embeddings from large-scale data, enabling zero-shot classification via similarity between image features and textual class names. Building on CLIP, many methods adapt it to downstream tasks using lightweight residual adapters [46], cache-based few-shot tuning [47], or weight interpolation between zero-shot and fine-tuned models for robustness–adaptation trade-offs [48]. However, real-world applications often span multiple visual domains with distinct styles and biases: although CLIP offers strong zero-shot performance and

good semantic coverage for rare classes, it applies a domain-agnostic classifier, whereas practical deployments increasingly require predictions that are conditioned on domain context.

Hypernetworks for Adaptive Learning. Hypernetworks [49] are meta-networks that generate parameters for other networks, enabling sample- or task-adaptive models. The original work showed that a small auxiliary network can dynamically produce weights for a larger model, achieving competitive performance with far fewer explicit parameters. Since then, hypernetworks have been applied in meta-learning [50, 51], continual learning [52, 53], and neural architecture search [54, 55]. More recently, they have also been explored in domain-related tasks: HyperStyle [56] uses a hypernetwork to modulate StyleGAN weights for cross- and out-of-domain image adaptation, while HyperDomainNet [57] predicts domain-specific modulation vectors for efficient multi- and unseen-domain adaptation in generative models. These developments underscore a growing practical need for hypernetworks that generate classifier parameters conditioned on domain context, particularly in multi-domain scenarios with heterogeneous data distributions.

Relation to Multi-Domain Long-Tailed Recognition. Multi-Domain Long-Tailed Recognition (MDLT) [27] is the closest prior formulation to our problem. MDLT studies supervised learning from multi-domain imbalanced data, where each domain can have its own long-tailed label distribution and the goal is to generalize across all domain–class pairs. It highlights an important insight shared by our work: in multi-domain data, the relevant unit of imbalance is often the domain–class pair rather than the class alone. However, DCI focuses on a different failure mode. Instead of starting from explicitly long-tailed multi-domain data, DCI studies the case where the class marginal and domain marginal are balanced under common dataset construction or auditing protocols, yet the joint domain–class allocation remains skewed. Thus, DCI is a hidden imbalance under marginal balance, rather than a general MDLT setting. Moreover, MDLT methods such as BoDA assume supervised access to class and domain labels and use domain–class statistics such as $N_{d,c}$ and feature means $\mu_{d,c}$ to align representations. In contrast, HyperIDAC addresses unlabeled multi-domain learning, where true domain–class counts are unavailable during training. It therefore relies on zero-shot domain/semantic priors, domain-conditioned classifier generation, and pseudo-label management rather than supervised domain–class alignment.

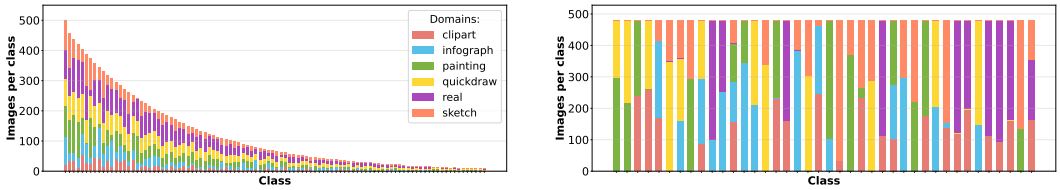
Taken together, prior work either assumes labeled data with globally defined class imbalance or addresses pairwise domain shifts with domain-agnostic classifiers, and thus does not directly cover unlabeled multi-domain regimes where class frequencies differ across domains. This gap motivates HyperIDAC, which uses domain context as a structural prior, leverages vision–language semantics, and dynamically adapts classifier parameters to Domain-Conditioned Class Imbalance.

D Implementation Details

Dataset. Office-Home [71] comprises around 15,500 images from 65 object categories, collected across four visually diverse domains: Real World, Clipart, Product, and Art. It captures variations between daily office and home scenes. **DomainNet** [72] is a large-scale benchmark featuring roughly 0.6 million images spanning 345 classes and six heterogeneous domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. Its scale and diversity make it one of the most challenging multi-domain datasets. **PACS** [15] contains 9,991 images covering 7 object categories shared across four stylistically distinct domains: Photo, Art Painting, Cartoon, and Sketch. The dataset is well known for its pronounced appearance and texture shifts. **VLCS** [73] integrates samples from four existing datasets (VOC2007, LabelMe, Caltech101, and SUN09), containing 10,729 images across 5 shared object classes. The four corresponding domains reflect natural distributional shifts among real-world image sources.

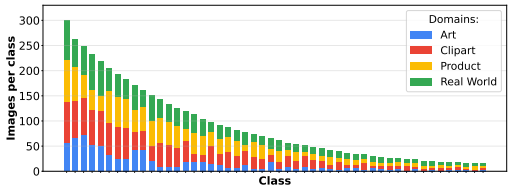
Dataset settings. Among the four datasets, DomainNet naturally exhibits severe class imbalance, while Office-Home presents a moderate level of imbalance. In contrast, PACS and VLCS are considered nearly balanced. To comprehensively evaluate the effectiveness of our approach, we design two additional data regimes applied to all four datasets beyond their original distributions.

(1) *Long-Tailed Distribution.* We define the imbalance ratio (IR) of a dataset as $\mathrm{IR} = \frac{n_{\max}}{n_{\min}}$, where $n_{\max}$ and $n_{\min}$ denote the number of samples in the largest and smallest classes, respectively. For DomainNet, we randomly sample 12,000 images from 100 classes with an imbalance ratio of 50. For Office-Home, 4,500 images are sampled from 50 classes with an imbalance ratio of 20. For PACS,

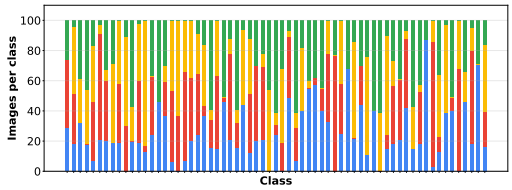


(a) DomainNet: Long-Tailed Distribution

(b) DomainNet: DCI Distribution



(c) Office-Home: Long-Tailed Distribution



(d) Office-Home: DCI Distribution

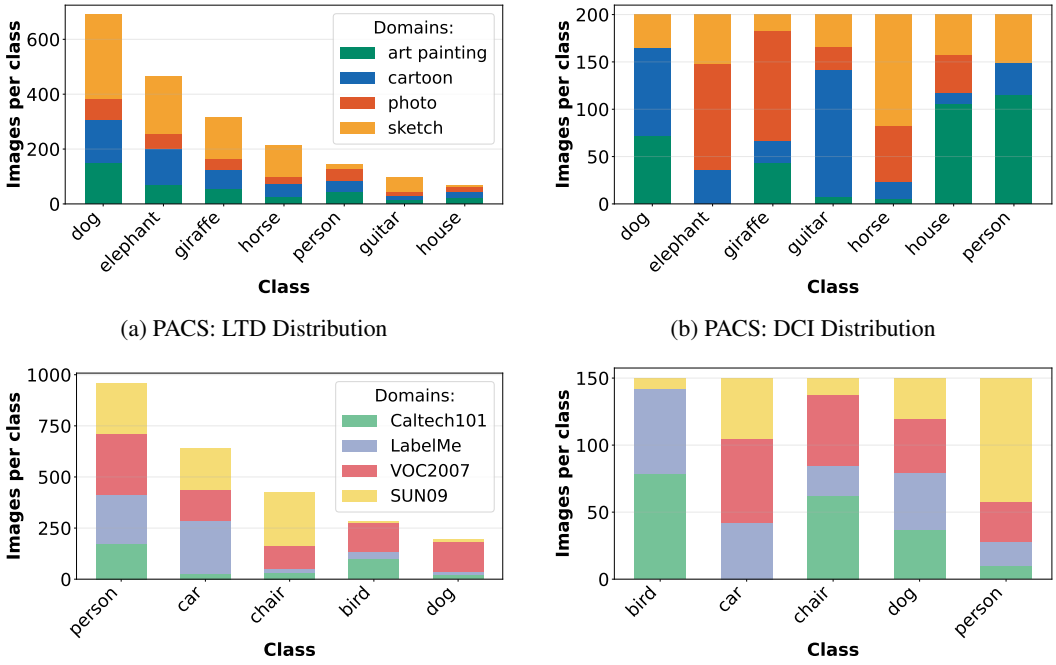
Figure 3: The class-domain distributions of datasets DomainNet and Office-Home under Long-Tailed (LTD) Distribution and Domain-Conditioned Class Imbalance (DCI) Distribution.

we sample 2,000 images across all 7 classes with IR $= 10$, and for VLCS, we sample 2,500 images from all 5 classes with IR $= 5$.

(2) *Domain-Conditioned Class Imbalance (DCI).* As illustrated in Fig. 1 and discussed in Section 3.1, Domain-Conditioned Class Imbalance refers to a multi-domain scenario where each domain contains an equal total number of samples and the overall class distribution remains globally balanced, but the per-domain class distributions are highly uneven across domains for the same class. To construct this setting, we create subsets for each dataset. For DomainNet, we sample 19,200 images from 40 classes, where each domain contains 3,200 images and each class has 480 samples. For Office-Home, 6,500 images are sampled from 65 classes, with 1,625 images per domain and 100 images per class. For PACS, we sample 1,400 images from 7 classes, with 350 images per domain and 200 images per class. For VLCS, we sample 750 images from 5 classes, with 187 images per domain and 150 images per class. Due to dataset size constraints, some domains may contain one image more or less. The detailed data distributions are visualized in Fig. 3 and Fig. 4.

Baseline. To investigate the impact of Domain-Conditioned Class Imbalance, we include baselines from both domain generalization/adaptation and long-tailed recognition literature. *Domain generalization and adaptation.* **Arith** [70] proposes an arithmetic meta-learning framework that balances domain-specific gradients by estimating the centroid of domain-optimal parameters, improving generalization under domain shift. *Long-tailed recognition.* **WVN** [64] adjusts classifier weight norms to rebalance decision boundaries between frequent and rare classes without additional training. **ImOOD** [65] introduces a generalized statistical framework that mitigates class-aware bias in imbalanced OOD detection via a unified training-time regularization on posterior probabilities. **ConMix** [66] introduces a representation-level contrastive mixup that implicitly re-balances head and tail class losses, enabling effective long-tailed deep clustering without supervision. **LOS** [67] revisits classifier re-training in long-tailed recognition and proposes a simple label over-smoothing strategy that balances logits magnitudes without requiring prior class distribution knowledge. **TAET** [68] proposes a two-stage adversarial equalization framework that improves both accuracy and robustness under long-tailed distributions. **Gran-LT** [69] introduces category extrapolation that leverages LLMs to discover visually related fine-grained neighbor classes, improving long-tail learning by enhancing data granularity and representation diversity.

Implementation details. For a fair comparison, we use the same network architecture, optimizer, and learning rate schedule, and train HyperIDAC and all baselines for 30 epochs. A pretrained frozen CLIP model is adopted as the backbone for feature extraction, and a single linear classifier is attached on top for every method. For baselines that require ground-truth labels for supervision, we generate training labels using CLIP zero-shot classification. For HyperIDAC, the domain prior network is implemented as a two-layer MLP. The hypernetwork is a single linear layer that maps the prior vector β to the weights of the dynamic classifier, and we use an attribute adapter (another single linear layer) to refine the class embeddings. We linearly anneal the confidence threshold τ from 0.8 to



(a) PACS: LTD Distribution

(b) PACS: DCI Distribution

(c) VLCS: LTD Distribution

(d) VLCS: DCI Distribution

Figure 4: The class-domain distributions of datasets PACS and VLCS under Long-Tailed (LTD) Distribution and Domain-Conditioned Class Imbalance (DCI) Distribution.

Table 6: Baseline adaptations under the benchmark protocol.

Baseline	Original protocol	Adaptation	Reason
ConMix	Unsupervised deep clustering, evaluated by K-means clustering with permutation matching	Linear readout on detached features, trained with the shared CLIP pseudo-labels. Projection head and view construction adjusted to the benchmark input pipeline	The benchmark requires fixed semantic class predictions rather than cluster assignments
Gran-LT	Long-tailed learning with an external category and image retrieval component	Ground-truth labels replaced by the shared CLIP pseudo-labels. External retrieval component disabled	No ground-truth labels and no external data source in the benchmark setting

0.9 over training, set the CLIP confidence threshold τ_c to 0.5, the distribution temperature $\tau_d = 1.2$ for both the zero-shot domain classifier and CLIP-based pseudo-labeling, the time-decaying trust factor $w_{\text{CLIP}}(t)$ from 1.0 to 0.5, and the KL regularization weight λ_{KL} to 0.5. The teacher network is updated with EMA coefficient $\alpha_{\text{EMA}} = 0.99$, and the temperature used in the KL divergence loss is set to $T = 1.2$.

E Additional Experiments

We present additional experiments to further evaluate HyperIDAC, including comparisons with additional baselines, evaluation under more sample-scarce long-tailed regimes, results on the single-domain CIFAR100-LT benchmark, and an analysis of computational cost. Unless otherwise stated, all models are trained for 30 epochs.

Table 7: Average accuracy (%) of ISFDA, BoDA, and HyperIDAC on four datasets under three regimes. The best result in each column is **bolded**.

Method	Venue	VLCS			PACS			Office-Home			DomainNet		
		Original	*LTD*	*DCI*	*Original*	*LTD*	*DCI*	*Original*	*LTD*	*DCI*	*Original*	*LTD*	*DCI*
ISFDA [42]	MM'21	66.43	65.49	62.23	83.57	86.04	85.15	77.12	81.75	75.56	44.02	58.47	66.57
BoDA [27]	ECCV'22	79.79	81.41	80.77	93.23	93.27	91.90	73.79	79.24	75.19	50.89	61.29	69.22
HyperIDAC	—	**82.03**	**82.61**	**81.60**	**95.00**	**93.60**	**93.08**	**81.44**	**83.33**	**78.69**	**53.87**	**64.96**	**72.98**

E.1 Comparison with ISFDA and BoDA

ISFDA and BoDA are representative baselines from related but assumption-mismatched settings. ISFDA [42] studies imbalanced source-free domain adaptation, where a classifier is first trained on a labeled source domain and then adapted to an unlabeled target domain without accessing source data. It addresses the joint presence of covariate shift and label shift through balanced source training, curriculum sampling, secondary pseudo-label correction, and intra-/inter-class feature regularization. BoDA [27] studies supervised Multi-Domain Long-Tailed Recognition (MDLT), where class and domain labels are available to construct domain-class pairs and estimate their feature statistics. It performs balanced domain-class distribution alignment by pulling same-class features across domains closer and separating different-class features within and across domains. In contrast, HyperIDAC targets unlabeled multi-domain learning, where domain-varying class frequencies must be inferred from pseudo-labels rather than obtained from class labels. We evaluate both baselines under the same protocol as our reproduced baselines.

The results are shown in Table 7. HyperIDAC outperforms both ISFDA and BoDA in all 12 dataset–regime settings. Compared with ISFDA, HyperIDAC improves the average accuracy by 9.23 percentage points. Compared with BoDA, HyperIDAC improves the average accuracy by 2.77 percentage points. Even when comparing against the stronger of ISFDA and BoDA in each column, HyperIDAC still achieves an average gain of 2.25 percentage points. BoDA is competitive on VLCS and PACS, but HyperIDAC maintains clear advantages on Office-Home and DomainNet, where the average gains over the stronger of ISFDA and BoDA are 3.01 and 3.47 percentage points, respectively. These results further support that explicitly modeling domain-conditioned priors and generating adaptive classifiers is beneficial under domain-varying class imbalance, especially when domain and class statistics must be inferred rather than directly supervised.

E.2 Long-Tailed Distribution with Fewer Samples

To investigate the effectiveness of our proposed HyperIDAC in extremely sample-scarce settings, we further reduce the number of samples used in the Long-Tailed Distribution (LTD) setting. The dataset configurations are as follows: for DomainNet, we randomly sample 6,000 images from 100 classes with an imbalance ratio of 50. For Office-Home, 2,000 images are sampled from 50 classes with an imbalance ratio of 30. For PACS, we sample 1,000 images across all 7 classes with $\mathrm{IR} = 10$, and for VLCS, we sample 1,000 images from all 5 classes with $\mathrm{IR} = 10$. The results are shown in Table 8. We observe that HyperIDAC remains the best-performing method on all four benchmarks even under this substantially more sample-scarce LTD regime. On VLCS and PACS, where the number of classes is relatively small, HyperIDAC still improves over the strongest competitor by +1.96 pp (vs. ARITH) and +1.65 pp (vs. ConMix), respectively. The gains become much larger on Office-Home and DomainNet, which contain more classes and domains and are therefore more challenging under heavy long-tailed skew with few samples: while the best baseline reaches 62.01% on Office-Home and 36.88% on DomainNet, HyperIDAC attains 80.07% and 58.67%, yielding margins of +18.06 pp and +21.79 pp. These results indicate that our domain-aware hypernetwork and multi-level pseudo-label management make more efficient use of scarce, imbalanced data than methods that rely on a single global classifier with resampling or pseudo-label heuristics, and that HyperIDAC scales favorably as both the tail severity and data scarcity increase.

E.3 Robustness to DCI Severity

We control DCI severity by sampling domain-wise class proportions from a Dirichlet distribution with concentration parameter α on PACS. Smaller α produces sharper and more mismatched class

Table 8: Average accuracy (%) of different methods on four datasets under the LTD regime with fewer samples. The best results are **bolded**, and the second-best are underlined. Markers as in Table 1.

Method	Venue	VLCS	PACS	Office-Home	DomainNet
		LTD	*LTD*	*LTD*	*LTD*
WVN [64]	IEEE Access '20	37.59	66.59	38.19	31.39
ISFDA [42]	MM'21	70.45	76.94	62.01	20.97
ImOOD [65]	NeurIPS'24	16.56	47.85	30.24	19.95
ConMix† [66]	ICLR'25	68.55	91.11	59.80	36.79
LOS [67]	ICLR'25	14.05	45.91	31.32	20.04
TAET [68]	CVPR'25	35.74	34.87	22.51	15.08
Gran-LT‡ [69]	CVPR'25	53.45	58.32	31.96	35.02
ARITH [70]	CVPR'25	76.36	61.21	33.72	36.88
HyperIDAC	—	**78.32**	**92.76**	**80.07**	**58.67**

Table 9: Accuracy (%) on CIFAR100-LT. The best results are **bolded**, and the second-best are underlined. Markers as in Table 1.

Method	Venue	CIFAR-100
WVN [64]	IEEE Access '20	55.25
ISFDA [42]	MM'21	60.32
ImOOD [65]	NeurIPS'24	46.11
ConMix† [66]	ICLR'25	53.88
LOS [67]	ICLR'25	46.73
TAET [68]	CVPR'25	36.02
Gran-LT‡ [69]	CVPR'25	49.28
ARITH [70]	CVPR'25	35.39
HyperIDAC	—	**63.07**

proportions across domains (i.e., more skewed $p(y \mid d)$). Following the construction in the main paper, we keep both the number of samples per domain and the global class totals fixed, so that only the severity of Domain-Conditioned Class Imbalance changes. We consider $\alpha \in \{1.0, 0.5, 0.3\}$ and additionally report the uniformly balanced (UB) case, where class proportions are identical across domains.

As shown in Table 4, all methods degrade as DCI becomes more severe. However, HyperIDAC degrades much more gracefully, dropping by only 4.65 percentage points from UB to $\alpha = 0.3$, compared with 13.18 for WVN and 27.87 for LOS. Moreover, HyperIDAC's advantage over WVN increases from 4.78 points in the UB case to 13.31 points at $\alpha = 0.3$, indicating that the proposed method is particularly robust under severe DCI.

E.4 Results on CIFAR100-LT

Although HyperIDAC is designed for Domain-Conditioned Class Imbalance in multi-domain learning, we also evaluate it on a single-domain long-tailed benchmark, CIFAR100-LT. Since CIFAR100-LT contains a single domain, we instantiate HyperIDAC with $K = 1$ and directly set the domain posterior to be degenerate, $p(d_1 \mid x) = 1$. Equivalently, this can be viewed as using one dummy domain. The original CIFAR100 dataset [75] consists of 50,000 training images and 10,000 test images of size 32×32 across 100 classes. Following the protocol in [18, 67], we construct CIFAR100-LT by applying an exponentially decaying sampling scheme to the training set so that per-class sample counts follow an imbalance ratio of 10. The resulting CIFAR100-LT split contains 19,573 training images, while the test set remains class-balanced and unchanged. The results are reported in Table 9. HyperIDAC achieves the highest accuracy on CIFAR100-LT with 63.07%, surpassing the strongest prior method ISFDA (60.32%) by **+2.75** percentage points and outperforming other recent approaches such as WVN (55.25%) and ConMix (53.88%) by larger margins. We hypothesize that this gain on CIFAR100-LT mainly stems from the structural inductive bias of the hypernetwork: generating classifier parameters via a hypernetwork introduces implicit regularization across classes, which may help stabilize tail-class learning in long-tailed settings.

Table 10: Accuracy (%) on ImageNet-LT with 1,000 classes, mean ± std over three seeds. Many, Medium, and Few denote classes with more than 100, 20 to 100, and fewer than 20 training images. Top-1 equals balanced accuracy on this test split. Best in **bold**. Markers as in Table 1.

Method	Top-1 / Bal. Acc.	Macro-F1	Many	Medium	Few
WVN	41.04 ± 0.20	36.67 ± 0.27	59.78 ± 0.21	33.00 ± 0.24	16.30 ± 0.21
ISFDA	35.38 ± 0.18	35.08 ± 0.20	37.87 ± 0.08	34.70 ± 0.13	30.77 ± 1.09
ConMix†	39.56 ± 0.17	35.53 ± 0.10	55.20 ± 0.06	32.89 ± 0.21	18.73 ± 0.37
LOS	45.71 ± 0.05	38.76 ± 0.06	62.40 ± 0.04	40.44 ± 0.10	17.01 ± 0.13
HyperIDAC	$\mathbf{55.06 \pm 0.16}$	$\mathbf{51.21 \pm 0.16}$	$\mathbf{62.45 \pm 0.05}$	$\mathbf{52.80 \pm 0.19}$	$\mathbf{42.13 \pm 0.38}$

Table 11: Online inference profile on the ImageNet-LT test set (50,000 images, same RTX 3090). Throughput and time are mean ± std over three full passes. WVN keeps the full CLIP object in its inference implementation. Markers as in Table 1.

Method	Throughput (images/s)	Time for 50k images (s)	Peak memory (GB)
WVN	975.4 ± 15.8	51.27 ± 0.84	0.72
ISFDA	989.5 ± 9.3	50.53 ± 0.47	0.47
ConMix†	977.0 ± 6.5	51.18 ± 0.34	0.46
LOS	987.6 ± 3.2	50.63 ± 0.16	0.46
HyperIDAC	973.5 ± 10.6	51.36 ± 0.56	1.11

E.5 Scalability to 1,000 Classes on ImageNet-LT

The generated classifier has $C(D+1)$ outputs, so the dense linear hypernetwork grows linearly with the number of classes C for fixed D and d_β. To test this scaling directly, we evaluate the single-layer HyperIDAC on ImageNet-LT [74] with all 1,000 classes. ImageNet-LT is single-domain, so we use the same $K = 1$ instantiation as for CIFAR100-LT. All methods follow the protocol of Section 4.1, with no ground-truth labels. We compare with WVN, ConMix, and LOS, three main-table baselines with distinct mechanisms (classifier calibration, contrastive clustering, and label over-smoothing), and with ISFDA, the strongest baseline on CIFAR100-LT (Table 9). BoDA, compared on the four multi-domain benchmarks in Appendix E.1, is not included here: its balanced domain–class alignment operates across multiple domains, so the $K = 1$ instantiation would remove its core mechanism, and its original protocol assumes observed class and domain labels. HyperIDAC keeps its unfactorized dense output with prior dimension $d_\beta = 128$: the generator maps $\mathbb{R}^{128}$ to all $1000 \times (512 + 1) = 513{,}000$ classifier coefficients without factorization, compression, or low-rank approximation, so a complete 1000×512 weight matrix and 1,000 biases are generated for every sample. Each method is trained for 30 epochs on a single 24-GB RTX 3090 with three seeds. Table 10 reports mean ± sample standard deviation. Top-1 accuracy equals balanced accuracy here because the test split holds 50 images per class, so we also report Macro-F1.

HyperIDAC is best in every column. Relative to LOS, the strongest baseline overall, it gains 9.35 pp in Top-1 and 25.12 pp on few-shot classes, while the two are tied on many-shot classes (62.45 versus 62.40). Relative to ISFDA, the strongest baseline on few-shot classes, it is 11.36 pp higher on that group. The ConMix row also serves as a control for the adaptation in Table 6: evaluating the same trained ConMix models through their original cluster-matching path gives 20.25 ± 0.28 Top-1, against 39.56 ± 0.17 for the appended linear readout, a paired gain of 19.31 pp averaged over the three seeds (18.94, 19.30, and 19.67). On this benchmark the adaptation raises rather than lowers the fixed-class accuracy of ConMix. We do not extrapolate this margin to other datasets.

The 30-epoch training times of WVN, ISFDA, ConMix, LOS, and HyperIDAC on the same GPU are 66.5, 129.0, 119.8, 65.1, and 110.6 minutes (one run per method). Table 11 profiles online inference on the 50,000-image test set with the same CLIP ViT-B/32 encoder for every method, in FP32 with autocast disabled, averaged over three full passes. At $C = 1000$ and $D = 512$ the generator of HyperIDAC is a single $\mathrm{Linear}(128, 513{,}000)$ layer with 66.18M parameters, which raises peak inference memory to 1.11 GB against 0.46 to 0.72 GB for the baselines. End-to-end time is 51.36 ± 0.56 s against 50.53 to 51.27 s, because all methods share the same CLIP encoding path.

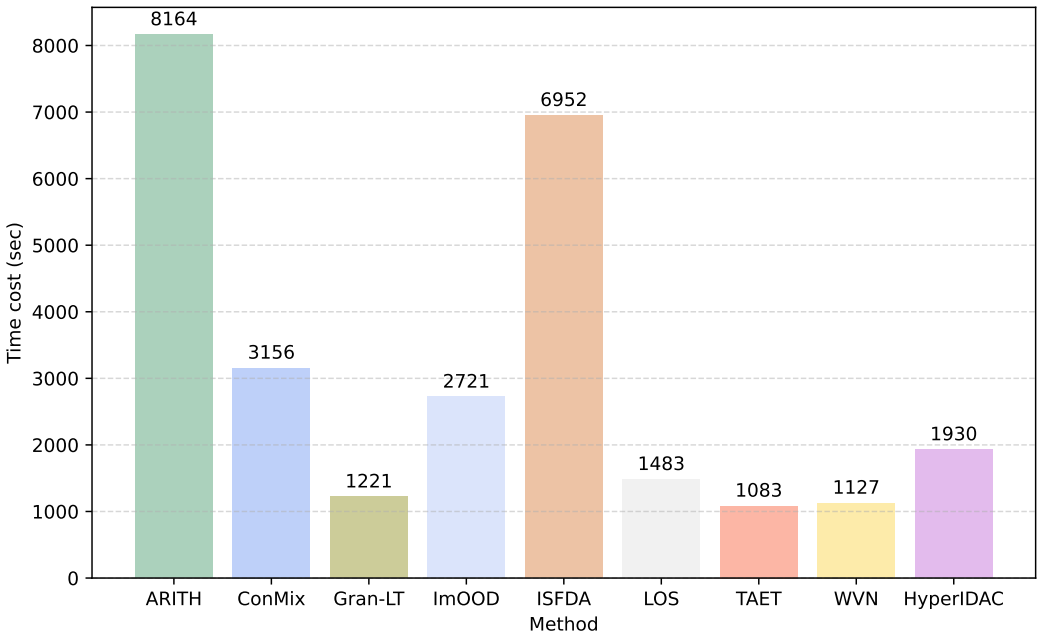


Figure 5: Training time of different methods on DomainNet-DCI setting over 30 epochs. ConMix and Gran-LT are the adapted versions marked † and ‡ in Table 1.

Table 12: Inference time (seconds) of different methods on DomainNet.

Method	WVN	LOS	TAET	HyperIDAC
Inference time (s)	246	286	205	207

These results show that the unmodified dense design runs at 1,000 classes on one consumer GPU. Substantially larger label spaces, such as tens of thousands of classes, would need factorized or low-rank parameter generation and are outside the demonstrated scope of this work.

E.6 Computational Cost

We compare the training time of all methods on the DomainNet-DCI setting using the same CLIP backbone and a unified 30-epoch schedule. As shown in Fig. 5, we report the wall-clock time measured on a single RTX 3090 GPU. Among the baselines, ARITH and ISFDA are the most computationally expensive, requiring 8164s and 6952s, respectively. ConMix† and ImOOD also incur relatively high cost (3156s and 2721s), whereas Gran-LT‡, LOS, TAET, and WVN are more efficient with 1221s, 1483s, 1083s, and 1127s. HyperIDAC completes training in 1930s, placing it in the mid-range: substantially faster than ARITH, ISFDA, and several heavier baselines, while being moderately slower than the most lightweight methods. Overall, HyperIDAC achieves a favorable balance between computational efficiency and accuracy across all regimes. Compared to the most expensive methods, HyperIDAC is around 4× faster, while maintaining only moderate overhead relative to the fastest approaches, demonstrating that the hypernetwork-based design achieves strong performance without prohibitive computational cost.

We further report inference time on DomainNet in Tab. 12. WVN, LOS, TAET, and HyperIDAC require 246s, 286s, 205s, and 207s, respectively, showing that HyperIDAC introduces no substantial inference overhead compared with other methods. This is expected because the additional hypernetwork is lightweight, while inference is dominated by the shared CLIP encoder used by all methods. Therefore, the proposed design does not introduce an additional deployment bottleneck.

Table 13: Per-class accuracy (%) of different variants on PACS (Long-Tailed setting). Avg. denotes macro accuracy averaged across all classes.

Method	dog	elephant	giraffe	horse	person	guitar	house	Avg.
CLIP	92.02	92.62	67.11	94.48	54.08	89.58	69.57	79.92
CLIP + Pseudo Label Mgmt.	90.17	94.74	75.10	96.62	91.49	97.93	98.62	92.10
CLIP + Dynamic Domain Learn.	82.82	97.88	84.61	98.12	95.71	97.93	99.47	93.79
HyperIDAC (full model)	88.72	97.10	85.76	96.95	93.22	98.65	99.79	94.31

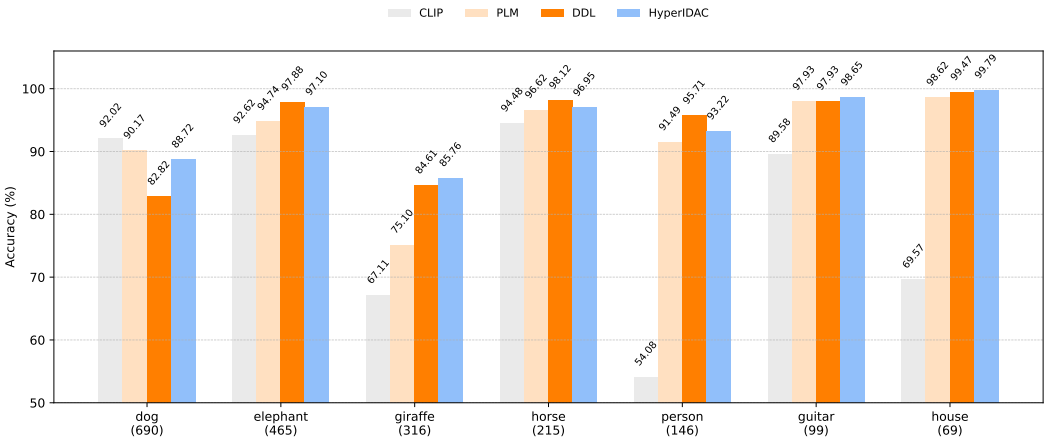


Figure 6: Per-class accuracy of HyperIDAC and its CLIP-based variants on PACS under the long-tailed setting. Classes are ordered by the number of training samples (in parentheses). CLIP denotes CLIP with a single linear classifier head. PLM augments CLIP with Pseudo Label Management. DDL augments CLIP with Dynamic Domain Learning.

E.7 Backbone Robustness

We further evaluate HyperIDAC with a smaller CLIP backbone by replacing ViT-B/32 with ResNet-101 on PACS. As expected, using the smaller backbone reduces performance across the compared methods. Nevertheless, HyperIDAC still outperforms two representative baselines under both LTD and DCI. Under LTD, HyperIDAC, WVN, and LOS achieve 65.59, 52.40, and 37.96, respectively. Under DCI, they achieve 68.75, 66.53, and 61.14, respectively. These results suggest that the benefit of HyperIDAC is not tied to a particular backbone choice.

F Per-class Accuracy Analysis

Per-class analysis on PACS–LTD (Fig. 6, Table 13) reveals that the gains of our components correlate strongly with the class frequency. The CLIP with a linear classifier head baseline achieves high accuracy on head classes such as *dog* and *elephant*, but its performance drops sharply on minority categories (e.g., 54.08% on *person* and 69.57% on *house*), resulting in a low macro accuracy of 79.92% and a large variance across classes (standard deviation 14.9 pp). This indicates that, under long-tailed and multi-domain conditions, the vanilla CLIP classifier heavily favours head classes and fails to provide balanced treatment of tail categories.

Adding Pseudo Label Management (PLM) primarily "rescues" those under-represented classes. PLM yields very large improvements on the most difficult categories (e.g., +37.41 pp on *person* and +29.05 pp on *house*, with additional gains on *giraffe* and *guitar*), while largely preserving the already strong performance on head classes (only a minor degradation on *dog*). As a result, the macro accuracy increases to 92.10% and the per-class standard deviation is almost halved (7.5 pp). This pattern suggests that PLM acts as a noise-robust, class-balancing supervision mechanism: by filtering and reweighting CLIP-derived pseudo labels, it is particularly effective when the supervised signal is scarce.

Table 14: Effect of τ_{start} on accuracy.

τ_{start}	0.4	0.5	0.6	0.7	0.8
Acc (%)	71.09	71.66	72.38	72.49	72.98

Table 15: Effect of τ_c on accuracy.

τ_c	0.4	0.5	0.6	0.7	0.8
Acc (%)	72.81	72.98	72.30	71.55	70.77

Dynamic Domain Learning (DDL) exhibits a complementary behaviour. It provides its largest gains on domain-shift–sensitive categories such as *person*, *giraffe* and *house* (e.g., +41.63 pp on *person* and +17.50 pp on *giraffe*), and further improves *elephant* and several tail classes, but substantially hurts the largest class *dog* (-9.20 pp). Overall, DDL increases the macro accuracy to 93.79% and further reduces the per-class standard deviation to 6.5 pp. We hypothesize that conditioning the classifier on domain priors helps reconcile multi-domain discrepancies for classes with strong stylistic variation, but may introduce unnecessary adaptation for data-rich categories.

The full HyperIDAC model combines PLM and DDL within a hypernetwork that generates domain-conditioned classifier weights, leading to the most balanced outcome. HyperIDAC attains the highest accuracy on several medium- and long-tail classes (e.g., *giraffe* 85.76%, *guitar* 98.65%, *house* 99.79%), keeps head-class performance competitive, and reduces the standard deviation of per-class accuracy to 4.9 pp. Consequently, its macro accuracy reaches 94.31%, outperforming all CLIP-based variants. In other words, HyperIDAC not only improves macro accuracy but also substantially reduces the accuracy gap across classes, providing a more equitable treatment of head and tail categories (a 67% reduction in per-class standard deviation, from 14.9 pp for CLIP to 4.9 pp) under long-tailed and multi-domain conditions.

G Hyperparameter Analysis

We ablate the main hyperparameters in our proposed HyperIDAC on the DomainNet–DCI setting (Tables 14 to 18). The threshold τ_{start} controls the initial confidence level required for teacher predictions to be stored in the historical pseudo-label buffer and is linearly annealed during training. The CLIP confidence threshold τ_c determines whether low-teacher-confidence samples are supervised by CLIP (set $\mathcal{I}_C$) or only used for KL consistency (set $\mathcal{I}_L$). The coefficient λ_{KL} balances the cross-entropy loss on high-confidence samples with the KL regularization on low-confidence ones. The temperature τ_d is used in both the zero-shot domain classifier and CLIP-based pseudo-labels to control the sharpness of the corresponding probability distributions. The time-dependent weight $w_{\text{CLIP}}(t)$ scales the contribution of CLIP-based pseudo-labels and gradually reduces their influence over training. Finally, in Table 19 we take our original zero-shot domain classifier (77.10% accuracy) and synthetically reduce its accuracy to several lower levels, then report the resulting training accuracy to isolate the impact of domain-prior quality on downstream classification.

As shown in Table 14, increasing τ_{start} from 0.4 to 0.8 yields monotonic improvements (71.09% $\rightarrow$ 72.98%), suggesting that stricter filtering of early pseudo-labels helps reduce noise accumulation and improves long-term training. The CLIP fallback threshold τ_c (Table 15) exhibits a peak at 0.5 (72.98%), with lower performance at both extremes: setting τ_c too low (0.4) introduces noisier CLIP predictions, while setting it too high (0.8) discards many samples from semantic supervision, leaving more data to rely on KL consistency alone.

The KL regularization weight λ_{KL} (Table 16) shows relatively stable performance across a broad range (0.4–1.0), with a slight optimum around 0.5 (72.98%). This robustness suggests that the multi-level pseudo-labeling framework is not overly sensitive to the exact balance between cross-entropy and KL terms, as long as low-confidence samples receive a reasonable amount of consistency regularization.

All main experiments use the default value $\tau_d = 1.2$ specified in Appendix D. Table 17 reports a sensitivity sweep on DomainNet–DCI, where increasing τ_d to 2.0 gives a small improvement in this setting. Overall, the results remain within about one percentage point across the tested range,

Table 16: Effect of λ_{KL} on accuracy.

λ_{KL}	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Acc (%)	72.26	72.98	72.78	72.77	72.76	72.74	72.58

Table 17: Effect of Temperature τ_d on accuracy.

τ_d	1.2	1.5	2.0	2.5
Acc (%)	72.98	73.74	73.96	73.92

indicating that HyperIDAC is not highly sensitive to this temperature. We kept $\tau_d = 1.2$ rather than selecting the best value of this sweep, because target accuracy is not available for such a choice on a new unlabeled dataset.

The time-decaying CLIP trust factor $w_{\text{CLIP}}(t)$ (Table 18) shows mild sensitivity, with performance remaining competitive when the initial weight varies from 1.0 to 0.7 (72.98% $\rightarrow$ 72.51%). This supports our design choice of starting from a relatively high CLIP weight and gradually reducing its influence as the model learns stronger data-driven boundaries. In Table 19, we then vary the zero-shot domain-classification accuracy and record the corresponding training accuracy: as domain-prior quality improves from 16.00% to 77.10%, training accuracy increases from 73.70% to 78.52%. This clear positive correlation (+4.82 pp) is consistent with the intuition that more accurate domain priors enable more effective hypernetwork-based parameter generation, underscoring the importance of reliable domain identification in multi-domain imbalanced learning.

Across a broad hyperparameter sweep, HyperIDAC remains consistently stable, with accuracy variations typically within 1–2 percentage points and a maximum spread of 2.21 pp, for τ_c. The spreads for λ_{KL}, τ_d, and $w_{\text{CLIP}}(t)$ are 0.72, 0.98, and 0.47 pp. Most hyperparameters show smooth trends with optima around reasonable mid-range values, indicating that the method does not rely on precise tuning and is overall insensitive to hyperparameter choices.

Use on a new unlabeled dataset. On a new unlabeled dataset we recommend starting from the reported defaults (τ annealed from 0.8 to 0.9, $\tau_c = 0.5$, $\lambda_{\text{KL}} = 0.5$, $\tau_d = 1.2$, and $w_{\text{CLIP}}(t)$ decaying from 1.0 to 0.5) rather than searching for a dataset-specific optimum with target labels. The resulting sample partitions can be monitored without labels: the proportions of teacher-confident, CLIP-confident, and KL-only samples, the update rate of the historical buffer, and the agreement between teacher and CLIP predictions. A configuration in which one supervision branch is nearly empty or dominates training shows up in these quantities, and the confidence thresholds can then be adjusted within the stable ranges above without reference to target accuracy. This is a label-free initialization and diagnostic procedure, not an automatic calibration method. Automatic calibration for deployment data that differ substantially from the evaluated benchmarks remains open.

H Architecture Sensitivity Analysis

Beyond the component-level ablations in the main paper, we further analyze whether HyperIDAC is sensitive to the depth of the hypernetwork and the attribute adapter. Specifically, we vary the depth of both modules from 1 to 5 and evaluate the resulting performance under the DomainNet-LTD setting.

As shown in Fig. 7, HyperIDAC remains highly stable across different depth choices. The overall variation is small, with a maximum gap of only 1.18 percentage points. Notably, stronger performance does not consistently require deeper variants. This suggests that the effectiveness of HyperIDAC mainly comes from the proposed domain-conditioned design, rather than from aggressive architectural tuning or increased module depth.

I Prompt Templates

For the zero-shot domain classifier, we use CLIP's standard image–text matching mechanism instantiated with domain prompts. For each domain (e.g., real, clipart, painting, sketch, infograph, quickdraw), we construct a descriptive textual template (e.g., "A clipart image, characterized by

Table 18: Effect of $w_{\text{CLIP}}(t)$ on accuracy.

$w_{\text{CLIP}}(t)$	1.0	0.9	0.8	0.7
Acc (%)	72.98	72.68	72.66	72.51

Table 19: Effect of domain classification accuracy on training accuracy.

Domain Cls Acc (%)	16.00	30.00	45.00	58.67	66.78	77.10
Train Acc (%)	73.70	77.71	77.84	78.26	78.40	78.52

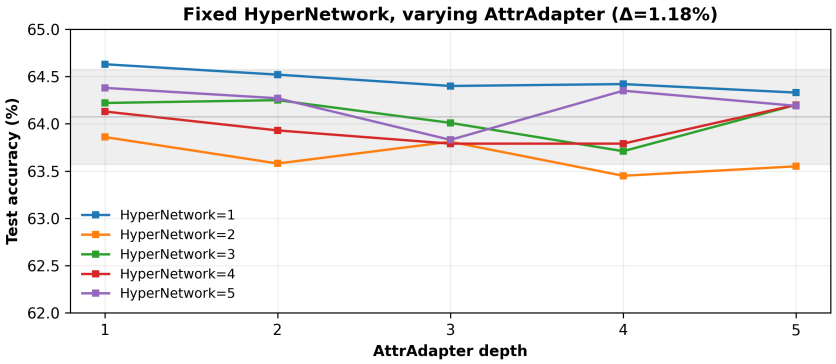


Figure 7: Architecture sensitivity of HyperIDAC to the depth of the hypernetwork and the attribute adapter. We vary both module depths from 1 to 5 and report the resulting performance. The results remain tightly clustered, with a maximum variation of only 1.18 percentage points, indicating that HyperIDAC is robust to these architectural choices.

simplified shapes, flat colors, and bold outlines.") and encode it using CLIP's text encoder to obtain a domain text feature. Given an image, we extract its visual feature with CLIP's image encoder and compute cosine similarities to all domain text features, followed by a temperature-scaled softmax with temperature τ_d to obtain a domain probability distribution $p(d_k \mid x)$ (as in Eq. (1)). These zero-shot domain probabilities are then used as inputs to the domain-aware prior network.

For CLIP-based zero-shot classification over semantic classes, we follow the same image–text matching pipeline but with class-name prompts and multiple templates. For each class name, we instantiate several textual templates (e.g., "A photo of a {}. A sketch image, characterized by hand-drawn lines, distinct shading, and a monochrome palette.") and encode them with CLIP's text encoder. Given an image, we extract its visual feature using CLIP's image encoder, compute cosine similarities to the text features of all templates separately, divide by the same temperature τ_d, and apply a softmax for each template. We then average the resulting probabilities across templates to obtain $p_{\text{CLIP}}(y \mid x)$ (as in Eq. (11)). The predicted class is the argmax of this distribution, and the maximum probability $p_{\text{CLIP}}(x) = \max_y p_{\text{CLIP}}(y \mid x)$ serves as the CLIP confidence used in our pseudo-labeling scheme.

I.1 Domain Classifier Templates

We use CLIP text prompts to construct a zero-shot domain classifier. The templates for each DomainNet domain are listed below:

- **real:** "A real image, characterized by high detail, natural lighting, and true-to-life colors."
- **clipart:** "A clipart image, characterized by simplified shapes, flat colors, and bold outlines."
- **painting:** "A painting image, characterized by visible brush strokes, rich textures, and vibrant colors."
- **sketch:** "A sketch image, characterized by hand-drawn lines, distinct shading, and a monochrome palette."

- **infograph:** "An infographic image, characterized by clear charts, structured layouts, and concise textual elements."
- **quickdraw:** "A quickdraw image, characterized by minimal lines, spontaneous strokes, and a casual doodle style."

I.2 CLIP Zero-shot Classification Templates

For CLIP-based class-name prompts, we append domain-style descriptors to each class name. Given a class name `{}`, the templates are:

- "A photo of a {}. A real image, characterized by high detail, natural lighting, and true-to-life colors."
- "A photo of a {}. A clipart image, characterized by simplified shapes, flat colors, and bold outlines."
- "A photo of a {}. A painting image, characterized by visible brush strokes, rich textures, and vibrant colors."
- "A photo of a {}. A sketch image, characterized by hand-drawn lines, distinct shading, and a monochrome palette."
- "A photo of a {}. An infographic image, characterized by clear charts, structured layouts, and concise textual elements."
- "A photo of a {}. A quickdraw image, characterized by minimal lines, spontaneous strokes, and a casual doodle style."

I.3 Relation to the Main Paper

The text embeddings obtained from the above prompts are used to construct the class text embeddings and domain text embeddings in the main paper. For semantic classes, the prompt-derived CLIP text features provide the textual representations used to form the class embedding matrix $\mathbf{W}$. Similarly, each domain prompt is encoded once to obtain a domain text embedding, corresponding to one column of $\mathcal{T}$ used by the zero-shot domain classifier.

I.4 Prompt Robustness

To assess whether HyperIDAC is sensitive to prompt design, we conduct additional experiments on DomainNet with simplified prompt templates. For the simplified domain prompt, we replace the descriptive domain template with "A {domain} image." For the simplified class prompt, we replace the class template with "A photo of {class}." In the last setting, both simplifications are applied simultaneously.

As shown in Tab. 20, replacing the default domain prompt with a simpler template leads to only minor changes across the Original, LTD, and DCI settings. Simplifying the class prompt or applying both simplifications introduces some fluctuation, but all variants still outperform the best-performing baseline in the main paper under the corresponding DomainNet setting. These results suggest that HyperIDAC is reasonably robust to prompt design and does not depend critically on carefully crafted prompt wording.

Table 20: Effect of prompt simplification on DomainNet.

Prompt setting	**Original**	**LTD**	**DCI**
Default prompts	53.87	64.96	72.98
Simplified domain prompt	54.22	64.54	72.89
Simplified class prompt	52.65	62.63	71.08
Both simplified	52.66	63.13	71.63

Table 21: Performance without domain names using unsupervised latent domain discovery on DomainNet and Office-Home.

DomainNet			
Method	**Original**	**LTD**	**DCI**
K-means (fixed K)	53.47	61.53	71.78
K-means (elbow-based K)	53.58	61.20	70.93
HyperIDAC	53.87	64.96	72.98
Best main-table baseline	52.08	48.63	69.66
Office-Home			
Method	**Original**	**LTD**	**DCI**
K-means (fixed K)	79.63	80.03	75.27
K-means (elbow-based K)	79.64	79.73	75.43
HyperIDAC	81.44	83.33	78.69
Best main-table baseline	79.77	74.19	73.76

I.5 Latent Domain Discovery without Domain Names

HyperIDAC does not require instance-level domain annotations. Instead, domain prompts provide only a weak, language-grounded signal to estimate a soft posterior for conditioning, rather than hard domain labels. Importantly, the required domain information is coarse and practically mild: it can be expressed by a small set of high-level descriptors and need not exactly match a dataset-defined taxonomy.

To further examine the scenario where explicit domain names or prompt descriptors are unavailable, we replace domain prompts with unsupervised latent domain discovery based on K-means clustering. Here, domains are treated as latent clusters rather than predefined semantic categories, and the resulting cluster assignments are used as the domain-conditioning signal. We evaluate two unsupervised variants: (i) K-means with fixed K, where clustering itself is unsupervised but the number of clusters is specified in advance (i.e., the number of domains), and (ii) K-means with elbow-based K selection, where K is determined automatically in an unsupervised manner. Notably, we adopt K-means here as a simple and classical clustering method rather than a highly engineered alternative.

As shown in Tab. 21, both K-means variants remain competitive on DomainNet and Office-Home across the Original, LTD, and DCI settings. Although they perform slightly below HyperIDAC with domain prompts, they still outperform the best-performing baseline in most settings. These results indicate that HyperIDAC does not critically rely on predefined domain names or exact prompt taxonomy, and that latent clusters provide an effective substitute when domain descriptors are unavailable.

J Algorithm

The detailed training algorithm is provided on the next page.

Algorithm 1 HyperIDAC: Hypernetwork-based Imbalanced Domain Adaptive Classifier

Input: Unlabeled multi-domain dataset $\mathbf{D} = \{x^i\}_{i=1}^N$ with domains $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_K\}$ and classes $\mathcal{C} = \{1, \ldots, C\}$, frozen CLIP feature extractor $\mathcal{F}$, domain text embeddings $\mathcal{T}$, class embeddings $\mathbf{W}$, confidence thresholds τ, τ_c, CLIP trust factor $w_{\text{CLIP}}(t)$
Output: Trained Domain embedding table $\mathbf{E}$, Prior network $\mathcal{P}$, Hypernetwork $\mathcal{H}$, Attribute Adapter A_ξ

Initialize: Historical buffer $\{(y^i, p^i)\}_{i=1}^N$ with $p^i = 0$ for all i
Initialize: Domain embedding table $\mathbf{E}$, Prior network $\mathcal{P}$, Hypernetwork $\mathcal{H}$, Attribute Adapter A_ξ
for epoch $= 1$ to N_{epoch} **do**
  Update τ and $w_{\text{CLIP}}(t)$
  for batch $\mathcal{B} \in \mathbf{D}$ **do**
    for $x \in \mathcal{B}$ **do** ▷ For simplicity, we omit the superscript i in this part
      # Domain-Aware Prior Network
      Compute Zero-shot domain probability distribution $p(d_k|x)$ ▷ via Eq. (1)
      $\mathbf{e}_x \leftarrow \sum_{k=1}^K p(d_k|x)\mathbf{E}_k$ ▷ Domain-weighted embedding
      $\beta \leftarrow \mathcal{P}(\mathcal{F}(x) \oplus \mathbf{e}_x)$ ▷ Generate prior vector
      # Hypernetwork-based Dynamic Classifier
      $\theta \leftarrow \mathcal{H}(\beta)$
      Reshape θ to (W_β, b_β)
      $\mathbf{z}_{\text{base}} = W_\beta \mathcal{F}(x) + b_\beta$ ▷ Data-driven logits
      $\tilde{\mathbf{W}} \leftarrow \text{AttributeAdapter}A_\xi(\mathbf{W}, \beta)$ ▷ Domain-conditioned embeddings via Eq. (5)
      Compute $\mathbf{z}_{\text{cos}}[c]$ for $c \in \{1, \ldots, C\}$ ▷ via Eq. (6)
      $\mathbf{z}_{\text{cos}} \leftarrow \text{Stack}(\{\mathbf{z}_{\text{cos}}[c]\}_{c=1}^C)$ ▷ Semantic logits
      $\lambda_{\text{cos}} = \text{std}(\mathbf{z}_{\text{base}})/\text{std}(\mathbf{z}_{\text{cos}})$
      $\mathbf{z} = \mathbf{z}_{\text{base}} + \lambda_{\text{cos}}\mathbf{z}_{\text{cos}}$ ▷ Combined prediction
    end for
    # Multi-Level Pseudo-Labeling
    $\mathcal{I}_T = \{\}, \mathcal{I}_C = \{\}, \mathcal{I}_L = \{\}$ ▷ Initialize sample sets
    for $x^i \in \mathcal{B}$ **do** ▷ Now we use superscript i to index samples
      Generate weak/strong augmentations $(\tilde{x}_w^i, \tilde{x}_s^i)$
      $(\tilde{y}_w^i, \tilde{p}_w^i) = \mathcal{G}\left(\mathcal{F}(\tilde{x}_w^i), \theta^{i,t}, \tilde{\mathbf{W}}^i\right)$, ▷ Teacher prediction and confidence
      if $\tilde{p}_w^i \geq \tau$ **and** $\tilde{p}_w^i > p^i$ **then**
        Update $(y^i, p^i) \leftarrow (\tilde{y}_w^i, \tilde{p}_w^i)$
      end if
      if $\tilde{p}_w^i \geq \tau$ **then**
        $\mathcal{I}_T \leftarrow \mathcal{I}_T \cup \{i\}$ ▷ High-confidence
      else
        Compute $p_{\text{CLIP}}(i)$ and y_{CLIP}^i via Eq. (11).
        if $p_{\text{CLIP}}(i) \geq \tau_c$ **then**
          $\mathcal{I}_C \leftarrow \mathcal{I}_C \cup \{i\}$ ▷ CLIP-confident
        else
          $\mathcal{I}_L \leftarrow \mathcal{I}_L \cup \{i\}$ ▷ Low-confidence
        end if
      end if
    end for
    # Training Losses
    for $i \in \mathcal{I}_T$ **do**
      $w_i \leftarrow \max\left(0.1, \left(\frac{p^i - \tau}{1-\tau}\right)^2\right), \text{label}^i \leftarrow y^i$ ▷ pseudo label from history buffer, Eq. (9)
    end for
    for $i \in \mathcal{I}_C$ **do**
      $w_i \leftarrow w_{\text{CLIP}}(t)\, p_{\text{CLIP}}(i), \text{label}^i \leftarrow y_{\text{CLIP}}^i$ ▷ pseudo label from CLIP, Eq. (11)
    end for
    $\mathcal{L}_{\text{CE}} \leftarrow$ weighted cross-entropy on $\mathcal{I}_T \cup \mathcal{I}_C$ using student logits $\mathbf{z}_s^i$ on $\tilde{x}_s^i$ and label^i
    $\mathcal{L}_{\text{KL}} \leftarrow$ KL consistency on $\mathcal{I}_L$ using teacher logits $\mathbf{z}_t^i$ on $\tilde{x}_w^i$ and student logits $\mathbf{z}_s^i$ on $\tilde{x}_s^i$
    $\mathcal{L} \leftarrow \mathcal{L}_{\text{CE}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}$
    Update trainable modules $\mathbf{E}$, $\mathcal{P}$, $\mathcal{H}$, A_ξ via back propagation
    $\Theta_t \leftarrow \alpha_{\text{EMA}}\Theta_t + (1 - \alpha_{\text{EMA}})\Theta_s$ ▷ EMA update for teacher
  end for
end for

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the scope of the work as Domain-Conditioned Class Imbalance in unlabeled multi-domain visual recognition and summarize the proposed HyperIDAC framework. The empirical claims are supported by experiments on four multi-domain benchmarks under Original, LTD, and DCI regimes, as reported in Sec. 4 and Appendix E.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations discussion clarifying that the study focuses on visual multi-domain classification benchmarks under controlled imbalance regimes. It explains that the Original, LTD, and DCI settings are designed to isolate the interaction between domain heterogeneity and class imbalance, but are built on benchmark-style domains that are given or discoverable as groups. The discussion further notes that continuously varying domains or highly ambiguous domain boundaries may require finer-grained domain discovery or continual adaptation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper states one formal result, Proposition 1 in Appendix B, with its assumption (strictly positive domain-specific class priors) and a complete proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the datasets, imbalance construction, baselines, backbone, architecture choices, thresholds, and training protocol needed to reproduce the main results. Additional dataset construction details and hyperparameters are provided in Appendix D, and the full algorithm in Algorithm 1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The experiments are based on public benchmark datasets and the paper provides detailed implementation and data-construction information, but the code is not released with the paper. We plan to release the code and reproduction scripts in a public repository.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4 and Appendix D specify the datasets, Original/LTD/DCI settings, baselines, CLIP backbone, training schedule, architecture details, confidence thresholds, EMA coefficient, KL temperature, and regularization weight.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Each main experiment is repeated three times and mean values are reported, but the current version does not report standard deviations, confidence intervals, or statistical significance tests for the main tables. We omit these due to space constraints and report the repetition protocol instead.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix E.6 reports wall-clock training time on a single RTX 3090 GPU for the DomainNet-DCI setting and inference time on DomainNet, and the implementation details specify that all methods are trained under a unified 30-epoch protocol. We also provide the hardware and approximate compute requirements needed to reproduce the reported experiments. However, the current version does not provide complete compute information for every individual experiment, total project-level compute, CPU/RAM/storage details, or preliminary and failed experiments. We therefore answer No conservatively.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research uses public benchmark datasets and does not involve human-subject experiments, private personal data, or deceptive data collection.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper discusses positive impacts from highlighting DCI as an overlooked issue in multi-domain dataset construction and encouraging DCI-aware dataset construction, fine-grained robustness evaluation, and analysis beyond average accuracy. It also discusses potential negative societal impacts and states that direct negative impacts are expected to be limited, since the work is a benchmark study built on public visual datasets and does not introduce new personal data or generate content.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk pretrained language model, image generator, or newly scraped dataset. The experiments use existing public benchmarks, and the proposed classifier is not designed for generative or dual-use content creation.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original sources for all datasets, pretrained models, and baseline methods used in the paper, including Office-Home, DomainNet, PACS, VLCS, CIFAR-100, ImageNet-LT, CLIP, and the compared baselines. Licenses are not fully enumerated.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce a new dataset or pretrained model asset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing, participant studies, or newly collected human-subject data.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human-subject research, crowdsourcing, or interaction with study participants, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method does not use an LLM as an important, original, or non-standard component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.