Domain Adaptive Hashing Retrieval via VLM Assisted Pseudo-Labeling and Dual Space Adaptation

Jingyao Li[1,2], **Zhanshan Li**[1,2], **Shuai Lü**[1,2*]
[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun 130012, China
Email addresses: jingyao@jlu.edu.cn, lizs@jlu.edu.cn, lus@jlu.edu.cn

Abstract

Unsupervised domain adaptive hashing has emerged as a promising approach for efficient and memory-friendly cross-domain retrieval. It leverages the model learned on labeled source domains to generate compact binary codes for unlabeled target domain samples, ensuring that semantically similar samples are mapped to nearby points in the Hamming space. Existing methods typically apply domain adaptation techniques to the feature space or the Hamming space, especially pseudo-labeling and feature alignment. However, the inherent noise of pseudo-labels and the insufficient exploration of complementary knowledge across spaces hinder the ability of the adapted model. To address these challenges, we propose a Vision-language model assisted Pseudo-labeling and Dual Space adaptation (VPDS) method. Motivated by the strong zero-shot generalization capabilities of pre-trained vision-language models (VLMs), VPDS leverages VLMs to calibrate pseudo-labels, thereby mitigating pseudo-label bias. Furthermore, to simultaneously utilize the semantic richness of high-dimensional feature space and preserve discriminative efficiency of low-dimensional Hamming space, we introduce a dual space adaptation approach that performs independent alignment within each space. Extensive experiments on three benchmark datasets demonstrate that VPDS consistently outperforms existing methods in both cross-domain and single-domain retrieval tasks, highlighting its effectiveness and superiority.

1 Introduction

Hashing retrieval aims to map samples into compact low-dimensional binary codes, enabling fast and memory-efficient retrieval via lightweight bit-wise operations in the Hamming space [12, 52, 53, 57, 65]. Existing hashing methods are typically categorized into supervised and unsupervised approaches. Supervised hashing methods [38, 59, 63] benefit from rich label information and enable the learning of highly discriminative representations, but they require large-scale labeled training data and assume that training and query data are drawn from the same distribution, which are difficult to be satisfied in real-world applications. Unsupervised hashing methods [24, 26, 46] eliminate the dependency on labeled training data, but their performance is limited by the absence of reliable supervision and they still struggle with the distribution shift between training and query data.

Unsupervised domain adaptation (UDA) [2, 5, 10, 34, 44, 45, 60] is developed to transfer knowledge from the labeled source domain to the unlabeled target domain, releasing the need for distribution consistency across domains. This makes UDA a compelling solution for addressing the challenges in both supervised and unsupervised hashing. As a result, domain adaptive hashing retrieval [3, 14,

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

49, 50, 51, 54] has gained increasing attention. The goal of domain adaptive hashing methods is to reduce the distribution discrepancy between the training and query data, while maintaining semantic structure within the Hamming space to ensure effective cross-domain retrieval.

To reduce distribution discrepancy across domains, most existing methods perform distribution alignment either in the high-dimensional feature space or in the low-dimensional Hamming space. Common strategies include minimizing the distribution difference measured by statistical metrics [14, 48] or adversarial training with a discriminator [29, 49]. However, features in the feature space possess richer semantic expressiveness than hash codes in the Hamming space, which are constrained by their dimensionality. When alignment is performed solely in the feature space, the aligned feature structure can be distorted when mapped into the Hamming space due to inevitable information loss. Conversely, when alignment is performed solely in the Hamming space, due to limited representational capacity, semantically similar samples can be misaligned even if global distributions are well aligned. Recent works have attempted to jointly align the feature and Hamming spaces, for example, via concatenated feature-hash representations [49] or structural consistency constraints [3]. Despite showing some effectiveness, such methods enforce a tight coupling between the feature and Hamming spaces. Given the substantial disparity in their dimensionality and semantic capacity, we argue that this tight coupling restricts alignment flexibility. Instead, we propose a decoupled alignment strategy that performs domain alignment independently in the feature and Hamming spaces, without imposing strict cross-space consistency. This relaxed design enables more flexible integration of complementary information from each space, leading to improved adaptation performance.

To maintain semantic structure in the Hamming space, pseudo-labeling is one of the most commonly used approaches. In the absence of ground-truth labels in the target domain, the generated pseudo-labels inevitably contain noise, which can misguide the adaptation process. To alleviate this issue, prior works have introduced various strategies, such as confidence-based thresholding [11, 43, 61], auxiliary modules to reduce the bias to the source domain [28, 35, 50, 58], and uncertainty-based weighting schemes to modulate the influence of noisy pseudo-labels [49, 64]. While these approaches alleviate pseudo-label noise, they remain susceptible to the bias introduced by the domain gap, especially in the early stages of training. With the recent emergence of pre-trained vision-language models (VLMs) [17, 36] that exhibit strong zero-shot generalization capabilities, we are motivated to leverage VLMs to calibrate pseudo-labels, thereby enhancing the reliability of pseudo-labels and providing more accurate supervision for target domain adaptation.

In this paper, we propose a **V**LM assisted **P**seudo-labeling and **D**ual **S**pace adaptation (**VPDS**) method. Considering that the high-dimensional feature space provides richer semantic information, while the low-dimensional Hamming space is essential for efficient and discriminative retrieval, VPDS performs pseudo-labeling in the feature space and utilizes the generated pseudo-labels to guide the learning of compact and discriminative hashing codes. To enhance the reliability of pseudo-labels, VPDS leverages the strong generalization ability of VLMs to provide calibration information. Moreover, VPDS achieves cross-domain alignment by reducing the domain gap in the feature space and Hamming space separately, facilitating the utilization of space-specific knowledge and enhancing overall adaptation performance.

The contributions of this paper are summarized as follows: (1) We propose VPDS that decouples cross-domain alignment into independent alignments in the feature space and Hamming space, exploring the complementary information across spaces. (2) We propose a VLM assisted pseudo-labeling strategy to enhance pseudo-label reliability. And this strategy is model-agnostic and can be seamlessly incorporated into other unsupervised learning frameworks. (3) Experiments on three standard benchmark datasets demonstrate that VPDS achieves superior performance in both cross-domain and single-domain retrieval scenarios.

2 Related Works

Unsupervised Domain Adaptation. To achieve knowledge transfer from the labeled source domain to the unlabeled target domain, UDA methods aim to learn a domain-invariant feature space. Early approaches rely on statistical metrics, such as Maximum Mean Discrepancy (MMD) [30, 32], CORelation ALignment (CORAL) [41, 42] and Maximum Density Divergence (MDD) [22], to quantify the cross-domain distribution discrepancy, which are then minimized to mitigate the domain gap. With the advent of generative adversarial networks, adversarial UDA methods [7, 31, 55, 56] in-

troduce a domain discriminator to measure the separability of the two domains. Through training the feature extractor and the discriminator in a min-max manner, domain-invariant features can be obtained. More recently, pre-trained VLMs, especially the contrastive language-image pre-training (CLIP) [36] model, have been incorporated into UDA frameworks. These methods typically freeze the backbone parameters of CLIP and design auxiliary adaptation modules [8, 20, 40]. For instance, DAMP [6] introduces a mutual prompting module to learn domain-agnostic prompts and domain-invariant visual embeddings. UniMoS [23] employs a modality separation strategy to enhance vision-language interplay. While UDA has shown impressive progress in tasks such as image classification and semantic segmentation, its application to image retrieval remains relatively underexplored. In this work, we focus on advancing UDA techniques for image retrieval.

Domain Adaptive Hashing. Similar to UDA, domain adaptive hashing also needs to reduce the domain gap. To achieve this, domain adaptive hashing methods typically employ alignment strategies based on statistical metrics [14, 48, 62] or adversarial training [29, 49]. Domain adaptive hashing first encodes data into a high-dimensional feature space and subsequently maps it to a low-dimensional Hamming space. Consequently, alignment is typically performed either in the feature space [14, 29, 48, 54] or in the Hamming space [50, 51]. As the feature space can provide rich semantic information for knowledge transfer, and the Hamming space is crucial for efficient and discriminative retrieval, some recent approaches attempt to exploit both and achieve joint alignment of the two spaces. For example, PEACE [49] concatenates features with their corresponding hash codes, and alignment is performed on the concatenated features. CPH [3] introduces a hypersphere space for feature alignment, then the aligned feature structure is preserved in the Hamming space. However, these joint alignment strategies tightly couple the two spaces, potentially restricting adaptation performance. To this end, we propose to perform alignment separately within each space, allowing the model to exploit the distinct characteristics of each space more effectively.

Pseudo-Labeling Strategy. Pseudo-labeling is a widely adopted strategy for extracting discriminative information from the unlabeled target domain. Due to the absence of ground-truth labels, pseudo-label bias is inevitable. To mitigate this bias, some methods use threshold to select high-confidence pseudo-labels [11, 43, 61]. Some methods design an auxiliary module, such as a teacher network [35, 50, 58] or a target domain-specific model [28], to reduce the impact of source domain information. The topological structure of the feature space provides an alternative strategy for generating pseudo-labels, allowing the model and class prototypes to be iteratively optimized and thus mutually reinforcing [1, 3, 54]. Additionally, some works measure the pseudo-label uncertainty and incorporate it as weighting in the learning objective [49, 64]. Despite these efforts, most methods rely solely on the adapted model, which is affected by the domain gap especially in the early stages of training. To thie end, we propose leveraging pre-trained VLMs to provide external calibration for enhancing pseudo-label reliability. Our method mitigates early-stage pseudo-label bias and prevents its accumulation throughout training, resulting in more stable and effective adaptation.

3 Method

The source domain containing N_s labeled samples is denoted as $\mathcal{D}^s = \{(\boldsymbol{x}_i^s, y_i^s)\}_{i=1}^{N_s}$, and the target domain containing N_t unlabeled samples is denoted as $\mathcal{D}^t = \{\boldsymbol{x}_i^t\}_{i=1}^{N_t}$. Assuming that $\mathcal{D}^s$ and $\mathcal{D}^t$ follow different data distributions but share the same label space $\mathcal{Y}$. The objective of domain adaptive retrieval is to learn a model on $\mathcal{D}^s$ and $\mathcal{D}^t$, which can map a sample $\boldsymbol{x} \in \mathcal{D}^s \cup \mathcal{D}^t$ to a binary code $\boldsymbol{b} \in \{-1, 1\}^L$, where L represents the code length. The learned model should ensure that semantically similar samples are mapped to compact binary codes in the Hamming space, enabling effective retrieval in both cross-domain and single-domain retrieval tasks. In cross-domain retrieval, query and retrieval samples are from $\mathcal{D}^t$ and $\mathcal{D}^s$, respectively. In single-domain retrieval, both query and retrieval samples are from $\mathcal{D}^t$. The framework of our method is shown in Figure 1.

3.1 Dual Space Adaptation

Feature space and Hamming space are two commonly used spaces where cross-domain alignment is performed. The high-dimensional feature space provides rich semantic capacity, making it effective for capturing transferable knowledge across domains. In contrast, the low-dimensional Hamming space is crucial for efficient and discriminative retrieval via compact binary codes. Although existing approaches have attempted to leverage the complementary strengths of both spaces, they often

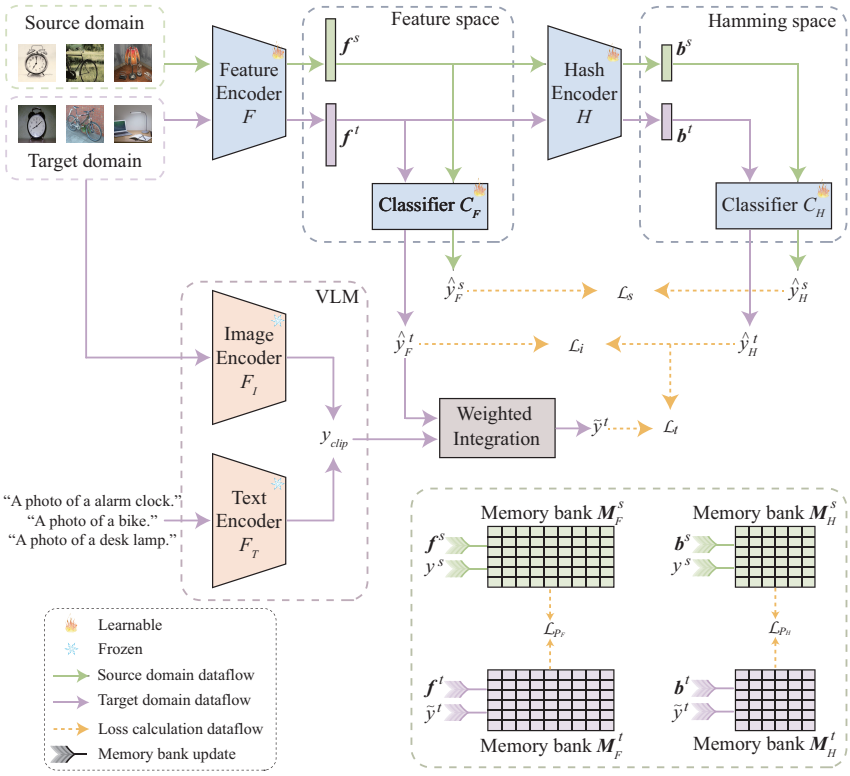


Figure 1: Overview of the VPDS framework. Pseudo-labels are generated in the feature space and used to guide the learning of discriminative hash codes. A frozen pre-trained VLM is utilized to provide auxiliary corrective information for improving pseudo-label reliability. Classifiers in the feature and Hamming spaces preserve the discriminability of learned representations, while memory bank-based domain alignment within each space facilitates capturing transferable knowledge.

impose strict structural consistency constraints between them. However, the dimensionality reduction from feature space to Hamming space inevitably induces information loss, which can distort semantic relationships and alter the space structure. Therefore, we argue that enforcing strict structural consistency between the two spaces suppress the leverage of advantages unique to each space, thereby constraining the overall capacity for effective knowledge transfer. Based on this idea, we propose a dual space adaptation strategy that performs domain alignment independently within each space, while relaxing mutual constraints across spaces to preserve their respective strengths.

Both transferability and discriminability are essential for effective model adaptation. The discriminability usually comes from two sources, including the ground-truth labels in the source domain and the self-supervised information in the target domain. Therefore, the model is initially trained on the labeled source domain and subsequently used to generate pseudo-labels for the unlabeled target domain, thereby enabling self-supervised learning within the target domain. Since our dual space adaptation strategy performs cross-domain alignment in two spaces separately, we introduce two classifiers: C_F in the feature space and C_H in the Hamming space, to ensure that sufficient discriminative information is retained in each space. These two classifiers are learned on source domain features $\boldsymbol{f}^s$ and hash codes $\boldsymbol{b}^s$, respectively, and the supervised loss function is formulated as:

$$\mathcal{L}_s = -\frac{1}{N_s}\sum_{i=1}^{N_s} \mathrm{CE}(C_F(\boldsymbol{f}_i^s), y_i^s) + \mathrm{CE}(C_H(\boldsymbol{b}_i^s), y_i^s), \tag{1}$$

where y_i^s is the ground-truth label of $\boldsymbol{x}_i^s$, $\mathrm{CE}(\cdot)$ represents the cross-entropy loss, $C_F(\cdot)$ and $C_H(\cdot)$ indicate the prediction results.

For the unlabeled target domain, we leverage the feature space to generate pseudo-labels, as it contains richer semantic information, providing a sufficient knowledge foundation for capturing discriminative information. The generated pseudo-labels are then used to supervise the learning of

discriminative hash codes in the Hamming space. Although this process introduces a connection between the two spaces, the information in the Hamming space does not inversely affect the feature space. Thus the mutual constraints between the two spaces of prior works are relaxed to a unidirectional constraint. Formally, for the target domain feature $\boldsymbol{f}^t$, the pseudo-label can be obtained by the adapted model, namely, $\hat{y}^t = C_F(\boldsymbol{f}^t)$. This paper proposes a pseudo-labeling strategy to enhance $\hat{y}^t$ via VLM, which will be introduced in Section 3.2, and the enhanced pseudo-label is denoted as $\tilde{y}^t$. Then, $\tilde{y}^t$ is used for the self-supervised learning in the Hamming space:

$$\mathcal{L}_t = -\frac{1}{N_t}\sum_{i=1}^{N_t} \mathrm{CE}(C_H(\boldsymbol{b}_i^t), \tilde{y}_i^t). \tag{2}$$

In addition, we use the information maximization technique [6, 13, 25] in each space to further mine semantic structure information of the unlabeled target domain. In each space, the information maximization objective can be formalized as:

$$\mathcal{L}_i = \frac{1}{N_t}\sum_{i=1}^{N_t}\sum_{c=1}^{|\mathcal{Y}|} \hat{p}_i^c \log \hat{p}_i^c - \sum_{c=1}^{|\mathcal{Y}|} \overline{p}^c \log \overline{p}^c, \tag{3}$$

where $\hat{p}_i^c$ denotes the confidence of C_F predicting the feature $\boldsymbol{f}_i^t$ as the c-th class, or C_H predicting the hash code $\boldsymbol{b}_i^t$ as the c-th class. $\overline{p}^c = \frac{1}{N_t}\sum_{i=1}^{N_t} \hat{p}_i^c$ represents the average class confidence over all target samples. The first term represents the entropy of the predictions, encouraging confident and discriminative classification, while the second term promotes prediction diversity across classes, mitigating the risk of class collapse.

To enhance transferability, we propose a prototype-based global alignment strategy. Unlike prior works that perform class-level alignment using class prototypes, we treat prototypes as the most representative samples and align domain-level distributions. Given that the model is continuously updated during training, the constructed prototypes may deviate from the true class centers. Therefore, rather than enforcing strict class-level alignment, we focus on global distribution alignment, while the class-level alignment can be implemented via classifiers C_F and C_H. In the feature space, we construct two memory banks to store class prototypes of the source and target domains, denoted as $\boldsymbol{M}_F^s \in \mathcal{R}^{|\mathcal{Y}|\times d_F}$ and $\boldsymbol{M}_F^t \in \mathcal{R}^{|\mathcal{Y}|\times d_F}$, respectively, where d_F is the dimension of features in the feature space. To improve training stability and robustness, we adopt a momentum-based strategy to incrementally update the memory banks at each training iteration:

$$\boldsymbol{M}_{F_{e+1}}^s = \alpha \boldsymbol{M}_{F_e}^s + (1-\alpha)\boldsymbol{\mathcal{P}}_{F_{e+1}}^s, \quad \boldsymbol{M}_{F_{e+1}}^t = \alpha \boldsymbol{M}_{F_e}^t + (1-\alpha)\boldsymbol{\mathcal{P}}_{F_{e+1}}^t, \tag{4}$$

where e denotes the iteration number, α is a momentum coefficient controlling the update rate. $\boldsymbol{\mathcal{P}}_F^s$ and $\boldsymbol{\mathcal{P}}_F^t$ represent the class prototypes of the source and target domains, respectively, computed by the features in the feature space. These prototypes are formulated as:

$$\boldsymbol{\mathcal{P}}_{F_c}^s = \frac{1}{|B_c^s|}\sum_{x_i^s \in B_c^s} \boldsymbol{f}_i^s, \quad \boldsymbol{\mathcal{P}}_{F_c}^t = \frac{1}{|B_c^t|}\sum_{x_i^t \in B_c^t} \boldsymbol{f}_i^t, \tag{5}$$

where B_c^s denotes the set of source domain samples with ground-truth label c, and B_c^t denotes the set of target domain samples with pseudo-label c. Then the domain-level alignment in the feature space is achieved by aligning each source domain prototype with each target domain prototype:

$$\mathcal{L}_{P_F} = -\frac{1}{|\mathcal{Y}|}\sum \boldsymbol{M}_F^s \times {\boldsymbol{M}_F^t}^{\mathrm{T}}. \tag{6}$$

Similar to the feature space, we construct two memory banks in the Hamming space, denoted as $\boldsymbol{M}_H^s \in \mathcal{R}^{|\mathcal{Y}|\times d_H}$ and $\boldsymbol{M}_H^t \in \mathcal{R}^{|\mathcal{Y}|\times d_H}$, where d_H is the length of hash codes in the Hamming space. The update strategy for $\boldsymbol{M}_H^s$ and $\boldsymbol{M}_H^t$ follows the same momentum-based rule as described in Eq. (4). And the class prototypes $\boldsymbol{\mathcal{P}}_H^s$ and $\boldsymbol{\mathcal{P}}_H^t$ in the Hamming space are computed by the hash codes $\boldsymbol{b}^s$ and $\boldsymbol{b}^t$, following the same strategy as in Eq. (5). Then the domain-level alignment in the Hamming space is achieved via:

$$\mathcal{L}_{P_H} = -\frac{1}{|\mathcal{Y}|}\sum \boldsymbol{M}_H^s \times {\boldsymbol{M}_H^t}^{\mathrm{T}}. \tag{7}$$

As a result, the dual space alignment approach can be implemented by minimizing:

$$\mathcal{L}_P = \mathcal{L}_{P_F} + \mathcal{L}_{P_H}. \tag{8}$$

3.2 VLM Assisted Pseudo-Labeling

In the previous section, pseudo-labels are utilized in Eqs. (2) and (5), thus their reliability plays a critical role in the adaptation process. Most existing methods rely solely on the adapted model to generate pseudo-labels, which can lead to pseudo-label bias due to the domain gap. To address this challenge, we are motivated by the strong generalization capabilities of the pre-trained VLMs, and propose to incorporate VLMs as auxiliary guidance for pseudo-label generation, particularly in the early stages of training when the model has not yet adapted to the target domain. To be specific, the final pseudo-labels are the combination of predictions from the VLM and from the adapted model, where the relative contributions of two terms are dynamically adjusted throughout training. As training progresses, the model progressively learns more target domain-specific knowledge, resulting in increasingly reliable predictions. Accordingly, we gradually increase the contribution of the adapted model, while gradually reducing the reliance on the VLM.

In this paper, we use CLIP as the VLM framework. To obtain predictions based on CLIP, the textual label descriptions are formalized as $\mathcal{T}_c$ = "*a photo of a* $[CLASS_c]$". Based on the similarity of image features and textual features, which are encoded by the image encoder F_I and text encoder F_T of CLIP, the prediction result can be obtained through:

$$y_{clip} = \text{Cosine}(F_T(\mathcal{T}_c), F_I(\boldsymbol{x}^t)). \tag{9}$$

Based on $\hat{y}^t$ and y_{clip}, we construct the final pseudo-label by:

$$\tilde{y}^t = \arg\max_c \beta \hat{y}^t + (1-\beta) y_{clip}, \tag{10}$$

here, β is implemented as a linearly increasing parameter, namely $\beta = e/E$, where e denotes the current training iteration and E is the total number of training iterations. Since CLIP is not optimized in our method, y_{clip} can be computed once in the first epoch and stored for use in subsequent epochs. Therefore, this pseudo-labeling strategy incurs minimal computational overhead, and can be seamlessly incorporated with other unsupervised learning frameworks.

3.3 Training and Inference

Finally, the overall training objective of our method is formalized as:

$$\mathcal{L} = \mathcal{L}_s + \eta \mathcal{L}_t + \gamma \mathcal{L}_i + \mathcal{L}_P, \tag{11}$$

where η and γ are the trade-off parameters. Since the $sign(\cdot)$ function used to generate binary hash codes is non-differentiable and thus unsuitable for gradient-based optimization [49], we adopt its smooth approximation $tanh(\cdot)$ during training. Specifically, we use $\boldsymbol{b} = tanh(H(F(x)))$ for training, and use $\boldsymbol{b} = sign(H(F(x)))$ for inference, where $F(\cdot)$ and $H(\cdot)$ denote the feature encoder and hash encoder, respectively.

4 Experiment

4.1 Setup

Datasets. We evaluate our method on three benchmark datasets. **Office-Home** [48] consists of 4 domains from 65 classes. Consistent with prior works [3, 50, 51], we construct 6 tasks based on this dataset. **Office-31** [37] contains 3 domains, with each domain containing 31 classes. **Digits** consists of 2 domains, MNIST [21] and USPS [16], and each domain contains 10 handwritten digits.

Baselines. We compare our method with several state-of-the art methods, including unsupervised hashing methods (i.e., ITQ [9], DSH [19], SGH [18], OCH [27] and GraphBit [53]) and domain adaptive hashing methods (i.e., CTH-g [62], PWCF [15], DAPH [14], DHLing [54], PEACE [49], DANCE [50], IDEA [51], CPH [3] and COUPLE [33]).

Implementation Details. For both cross-domain and single-domain retrieval tasks, 10% of the target domain samples are randomly selected as the query set. The remaining 90% along with all source domain samples constitute the training set. The retrieval set is constructed from the training set, depending on whether the task is cross-domain or single-domain retrieval. The feature encoder F is implemented using the VGG [39] backbone. For the VLM, we adopt the pre-trained CLIP [36] with ViT-B/16 [4], and keep its parameters frozen during training. The model is optimized using

Table 1: MAP (%) of cross-domain retrieval tasks with 64-bit hash codes on Office-Home and Office-31. The best and second-best results are highlighted in bold and underlined.

	Office-Home						Office-31						
Task	A→R	R→A	C→R	R→C	P→R	R→P	A→D	A→W	D→A	D→W	W→A	W→D	Avg
ITQ	25.88	25.37	14.83	14.92	26.81	28.19	29.55	28.53	26.83	58.89	25.09	58.00	30.24
DSH	9.69	9.67	5.47	5.28	8.49	8.26	16.66	15.09	16.33	41.07	13.58	39.24	15.74
SGH	22.93	22.53	13.62	13.51	24.51	25.73	24.98	22.47	22.17	56.36	20.52	53.94	26.94
OCH	18.09	17.54	10.27	10.05	18.65	20.15	24.86	22.49	22.45	53.64	20.79	51.03	24.17
GraphBit	18.18	16.87	11.51	10.81	18.91	21.32	24.48	23.12	22.09	53.82	21.34	51.43	24.49
GTH-g	16.95	17.54	8.46	11.88	17.82	18.57	30.85	18.44	21.99	48.48	20.02	50.23	23.56
PWCF	34.57	28.95	24.22	18.42	34.03	34.44	39.78	34.86	35.12	72.91	35.01	67.94	38.35
DAPH	21.19	22.28	13.25	12.26	26.61	24.26	29.60	22.94	25.48	60.67	24.31	45.42	31.85
DHLing	48.47	30.81	38.68	45.24	25.15	43.30	41.96	45.10	75.23	42.89	41.74	79.91	46.54
PEACE	45.97	42.68	38.72	28.36	53.04	54.39	46.69	48.89	46.91	83.18	46.95	78.82	51.22
DANCE	44.53	43.54	39.03	28.87	53.73	55.14	44.78	47.66	46.68	84.75	48.61	78.39	51.31
IDEA	51.19	49.64	45.71	32.77	59.18	61.84	48.70	54.43	53.53	88.69	53.71	84.97	57.03
COUPLE	54.14	54.35	49.24	41.39	63.94	64.29	50.27	59.32	56.04	88.90	56.35	85.26	60.29
CPH	71.18	63.28	58.65	**42.84**	71.27	74.77	68.37	60.61	52.84	95.88	60.14	99.90	68.31
VPDS	**72.54**	**63.36**	**63.58**	41.95	**75.17**	**81.80**	**87.22**	**91.40**	**65.88**	**98.91**	**64.88**	**100**	**75.56**

Table 2: MAP (%) of cross-domain retrieval tasks with various bits on Digits. The best and second-best results are highlighted in bold and underlined.

	MNIST→USPS						USPS→MNIST						
Bit	16	32	48	64	96	128	16	32	48	64	96	128	Avg
ITQ	13.05	15.57	18.54	20.12	23.12	23.89	13.69	17.51	20.40	20.30	22.79	24.59	19.46
DSH	20.60	22.21	23.68	24.28	25.73	26.50	19.54	21.22	22.89	23.79	25.91	26.46	23.57
SGH	14.24	16.69	18.72	19.70	21.00	21.95	13.26	17.71	18.22	19.01	21.69	22.09	18.69
OCH	13.73	17.22	19.59	20.18	20.66	23.34	15.51	17.75	18.97	21.50	21.27	23.68	19.45
GTH-g	20.45	17.64	16.60	17.25	17.26	17.06	15.17	14.07	15.02	15.01	14.80	17.34	16.47
PWCF	47.47	51.99	51.44	51.75	50.89	59.35	47.14	50.86	52.06	52.18	57.14	58.96	52.60
DAPH	25.13	27.10	26.10	28.51	30.53	30.70	26.60	26.43	27.27	27.99	30.19	31.40	28.16
DHLing	49.24	54.90	56.30	58.28	58.80	59.14	50.14	51.35	53.67	58.65	58.42	59.17	55.67
PEACE	52.87	59.72	60.69	62.84	65.13	68.16	53.97	54.82	58.69	60.91	62.65	65.70	60.51
DANCE	53.18	57.98	61.23	63.15	65.92	68.87	54.31	55.64	57.26	61.49	63.43	66.23	60.72
IDEA	58.89	64.48	65.72	67.48	70.24	74.34	60.99	61.47	65.45	67.97	69.72	72.31	66.59
CPH*	64.55	63.33	65.94	71.04	68.06	71.85	54.36	61.64	64.17	65.59	68.47	70.70	65.81
COUPLE	60.56	66.05	66.23	67.98	73.02	75.12	63.28	64.94	67.44	70.19	72.87	74.62	68.53
VPDS	**86.21**	**90.53**	**92.98**	**90.86**	**93.13**	**90.14**	**80.90**	**83.44**	**67.50**	**89.50**	**76.78**	**87.97**	**85.83**

* indicates that the results for CPH are obtained by running its original code, while other results are obtained directly from their original papers.

SGD with momentum 0.9 and weight decay 1e-5. We set the initial learning rate to 1e-3 and use a batch size of 72. Training epoch is set to 20 for Digits and 100 for others. The hyperparameters are fixed as $\alpha = 0.9$, $\eta = 0.2$, $\gamma = 0.2$ across all datasets. All experiments are conducted on NVIDIA A30 GPU. We evaluate the retrieval performance using 4 standard metrics: mean average precision (MAP), precision-recall curve, Top-N precision curve and Top-N recall curve.

4.2 Comparison Results

Cross-Domain Retrieval. The MAP results of cross-domain retrieval tasks on three datasets are presented in Tables 1–2. As shown in Table 1, the proposed VPDS method consistently achieves the best performance on most tasks, with a notable average improvement of 7.25% over the second-best method CPH. Among the datasets, Office-Home presents a greater challenge due to its larger number of categories. The strong performance of VPDS on this dataset highlights its effectiveness in learning discriminative hash codes even in more complex category spaces. Additionally, tasks such as D→A and W→A from Office-31 require transferring knowledge from a source domain with significantly fewer samples to a target domain with many more. On these tasks, VPDS yields

Table 3: MAP (%) of single-domain retrieval tasks with various bits on Office-Home, Office-31 and Digits. The best and second-best results are highlighted in bold and underlined.

Task	P→R				A→D				MNIST→USPS			
Bit	16	32	64	128	16	32	64	128	16	32	64	128
ITQ	20.07	29.64	33.15	34.81	40.83	49.27	56.16	59.41	13.39	22.58	39.67	40.16
DSH	6.10	11.44	16.61	14.45	22.45	33.38	40.09	46.31	41.42	45.30	47.85	50.76
SGH	18.97	26.18	32.61	34.97	38.67	45.59	53.57	57.37	15.60	30.78	35.55	41.78
OCH	13.45	21.14	25.34	28.02	33.30	41.65	50.78	53.74	24.23	32.90	36.34	44.36
GraphBit	15.42	21.80	24.89	28.97	33.21	41.17	51.46	53.48	24.96	32.54	37.54	44.82
GTH-g	15.05	21.20	27.67	28.40	37.11	45.69	50.22	55.81	45.41	39.72	34.34	34.73
PWCF	24.80	34.03	37.98	39.14	49.94	53.05	59.08	62.35	50.21	49.41	60.06	64.00
DAPH	20.77	29.01	33.35	34.92	46.74	49.43	58.63	60.41	47.53	54.86	60.15	60.39
DHLing	27.81	36.05	40.91	44.07	52.08	56.43	60.17	63.44	51.25	50.48	63.13	67.02
PEACE	28.99	37.93	42.97	47.29	55.43	57.89	61.21	64.14	52.77	56.25	65.27	69.99
DANCE	31.37	37.64	44.13	48.93	54.42	58.02	63.09	67.91	52.65	55.98	66.81	70.47
IDEA	34.88	44.83	49.91	<u>54.40</u>	<u>61.25</u>	<u>62.65</u>	<u>67.06</u>	<u>70.04</u>	60.81	63.32	72.11	<u>76.73</u>
CPH	<u>44.99</u>	<u>49.35</u>	<u>52.45</u>	51.40	60.60	62.11	65.76	68.20	<u>66.76</u>	<u>71.34</u>	<u>72.64</u>	72.46
VPDS	**47.20**	**54.86**	**59.12**	**59.45**	**71.22**	**77.66**	**85.90**	**85.52**	**80.12**	**88.34**	**90.35**	**89.30**

substantial improvements of 9.84% and 4.74%, respectively, further demonstrating its robustness in handling severe domain shifts and data imbalance. Table 2 presents the retrieval performance on the Digits dataset across different hash code lengths. Our method outperforms all compared methods at each code length, achieving an average improvement of 17.30% over the second-best method. These results demonstrate the robustness and effectiveness of VPDS in learning compact and discriminative hash codes. Additional evaluations of cross-domain retrieval performance with varying code lengths on Office-Home and Office-31 are provided in Appendix A.1.

Single-Domain Retrieval. We evaluate single-domain retrieval performance on three randomly selected tasks, each from a different dataset. As shown in Table 3, VPDS outperforms all compared methods by a substantial margin, highlighting its effectiveness and strong generalization capability in single-domain retrieval scenarios.

4.3 Analyses.

Effect of VLM Assisted Pseudo-Labeling. To evaluate the effectiveness of our proposed VLM assisted pseudo-labeling strategy, we construct two variants by replacing it with commonly used strategies: (1) **VPDS w/ Pred**: pseudo-labels are generated from the predictions of the adapted model, using only those with confidence scores above 0.9. (2) **VPDS w/ Prot**: pseudo-labels are assigned based on the nearest class prototypes computed from the source domain. The experimental results are shown in Table 4, we can see that VPDS outperforms the two variants of it, demonstrating the effectiveness of our pseudo-labeling strategy.

Effect of Dual Space Adaptation. To assess the effectiveness of dual space adaptation approach, we design two variants: (1) **VPDS w/ Hamming**: utilizing only the Hamming space for learning, discarding all information from the feature space. (2) **VPDS w/ VLM_Hamm**: generating pseudo-labels based on predictions of the classifier in the Hamming space. As shown in Table 4, both variants result in a noticeable performance drop, highlighting the critical role of transferring knowledge from the semantically rich feature space to guide learning in the Hamming space.

Effect of Prototype-Based Global Alignment. The prototype-based global alignment strategy is evaluated by replacing it with prototype-based class-level alignment and MMD-based global alignment, constructing **VPDS w/ class_prot** and **VPDS w/ global_MMD**. As shown in Table 4, VPDS outperforms the two variants, which demonstrates the superiority of our alignment strategy over conventional class-level and domain-level alignment strategies.

Ablation Study. To assess the contribution of each component in VPDS, we conduct an ablation study by removing each component from VPDS. Details of variants are provided in Appendix A.2. The results, denoted as **VPDS w/o**, are shown in Table 4. We observe performance degradation in all ablation variants, demonstrating the effectiveness and necessity of each component.

Table 4: MAP (%) of VPDS variants on Office-Home for cross-domain retrieval tasks.

Method	A→R	R→A	C→R	R→C	P→R	R→P	Avg
VPDS w/ Pred	73.32	62.95	57.22	40.53	72.91	81.86	64.80
VPDS w/ Prot	68.68	58.73	43.54	35.64	61.56	74.32	57.08
VPDS w/ Hamming	56.90	51.97	13.54	21.33	45.77	50.18	39.95
VPDS w/ VLM_Hamm	67.28	57.68	18.97	29.78	41.29	66.00	46.83
VPDS w/ class_prot	71.57	62.05	61.01	40.19	75.59	82.74	65.53
VPDS w/ global_MMD	72.94	60.89	63.26	41.39	73.97	82.56	65.84
VPDS w/o $\mathcal{L}_{P_F}$	70.23	60.25	60.91	40.15	73.49	80.02	64.18
VPDS w/o $\mathcal{L}_{P_H}$	71.22	62.28	62.72	42.73	73.79	82.52	65.88
VPDS w/o $\mathcal{L}_P$	70.16	63.16	63.16	40.99	75.30	82.12	65.82
VPDS w/o $\mathcal{L}_i$ (feat)	69.28	62.07	55.32	37.58	71.89	76.86	62.17
VPDS w/o $\mathcal{L}_i$ (Hamm)	70.68	61.05	58.79	38.73	73.60	80.92	63.97
VPDS w/o $\mathcal{L}_i$	63.83	56.60	46.95	32.95	67.60	75.24	57.20
VPDS w/o $\mathcal{L}_t$	71.92	60.67	58.79	40.76	74.60	80.23	64.50
VPDS	72.54	63.36	63.58	41.95	75.17	81.80	66.40

Table 5: Scalability (under the unsupervised hashing retrieval setting) and generalization (under the out-of-sample hashing retrieval setting) of VPDS.

Cross-Domain Hashing												
Tasks	A→R	R→A	C→R	R→C	P→R	R→P	A→D	A→W	D→A	D→W	W→A	W→D
Unsupervised	54.18	50.48	32.31	33.14	65.50	68.22	67.63	70.43	67.38	88.81	62.65	88.50
Out-of-Sample	46.68	47.27	45.75	30.95	62.55	64.75	80.58	70.69	62.70	90.15	69.16	98.20

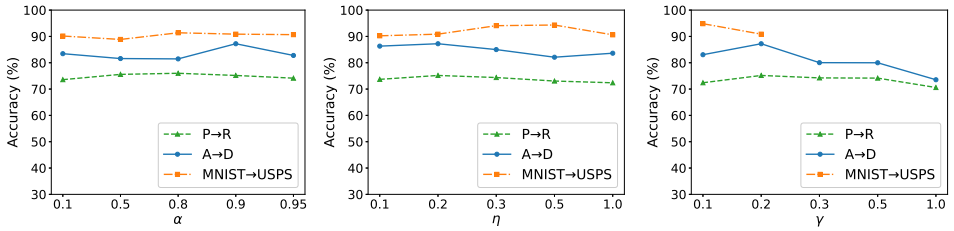
Single-Domain Hashing												
Tasks	P→R				A→D				MNIST→USPS			
Bit	16	32	64	128	16	32	64	128	16	32	64	128
Out-of-Sample	42.22	47.70	52.24	51.68	69.46	78.09	83.75	88.81	87.27	76.22	74.83	88.33

Parameter Sensitivity. We evaluate the sensitivity of our method to all hyperparameters, including α, η and γ. The results in Figure 2 show that the performance remains stable across a wide range of values, indicating that our method is robust to hyperparameter settings.

Precision-Recall Curve. We present the precision-recall curves on four cross-domain retrieval tasks randomly selected from Office-Home and Office-31 in Figure 3. The results show that our method works better than the compared methods. Additionally, the Top-N precision curves and Top-N recall curves are presented in Appendix A.3.

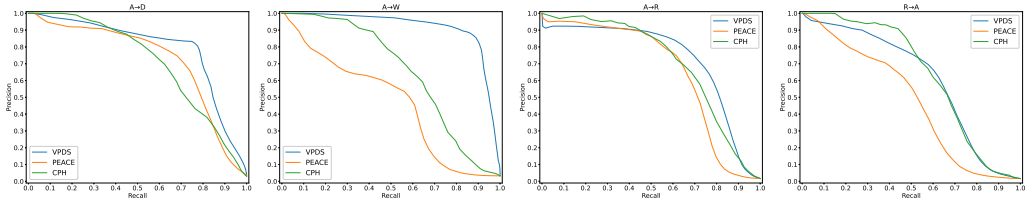
Scalability. Our framework can be readily extended to unsupervised hashing retrieval scenarios. Specifically, when labels for both source and target domain data are unavailable, our VLM assisted pseudo-labeling approach can generate pseudo-labels for samples in both domains. By replacing the ground-truth labels of the source domain in VPDS with these generated pseudo-labels, our method can directly addresses the unsupervised hashing retrieval task. To validate this scalability, we evaluate VPDS under the unsupervised hashing retrieval setting on the Office-Home and Office-31 datasets. As shown in Table 5, our method outperforms all unsupervised baselines and most domain adaptive methods when compared to the results reported in Table 1.

Generalization. To assess the generalization (out-of-sample) ability of our method, we split the source domain data into a 70% training set and a 30% retrieval set, and we divide the target domain data into a 60% training set, a 30% retrieval set and a 10% query set. We evaluate VPDS under this out-of-sample setting, and the results are presented in Table 5. Comparing to the results in Tables 1 and 3, we observe that although the retrieval data is not involved in the training process, our method still outperforms most compared methods that leverage training data as the retrieval set. This validates the strong generalization capability of our approach.



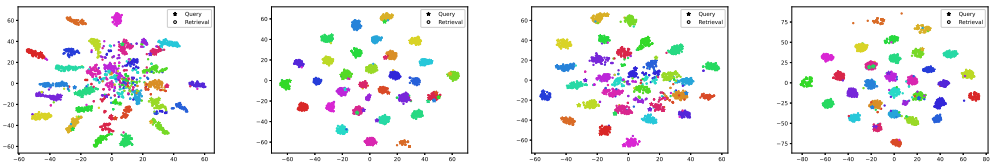
(a) (b) (c)

Figure 2: Parameter sensitivity of VPDS to (a) α, (b) η and (c) γ. When $\gamma \geq 0.3$, gradient vanishing issue emerges in task MNIST→USPS, resulting in an incomplete curve in (c).



(a) (b) (c) (d)

Figure 3: Precision-recall curves on four tasks randomly selected from Office-Home and Office-31.



(a) Feature of PEACE. (b) Hamming of PEACE. (c) Feature of VPDS. (d) Hamming of VPDS.

Figure 4: t-SNE visualization of A→D in feature and Hamming spaces.

Visualization. To illustrate the effect of our dual space adaptation approach, we employ t-SNE [47] to visualize the features in the feature space and hash codes in the Hamming space. We compare the visualization results with PEACE, which enforces a strict relationship between the two spaces via feature concatenation. As shown in Figure 4, our approach produces discriminative hash codes and yields better clustering in the feature space compared to PEACE. These results validate our design of relaxing the constraints across spaces to enable more flexible and effective cross-domain alignment.

5 Conclusion

This paper propose VPDS, a framework that performs independent domain alignment in both the feature and Hamming spaces to exploit their complementary knowledge. To address potential bias from prototype representations, we introduce a prototype-based global alignment strategy. Additionally, we design a VLM assisted pseudo-labeling strategy to improve pseudo-label reliability, which can be seamlessly integrated with other unsupervised learning methods. Extensive experiments on multiple benchmarks validate the effectiveness and generalizability of our method.

Acknowledgments

This work was supported by the Natural Science Research Foundation of Jilin Province of China under Grant No. 20220101106JC, and the National Natural Science Foundation of China under Grant No. 62407010.

References

[1] Sk Miraj Ahmed, Dripta S. Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K. Roy-Chowdhury. 2021. Unsupervised Multi-source Domain Adaptation Without Access to Source Data. In *CVPR*. 10103–10112.

[2] Shuanghao Bai, Min Zhang, Wanqi Zhou, Siteng Huang, Zhirong Luan, Donglin Wang, and Badong Chen. 2024. Prompt-Based Distribution Alignment for Unsupervised Domain Adaptation. In *AAAI*, Vol. 38. 729–737.

[3] Hui Cui, Lihai Zhao, Fengling Li, Lei Zhu, Xiaohui Han, and Jingjing Li. 2024. Effective Comparative Prototype Hashing for Unsupervised Domain Adaptation. In *AAAI*, Vol. 38. 8329–8337.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

[5] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. 2021. Cross-Domain Gradient Discrepancy Minimization for Unsupervised Domain Adaptation. In *CVPR*. 3937–3946.

[6] Zhekai Du, Xinyao Li, Fengling Li, Ke Lu, Lei Zhu, and Jingjing Li. 2024. Domain-Agnostic Mutual Prompting for Unsupervised Domain Adaptation. In *CVPR*. 23375–23384.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. 2016. Domain-Adversarial Training of Neural Networks. *JMLR* 17, 59 (2016), 1–35.

[8] Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. 2025. Domain Adaptation via Prompt Learning. *IEEE TNNLS* 36, 1 (2025), 1160–1170.

[9] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. 2013. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval. *IEEE TPAMI* 35, 12 (2013), 2916–2929.

[10] Qichen He, Siying Xiao, Mao Ye, Xiatian Zhu, Ferrante Neri, and Dongde Hou. 2023. Independent Feature Decomposition and Instance Alignment for Unsupervised Domain Adaptation. In *IJCAI*. 819–827.

[11] Tao He, Yuan-Fang Li, Lianli Gao, Dongxiang Zhang, and Jingkuan Song. 2019. One Network for Multi-Domains: Domain Adaptive Hashing with Intersectant Generative Adversarial Networks. In *IJCAI*. 2477–2483.

[12] Jiun Tian Hoe, Kam Woh Ng, Tianyu Zhang, Chee Seng Chan, Yi-Zhe Song, and Tao Xiang. 2021. One loss for all: Deep hashing with a single cosine similarity based learning objective. In *NeurIPS*.

[13] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. 2017. Learning discrete representations via information maximizing self-augmented training. In *ICML*. 1558–1567.

[14] Fuxiang Huang, Lei Zhang, and Xinbo Gao. 2022. Domain Adaptation Preconceived Hashing for Unconstrained Visual Retrieval. *IEEE TNNLS* 33, 10 (2022), 5641–5655.

[15] Fuxiang Huang, Lei Zhang, Yang Yang, and Xichuan Zhou. 2020. Probability weighted compact feature for domain adaptive retrieval. In *CVPR*. 9582–9591.

[16] Jonathan J. Hull. 1994. A database for handwritten text recognition research. *IEEE TPAMI* 16, 5 (1994), 550–554.

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling Up Visual and Vision-Language Representation Learning With Noisy Text Supervision. In *ICML*, Vol. 139. 4904–4916.

[18] Qing-Yuan Jiang and Wu-Jun Li. 2015. Scalable graph hashing with feature transformation. In *IJCAI*. 2248–2254.

[19] Zhongming Jin, Cheng Li, Yue Lin, and Deng Cai. 2014. Density Sensitive Hashing. *IEEE T. Cybern* 44, 8 (2014), 1362–1371.

[20] Zhengfeng Lai, Noranart Vesdapunt, Ning Zhou, Jun Wu, Cong Phuoc Huynh, Xuelu Li, Kah Kuen Fu, and Chen-Nee Chuah. 2023. PADCLIP: Pseudo-labeling with Adaptive Debiasing in CLIP for Unsupervised Domain Adaptation. In *ICCV*. 16155–16165.

[21] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. In *Proc IEEE*, Vol. 86. 2278–2324.

[22] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. 2021. Maximum Density Divergence for Domain Adaptation. *IEEE TPAMI* 43, 11 (2021), 3918–3930.

[23] Xinyao Li, Yuke Li, Zhekai Du, Fengling Li, Ke Lu, and Jingjing Li. 2024. Split to Merge: Unifying Separated Modalities for Unsupervised Domain Adaptation. In *CVPR*. 23364–23374.

[24] Yunqiang Li and Jan van Gemert. 2021. Deep unsupervised image hashing by maximizing bit entropy. In *AAAI*, Vol. 35. 2002–2010.

[25] Jian Liang, Dapeng Hu, and Jiashi Feng. 2020. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *ICML*. 6028–6039.

[26] Qinghong Lin, Xiaojun Chen, Qin Zhang, Shaotian Cai, Wenzhe Zhao, and Hongfa Wang. 2022. Deep unsupervised hashing with latent semantic components. In *AAAI*, Vol. 36. 7488–7496.

[27] Hong Liu, Rongrong Ji, Jingdong Wang, and Chunhua Shen. 2019. Ordinal Constraint Binary Coding for Approximate Nearest Neighbor Search. *IEEE TPAMI* 41, 4 (2019), 941–955.

[28] Hong Liu, Jianmin Wang, and Mingsheng Long. 2021. Cycle Self-Training for Domain Adaptation. In *NeurIPS*, Vol. 34. 22968–22981.

[29] Fuchen Long, Ting Yao, Qi Dai, Xinmei Tian, Jiebo Luo, and Tao Mei. 2018. Deep Domain Adaptation Hashing with Adversarial Learning. In *SIGIR*. 725–734.

[30] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning Transferable Features with Deep Adaptation Networks. In *ICML*, Vol. 37. 97–105.

[31] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. 2018. Conditional Adversarial Domain Adaptation. In *NeurIPS*.

[32] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. 2017. Deep Transfer Learning with Joint Adaptation Networks. In *ICML*, Vol. 70. 2208–2217.

[33] Junyu Luo, Yusheng Zhao, Xiao Luo, Zhiping Xiao, Wei Ju, Li Shen, Dacheng Tao, and Ming Zhang. 2025. Cross-domain diffusion with progressive alignment for efficient adaptive retrieval. *IEEE TIP* 34 (2025), 1820–1834.

[34] Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE TKDE* 22, 10 (2010), 1345–1359.

[35] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. 2021. Meta Pseudo Labels. In *CVPR*. 11557–11568.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwala, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, Vol. 139. 8748–8763.

[37] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting Visual Category Models to New Domains. In *ECCV*. 213–226.

[38] Yang Shi, Xiushan Nie, Xingbo Liu, Li Zou, and Yilong Yin. 2022. Supervised adaptive similarity matrix hashing. *IEEE TIP* 31 (2022), 2755–2766.

[39] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. In *arXiv preprint arXiv:1409.1556,*.

[40] Mainak Singha, Harsh Pal, Ankit Jha, and Biplab Banerjee. 2023. AD-CLIP: Adapting Domains in Prompt Space Using CLIP. In *ICCV Workshops*. 4355–4364.

[41] Baochen Sun, Jiashi Feng, and Kate Saenko. 2016. Return of Frustratingly Easy Domain Adaptation. In *AAAI*, Vol. 30. 2058–2065.

[42] Baochen Sun and Kate Saenko. 2016. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *ECCV Workshops*. 443–450.

[43] Tao Sun, Cheng Lu, and Haibin Ling. 2023. Domain Adaptation with Adversarial Training on Penultimate Activations. In *AAAI*, Vol. 37. 9935–9943.

[44] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. 2022. Safe Self-Refinement for Transformer-Based Domain Adaptation. In *CVPR*. 7191–7200.

[45] Song Tang, Wenxin Su, Yan Gan, Mao Ye, Jianwei Zhang, and Xiatian Zhu. 2025. Proxy Denoising for Source-Free Domain Adaptation. In *ICLR*.

[46] Rong-Cheng Tu, Xian-Ling Mao, Kevin Qinghong Lin, Chengfei Cai, Weize Qin, Wei Wei, Hongfa Wang, and Heyan Huang. 2023. Unsupervised hashing with semantic concept mining. In *SIGMOD*.

[47] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *JMLR* 9, 86 (2008), 2579–2605.

[48] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep Hashing Network for Unsupervised Domain Adaptation. In *CVPR*. 5018–5027.

[49] Haixin Wang, Jinan Sun, Xiao Luo, Wei Xiang, Shikun Zhang, Chong Chen, and Xian-Sheng Hua. 2023. Toward Effective Domain Adaptive Retrieval. *IEEE TIP* 32 (2023), 1285–1299.

[50] Haixin Wang, Jinan Sun, Shikun Zhang, Wei Xiang, Chong Chen, Xian-Sheng Hua, and Xiao Luo. 2023. DANCE: Learning A Domain Adaptive Framework for Deep Hashing. In *ACM MM*. 3319–3330.

[51] Haixin Wang, Hao Wu, Jinan Sun, Shikun Zhang, Chong Chen, Xian-Sheng Hua, and Xiao Luo. 2023. IDEA: An Invariant Perspective for Efficient Domain Adaptive Image Retrieval. In *NeurIPS*, Vol. 36. 57256–57275.

[52] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. 2018. A Survey on Learning to Hash. *IEEE TPAMI* 40, 4 (2018), 769–790.

[53] Ziwei Wang, Han Xiao, Yueqi Duan, Jie Zhou, and Jiwen Lu. 2023. Learning Deep Binary Descriptors via Bitwise Interaction Mining. *IEEE TPAMI* 45, 2 (2023), 1919–1933.

[54] Haifeng Xia, Taotao Jing, Chen Chen, and Zhengming Ding. 2021. Semi-supervised Domain Adaptive Retrieval via Discriminative Hashing Learning. In *ACM MM*. 3853–3861.

[55] Zhiqing Xiao, Haobo Wang, Ying Jin, Lei Feng, Gang Chen, Fei Huang, and Junbo Zhao. 2023. SPA: A Graph Spectral Alignment Perspective for Domain Adaptation. In *NeurIPS*, Vol. 36. 37252–37272.

[56] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. 2020. Adversarial Domain Adaptation with Domain Mixup. In *AAAI*, Vol. 34. 6502–6509.

[57] Chenggang Yan, Biao Gong, Yuxuan Wei, and Yue Gao. 2020. Deep multi-view enhancement hashing for image retrieval. *IEEE TPAMI* 43, 4 (2020), 1445–1451.

[58] Lihe Yang, Lei Qi, Litong Feng, Wayne Zhang, and Yinghuan Shi. 2023. Revisiting Weak-to-Strong Consistency in Semi-Supervised Semantic Segmentation. In *CVPR*. 7236–7246.

[59] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. 2020. Central Similarity Quantization for Efficient Image and Video Retrieval. In *CVPR*. 3083–3092.

[60] Zhongqi Yue, Hanwang Zhang, and Qianru Sun. 2023. Make the U in UDA Matter: Invariant Consistency Learning for Unsupervised Domain Adaptation. In *NeurIPS*, Vol. 36. 26991–27004.

[61] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. FlexMatch: Boosting Semi-Supervised Learning with Curriculum Pseudo Labeling. In *NeurIPS*, Vol. 34. 18408–18419.

[62] Lei Zhang, Ji Liu, Yang Yang, Fuxiang Huang Feiping Nie, and David Zhang. 2020. Optimal Projection Guided Transfer Hashing for Image Retrieval. *IEEE TCSVT* 30, 10 (2020), 3788–3802.

[63] Qi Zhang, Liang Hu, Longbing Cao, Chongyang Shi, Shoujin Wang, and Dora D Liu. 2022. A probabilistic code balance constraint with compactness and informativeness enhancement for deep supervised hashing. In *IJCAI*. 1651–1657.

[64] Chaoyang Zhou, Zengmao Wang, Bo Du, and Yong Luo. 2024. Cycle Self-Refinement for Multi-Source Domain Adaptation. In *AAAI*, Vol. 38. 17096–17104.

[65] Lei Zhu, Chaoqun Zheng, Weili Guan, Jingjing Li, Yang Yang, and Heng Tao Shen. 2024. Multi-Modal Hashing for Efficient Multimedia Retrieval: A Survey. *IEEE TKDE* 36, 1 (2024), 239–260.

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 main claims made in abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA 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 NA 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: We discuss the limitations of the work in Appendix B.

   Guidelines:

   - The answer NA 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 theory assumptions are included in Section 3.

Guidelines:

- The answer NA 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: All the information needed to reproduce the main experimental results of the paper have been provided.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: The code is available in Supplementary, and the datasets are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All the training and test details have been clearly introduced.

Guidelines:

- The answer NA 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: [Yes]

Justification: We evaluate the method through different metrics, and the details can be found in Section 4 and Appendix A.

Guidelines:

- The answer NA 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: [Yes]

   Justification: We have provided sufficient information on the computer resources needed to reproduce the experiments.

   Guidelines:

   - The answer NA 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: This paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA 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 broader impacts in Appendix B.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: The codes, datasets and models used in this paper are publicly available. And the compared methods are all designed for image retrieval task.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

A Additional Experimental Results

A.1 Performance of Cross-Domain Retrieval

Table 6 presents the MAP results for cross-domain retrieval tasks under varying hash code lengths. Across all code lengths, VPDS consistently outperforms the compared methods on most tasks and achieves a notable improvement in average performance, demonstrating its robustness and effectiveness in different hash settings.

Table 6: MAP (%) of cross-domain retrieval tasks with various bits on Office-Home and Office-31.

		Office-Home						Office-31						
Method	Bit	A→R	R→A	C→R	R→C	P→R	R→P	A→D	A→W	D→A	D→W	W→A	W→D	Avg
GTH-g	16	10.20	9.51	6.04	5.90	10.84	11.08	26.25	11.85	15.76	34.40	16.14	31.79	15.81
	32	13.08	13.93	7.86	9.52	15.28	16.17	28.35	15.76	21.15	41.36	19.23	42.86	20.38
	128	16.51	19.52	8.53	13.92	20.81	21.24	31.68	20.55	21.93	50.09	20.17	53.54	24.87
GTH-h	16	9.54	8.18	6.17	6.30	11.32	10.81	24.86	11.94	19.02	34.15	14.66	40.58	16.46
	32	13.43	12.67	7.77	8.97	15.71	15.36	24.65	15.56	20.98	41.67	17.97	42.33	19.76
	128	13.78	19.73	8.57	14.54	21.16	20.16	32.01	22.16	22.56	51.62	21.55	51.57	24.20
DAPH	16	11.92	14.46	8.16	8.12	17.11	14.37	22.46	15.94	19.69	52.39	19.44	34.01	19.84
	32	17.72	19.63	10.48	10.64	22.47	20.25	25.15	19.09	21.99	54.28	22.00	36.58	23.36
	128	22.27	23.78	14.32	13.39	28.25	25.34	32.90	27.49	29.11	64.25	26.58	47.59	29.61
CPH	16	62.19	53.41	53.47	35.64	64.18	66.65	59.54	52.54	43.47	90.86	55.53	98.33	61.32
	32	67.87	60.16	56.71	39.71	68.31	71.72	64.00	56.33	49.34	95.04	59.18	99.49	65.66
	128	72.17	64.62	53.25	42.71	72.89	75.50	69.20	61.18	54.89	97.54	61.49	99.99	68.79
VPDS	16	67.05	54.63	57.28	37.66	68.67	78.25	77.36	84.11	63.67	98.78	68.39	100	71.32
	32	70.67	56.65	62.07	41.89	73.42	82.18	83.15	88.85	68.7	98.88	67.96	98.43	74.40
	128	70.74	61.86	61.86	40.84	73.2	82.42	85.67	88.76	72.84	98.80	64.46	99.86	75.11

A.2 Details of Ablation Variants

In Section 4.3, we perform an ablation study to evaluate the contribution of each component in VPDS. Specifically, we construct seven variants by removing individual components from the full model, including:

(1) VPDS w/o $\mathcal{L}_{P_F}$: removing prototype-based alignment in the feature space.

(2) VPDS w/o $\mathcal{L}_{P_H}$: removing prototype-based alignment in the Hamming space.

(3) VPDS w/o $\mathcal{L}_P$: removing prototype-based alignments in both the feature and Hamming spaces.

(4) VPDS w/o $\mathcal{L}_i$ (feat): excluding the information maximization technique in the feature space.

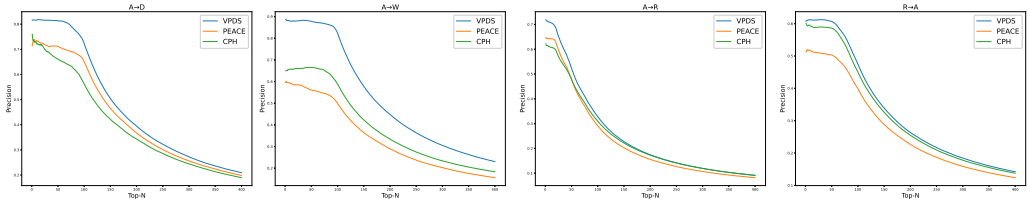
(5) VPDS w/o $\mathcal{L}_i$ (Hamm): excluding the information maximization technique in the Hamming space.

(6) VPDS w/o $\mathcal{L}_i$: excluding the information maximization technique in both the feature and Hamming spaces.

(7) VPDS w/o $\mathcal{L}_t$: removing the pseudo-label-based self-supervised learning objective.

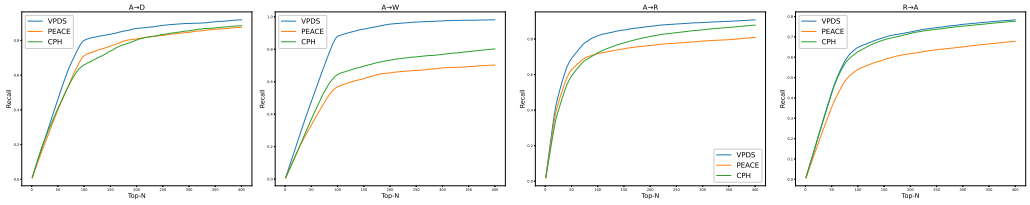
A.3 Top-N Precision Curve and Top-N Recall Curve

The Top-N precision curves and Top-N recall curves for four randomly selected tasks are shown in Figures 5 and 6. Our method consistently achieves higher precision and recall across different values of N, highlighting its effectiveness in cross-domain retrieval tasks.



(a) (b) (c) (d)

Figure 5: Top-N precision curves on four tasks randomly selected from Office-Home and Office-31.



(a) (b) (c) (d)

Figure 6: Top-N recall curves on four tasks randomly selected from Office-Home and Office-31.

A.4 Time and Memory Consumption

To evaluate the computational efficiency of our method, we compare its time and memory consumption against PEACE, which also exploits both feature and Hamming spaces. While CPH similarly leverages dual-space information, it uses pre-encoded features as input, thereby bypassing the feature extraction stage. As shown in Table 7, we report the training time over 10 epochs using a fixed batch size. Although our method incurs a higher training time compared to PEACE, it exhibits obviously lower memory consumption. Considering the obvious performance improvements observed in prior metrics such as MAP, the increased computational cost is a reasonable trade-off and can be acceptable.

Table 7: Time and memory consumption.

Office-Home								
	Method	A→R	R→A	C→R	R→C	P→R	R→P	Avg
Time (s)	PEACE	175.93	160.62	277.41	280.05	278.08	288.44	243.42
	VPDS	193.26	174.29	308.36	299.77	303.58	302.57	263.64
Memory (MiB)	PEACE	17,688						
	VPDS	9,840						
Office-31								
	Method	A→D	A→W	D→A	D→W	W→A	W→D	Avg
Time (s)	PEACE	37.82	55.08	39.08	39.97	59.09	37.22	44.71
	VPDS	37.94	53.76	39.80	39.53	60.84	37.92	44.97
Memory (MiB)	PEACE	17,686						
	VPDS	9,864						

B Broader Impacts and Limitations

This paper proposes a novel domain adaptive retrieval method, which achieves substantial performance gains over existing methods. We propose a VLM assisted pseudo-labeling strategy that enhances the reliability of pseudo-labels. This strategy can be seamlessly integrated with other unsupervised and cross-domain learning approaches, thereby improving performance in scenarios with limited labeled data. Additionally, we present a dual space adaptation approach that leverages

the semantically rich feature space to guide learning of discriminative hash codes, while eliminating the explicit constraints from Hamming space to feature space, providing a new direction for advancing cross-domain retrieval. In this paper, we only focus on adaptive retrieval across two domains, while adaptive retrieval across multiple domains is more practical in real-world, thus we will explore such scenarios in our future work.