A GENERAL DOMAIN-ADAPTIVE CLUSTERING APPROACH VIA CROSS-DATASET PROTOTYPE ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep clustering typically learns cluster structures independently for each dataset, limiting its ability to exploit shared patterns across heterogeneous domains. We propose CPC, a general domain-adaptive clustering approach based on cross-dataset prototype alignment, which jointly learns a shared prototype space across multiple domains. By allowing different domains to activate shared prototypes, CPC establishes cross-dataset structural correspondence and discovers representative prototypes that serve as compact cluster anchors. Experiments on 11 heterogeneous real-world datasets show that CPC achieves the best or tied-best accuracy on 9 datasets and substantially outperforms existing clustering methods, with improvements of over 40 percentage points on challenging datasets such as Primary Tumor. These results demonstrate the effectiveness of shared prototype spaces for domain-adaptive clustering.

1 INTRODUCTION

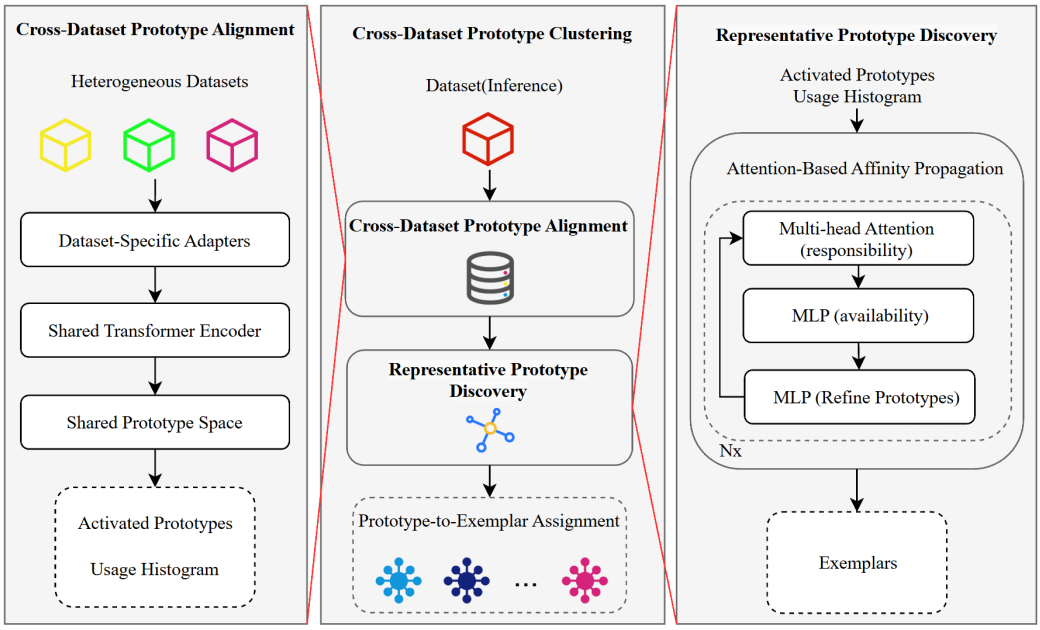

Figure 1: Overview of CPC. In the first stage, Cross-Dataset Prototype Alignment (CPA) jointly maps heterogeneous datasets into a common latent space and learns a shared prototype codebook. In the second stage, Representative Prototype Discovery (RPD) models interactions among activated prototypes and selects a compact subset of representative ones as cluster anchors, from which the final cluster assignments are derived.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Deep clustering aims to discover meaningful structures from unlabeled data by jointly learning representations and cluster assignments (Ren et al., 2024; Yin et al., 2026). Recent advances have improved clustering through neural representation learning (Cai et al., 2022), autoencoder-based architectures (Abbasimehr & Noshad, 2025), self-supervised learning (Li et al., 2022), and learnable cluster structures (Wang et al., 2023). However, most existing methods optimize representations and cluster structures independently for each dataset (Zhu et al., 2026), leaving potentially shared structural patterns across heterogeneous datasets underexploited. Although heterogeneous datasets may differ substantially in feature spaces, scales, and distributions (Putrama & Martinek, 2024; Wenz et al., 2023), they can still exhibit recurring structures at a higher level of abstraction. This motivates us to learn a shared structural space across datasets, rather than optimizing a separate clustering space for each dataset. This setting differs from conventional domain adaptation, which typically transfers knowledge from a designated source domain to a target domain (Lu et al., 2025; Qin et al., 2025). Here, multiple heterogeneous datasets jointly contribute to a shared representation without assuming a source–target relationship. We consider this setting as a form of *domain-adaptive*, where cross-dataset correspondence is established through a shared prototype space. Specifically, heterogeneous features are mapped into a common latent space and quantized using a jointly learned prototype codebook. Since the codebook is shared, samples from different domains can activate the same prototype vectors, providing an implicit prototype-level alignment while preserving dataset-specific characteristics. The shared prototypes represent latent structural patterns rather than predefined classes or dataset-specific cluster centers.

A shared prototype space, however, does not directly provide a compact clustering structure. Multiple activated prototypes may describe similar patterns, while several prototypes may contribute to the same underlying cluster. Directly treating all activated prototypes as cluster representatives can therefore introduce redundancy and fragmented structures. To address this issue, we propose *CPC*, a general domain-adaptive clustering approach via cross-dataset prototype alignment, as illustrated in Fig. 1. CPC follows a two-stage *Learn–Select* framework. First, *Cross-Dataset Prototype Alignment (CPA)* jointly learns representations and a shared prototype space from heterogeneous datasets. Dataset-specific adapters accommodate differences among input spaces, while a shared Transformer encoder and vector-quantization codebook capture recurring structural patterns. Second, *Representative Prototype Discovery (RPD)* models interactions among activated prototypes and selects a compact set of representative prototypes as cluster anchors, from which the final cluster assignments are derived. We evaluate CPC on 11 heterogeneous real-world datasets using four widely adopted metrics. CPC achieves the best or tied-best accuracy on 9 of the 11 datasets and substantially outperforms existing clustering methods, with accuracy improvements of more than 40 percentage points on challenging datasets such as Primary Tumor. These results demonstrate the effectiveness of cross-dataset prototype alignment for domain-adaptive clustering.

The main contributions of this work are summarized as follows:

- **Cross-dataset prototype alignment for domain-adaptive clustering.** We propose CPC, which jointly learns a shared prototype space across heterogeneous datasets and establishes cross-dataset structural correspondence through shared prototype activation.
- **Representative prototype discovery.** We develop an attention-based prototype discovery mechanism that identifies a compact and non-redundant subset of activated prototypes as cluster anchors.
- **Comprehensive evaluation.** Experiments on 11 heterogeneous real-world datasets demonstrate the benefits of joint cross-dataset prototype alignment and representative prototype discovery, with CPC achieving the best or tied-best accuracy on 9 datasets.

2 Related Work

Deep clustering and representation learning. Deep clustering jointly learns representations and cluster assignments from unlabeled data. Representative approaches include DEC, which iteratively refines an auxiliary target distribution Xie et al. (2016), IDEC, which incorporates reconstruction to preserve local structure Guo et al. (2017), and DeepCluster, which alternates between clustering and representation learning Caron et al. (2018). Recent methods further exploit contrastive objectives to improve cluster-level structure Li et al. (2021); Lv et al. (2025). These approaches primarily

optimize clustering within individual datasets, limiting their ability to exploit structural regularities shared across heterogeneous datasets. CPC instead jointly learns representations from multiple datasets and constructs a shared prototype space while retaining dataset-specific adaptation.

Domain adaptation and multi-domain learning. Domain adaptation typically addresses distributional differences between a source and a target domain through representation alignment or knowledge transfer Wilson & Cook (2020); Singhal et al. (2023). Multi-domain learning extends knowledge sharing across multiple domains, often by learning domain-invariant representations Royer et al. (2023). CPC considers a different setting for domain-adaptive clustering: multiple heterogeneous datasets jointly participate in learning without a designated source–target relationship. Rather than explicitly aligning complete feature distributions, CPC establishes cross-dataset correspondence through a shared prototype space, where different datasets can activate common prototypes while preserving domain-specific characteristics.

Prototype-based and discrete representation learning. Prototype-based methods represent data using a finite set of latent vectors that capture recurring patterns. VQ-VAE demonstrates how a learned codebook can provide a compact discrete representation Razavi et al. (2019), while SwAV and ProPos exploit learnable prototypes for clustering and representation learning Caron et al. (2020); Huang et al. (2022). Existing approaches mainly use prototypes within a single learning setting. CPC instead learns the prototype codebook jointly across heterogeneous datasets. The shared codebook allows different datasets to activate the same prototype vectors, forming an implicit prototype-level correspondence that serves as the basis for cross-dataset prototype alignment.

Attention-based representative discovery. Attention mechanisms provide an effective way to model interactions among elements in a set, as demonstrated by Set Transformer and Slot Attention Lee et al. (2019); Locatello et al. (2020). Exemplar-based methods such as Affinity Propagation further investigate representative selection through iterative message passing Dueck & Frey (2007). CPC operates at a different level: its Representative Prototype Discovery (RPD) models interactions among activated prototypes rather than individual samples, and selects a compact subset of representative prototypes as cluster anchors. This prototype-level selection reduces redundancy in the shared prototype space and bridges prototype learning with the final dataset-specific clustering structure.

3 METHOD

Let $\mathcal{D} = \{\mathcal{X}^{(m)}\}_{m=1}^{M}$ denote M heterogeneous datasets, where $\mathcal{X}^{(m)} = \{\mathbf{x}_i^{(m)}\}_{i=1}^{N_m}$ with $\mathbf{x}_i^{(m)} \in \mathbb{R}^{D_m}$. Feature dimensions and distributions may differ substantially across datasets. Instead of learning an independent clustering representation for each dataset, we jointly learn a shared prototype space that captures recurring structural patterns across datasets and derive a compact cluster structure from it. CPC follows a two-stage *Learn–Select* framework: **Cross-Dataset Prototype Alignment (CPA)** jointly learns a shared prototype space, and **Representative Prototype Discovery (RPD)** selects representative prototypes from those activated by each dataset.

3.1 CROSS-DATASET PROTOTYPE ALIGNMENT (CPA)

Dataset-specific input adaptation. Since the original feature spaces differ across datasets, we introduce a lightweight input adapter $\phi_m(\cdot)$ for each dataset: $\mathbf{H}_i^{(m)} = \phi_m(\mathbf{x}_i^{(m)})$. The adapters provide dataset-specific interfaces to a common representation space without constructing independent clustering spaces.

Shared representation learning. The adapted representations are processed by a shared Transformer encoder with learnable positional embeddings $\mathbf{P}$, followed by pooling:

$$\mathbf{Z}_i^{(m)} = \mathrm{Transformer}(\mathbf{H}_i^{(m)} + \mathbf{P}), \qquad \mathbf{z}_i^{(m)} = \mathrm{Pool}(\mathbf{Z}_i^{(m)}) \in \mathbb{R}^d. \tag{1}$$

Because all datasets jointly optimize the same encoder, heterogeneous samples are mapped into a common latent space in which shared structural patterns can be captured.

Shared prototype space and implicit alignment. We introduce a learnable codebook $\mathcal{S} = \{\mathbf{s}_k\}_{k=1}^{K}$ with $\mathbf{s}_k \in \mathbb{R}^d$, and jointly optimize it using all datasets. Each sample is quantized to its closest prototype:

$$k_i^{(m)} = \arg\min_k \left\| \mathbf{z}_i^{(m)} - \mathbf{s}_k \right\|_2^2, \qquad \mathbf{q}_i^{(m)} = \mathbf{s}_{k_i^{(m)}}. \tag{2}$$

The codebook is shared across datasets rather than tied to predefined classes or dataset-specific cluster centers. Consequently, different datasets can activate the same prototype vectors, establishing an implicit prototype-level correspondence while preserving their dataset-specific characteristics. For dataset m, the activated prototypes and their usage distribution are

$$\mathcal{A}^{(m)} = \left\{ \mathbf{s}_k \mid k \in \mathcal{I}^{(m)} \right\}, \qquad u_k^{(m)} = \frac{n_k^{(m)}}{\sum_{j=1}^{K} n_j^{(m)}}, \tag{3}$$

where $\mathcal{I}^{(m)}$ denotes the activated prototype indices and $n_k^{(m)}$ is the number of samples assigned to prototype k. The K prototypes form a shared structural vocabulary rather than a predefined set of clusters; multiple prototypes may describe similar patterns and are therefore further processed by RPD.

Dataset-specific decoders. Each dataset uses an independent decoder to reconstruct its original feature space: $\hat{\mathbf{x}}_i^{(m)} = \psi_m(\mathbf{q}_i^{(m)})$. Thus, adapters and decoders accommodate dataset-specific characteristics, while the Transformer encoder and codebook provide the shared representation components.

Joint optimization. CPA is optimized by four complementary objectives:

$$\mathcal{L}_{\text{CPA}} = \mathcal{L}_{\text{rec}} + \lambda_1 \mathcal{L}_{\text{commit}} + \lambda_2 \mathcal{L}_{\text{con}} + \lambda_3 \mathcal{L}_{\text{ent}}. \tag{4}$$

The reconstruction loss $\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$, with $N = \sum_{m=1}^{M} N_m$, preserves information from the original feature space. The commitment loss $\mathcal{L}_{\text{commit}} = \|\text{sg}[\mathbf{z}_i] - \mathbf{q}_i\|_2^2 + \beta \|\mathbf{z}_i - \text{sg}[\mathbf{q}_i]\|_2^2$ stabilizes quantization, where $\text{sg}[\cdot]$ denotes the stop-gradient operation. The contrastive consistency loss $\mathcal{L}_{\text{con}}$, an InfoNCE objective between two augmented views of the same sample with cosine similarity, improves robustness under perturbations. The utilization regularization $\mathcal{L}_{\text{ent}} = 1 - H(\mathbf{u}^{(m)})/\log K$, defined on the usage distribution in Eq. (3) and averaged over datasets, prevents collapse of most samples onto a few prototypes; normalization by $\log K$ bounds the loss in $[0, 1]$ and makes its strength independent of codebook size.

3.2 Representative Prototype Discovery (RPD)

The shared prototype space provides broad structural coverage, but a dataset may activate multiple similar or redundant prototypes. Treating all activated prototypes as cluster representatives can therefore produce fragmented structures. RPD performs clustering at the prototype level. For dataset m, let K_m prototypes be activated, with

$$\mathbf{P}^{(m)} = \left[\mathbf{p}_1^{(m)}, \ldots, \mathbf{p}_{K_m}^{(m)} \right] \in \mathbb{R}^{K_m \times d}, \qquad \mathbf{u}^{(m)} = \left[u_1^{(m)}, \ldots, u_{K_m}^{(m)} \right]. \tag{5}$$

Iterative prototype interaction. We model relationships among activated prototypes using multi-head self-attention, initialized by $\mathbf{H}^{(0)} = \mathbf{P}^{(m)}$:

$$\widetilde{\mathbf{H}}^{(t+1)} = \text{MHA}(\mathbf{H}^{(t)}, \mathbf{H}^{(t)}, \mathbf{H}^{(t)}), \qquad \mathbf{H}^{(t+1)} = f_{\text{refine}}\left(\mathbf{R}^{(t)} \widetilde{\mathbf{H}}^{(t+1)} \right), \tag{6}$$

where $\mathbf{R}^{(t)} \in \mathbb{R}^{K_m \times K_m}$ is the soft responsibility matrix and $R_{ij}^{(t)}$ measures the degree to which prototype j represents prototype i. Repeated interaction enables each prototype to incorporate information from the entire activated prototype set.

Representative prototype probability. The representative probability combines learned availability, prototype usage, and a preference bias:

$$a_j^{(m)} = f_{\text{avail}}\left(\mathbf{p}_j^{(m)} \right) + \log\left(u_j^{(m)} + \epsilon \right) + b, \qquad e_j^{(m)} = \sigma\left(a_j^{(m)} / \alpha \right), \tag{7}$$

where $f_{\text{avail}}(\cdot)$ is learnable, b is a bias, ϵ ensures numerical stability, and α controls the selection sharpness.

Prototype-level cluster formation. Representative prototypes are selected as $\mathcal{E}^{(m)} = \{\mathbf{p}_j^{(m)} \mid e_j^{(m)} > \gamma\}$. Each activated prototype is then assigned to its most similar representative according to cosine similarity:

$$c_i^{(m)} = \arg \max_{\mathbf{p}_j^{(m)} \in \mathcal{E}^{(m)}} \frac{\left(\mathbf{p}_i^{(m)}\right)^\top \mathbf{p}_j^{(m)}}{\left\|\mathbf{p}_i^{(m)}\right\|_2 \left\|\mathbf{p}_j^{(m)}\right\|_2}, \qquad y_i^{(m)} = c_{k_i^{(m)}}^{(m)}. \tag{8}$$

Thus, clustering follows the hierarchy *sample* $\rightarrow$ *activated prototype* $\rightarrow$ *representative prototype* $\rightarrow$ *cluster*, reducing the number of objects involved in structural discovery while explicitly addressing prototype redundancy.

Self-supervised optimization. RPD is optimized without ground-truth assignments using

$$\mathcal{L}_{\mathrm{RPD}} = \lambda_4 \mathcal{L}_{\mathrm{pref}} + \lambda_5 \mathcal{L}_{\mathrm{use}} + \lambda_6 \mathcal{L}_{\mathrm{div}}. \tag{9}$$

The preference loss $\mathcal{L}_{\mathrm{pref}} = \left(\frac{\sum_{j=1}^{K_m} e_j - K_t}{K_m}\right)^2$ regulates the expected number of selected prototypes around a target budget $K_t = \max(2, \lfloor \rho K_m \rfloor)$ with selection ratio $\rho = 0.4$. The usage-consistency loss $\mathcal{L}_{\mathrm{use}} = \|\mathbf{r} - \mathbf{e}\|_2^2$, with incoming responsibility $r_i = \frac{1}{K_m} \sum_{j=1}^{K_m} R_{ji}$, encourages prototypes receiving stronger structural responsibility to obtain higher representative probabilities. The diversity loss $\mathcal{L}_{\mathrm{div}} = \frac{\sum_{i,j} \mathrm{Cos}_{ij}\, e_i e_j}{\sum_{i,j} e_i e_j + \epsilon}$ discourages simultaneous selection of similar prototypes and promotes complementary structural coverage.

3.3 Overall Framework

CPC follows a two-stage *Learn–Select* process (Figs. 3 and 4, Appendix). In the first stage, all datasets jointly learn the shared prototype space through $\mathcal{L}_{\mathrm{CPA}}$ (Eq. 4), where shared prototype activation provides implicit cross-dataset alignment. In the second stage, the learned prototype space is fixed and RPD independently discovers representative prototypes for each dataset through $\mathcal{L}_{\mathrm{RPD}}$ (Eq. 9). The resulting hierarchy from shared prototypes to representative prototypes provides a compact dataset-specific clustering structure.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate CPC on 11 real-world datasets from the UCI Machine Learning Repository Dua & Graff (2019), covering diverse application domains and exhibiting substantial variations in sample size, feature dimensionality, and class structure. This heterogeneity provides a natural testbed for cross-dataset joint learning and shared prototype alignment. Dataset statistics are summarized in Table 1.

Baselines. We compare CPC with seven representative clustering methods covering classical, deep, generative, spectral, graph-based, and prototype-oriented learning paradigms: **KMeans** Steinley (2006), **DEC** Xie et al. (2016), **VaDE** Jiang et al. (2016), **SpectralNet** Alshammari et al. (2022), **GMM** He et al. (2010), **SDSGC** Wang et al. (2025), and **SSAE** Vardakas et al. (2026). These methods perform clustering independently on each dataset and therefore provide a comparison against CPC's cross-dataset joint learning setting. All baselines use their official implementations, with hyperparameters tuned separately for each dataset following the corresponding original papers.

Evaluation metrics. We evaluate clustering quality using four widely adopted metrics: Accuracy (ACC), Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and F1-score Mahon & Lukasiewicz (2023). Before evaluation, predicted cluster indices are optimally matched to ground-truth labels using the Hungarian algorithm. Higher values indicate better clustering quality. We report the mean and standard deviation over 20 independent runs with different random seeds.

Table 1: Statistics of the 11 UCI datasets used in our experiments.

Dataset	Abbreviation	Samples	Clusters
Lenses	Ls	24	3
Lung Cancer	Lc	32	3
Promoter Sequences	Ps	106	2
Hayes-Roth	Hr	132	3
Heart Disease	Hd	303	5
Solar Flare	Sf	323	6
Primary Tumor	Pt	339	21
Balance Scale	Bs	625	3
Credit Approval	Ca	690	2
Mammographic Mass	Mm	824	2
Tic-Tac-Toe	Tt	958	2

Implementation details. All features are standardized to zero mean and unit variance. The 11 datasets jointly participate in the CPA stage, where dataset-specific adapters map their inputs into a shared latent space. The latent dimension is 64, the shared Transformer encoder has 2 layers with 4 attention heads, and the shared codebook contains $K = 64$ prototypes. We use Adam with a learning rate of 1×10^{-3} and $(\lambda_1, \lambda_2, \lambda_3) = (1.0, 0.5, 0.1)$ for the commitment, contrastive, and entropy terms. After CPA, the shared prototype space is fixed and RPD is optimized separately for each dataset, using 3 interaction steps, a learning rate of 1×10^{-4}, and $(\lambda_4, \lambda_5, \lambda_6) = (1.0, 1.0, 0.1)$.

Computational cost. CPC remains lightweight despite jointly processing multiple datasets. On Bs, for example, the complete model contains approximately 171.2K trainable parameters, with the shared encoder and prototype space accounting for the jointly learned components and dataset-specific adapters and decoders introducing only a small additional overhead. Once the shared prototype space is learned, RPD requires only 25.3K trainable parameters per dataset. A detailed parameter breakdown is provided in Appendix B.

4.2 Main Results

Table 2 summarizes the performance of CPC and seven representative clustering methods on 11 heterogeneous datasets. Across the 44 dataset–metric comparisons, CPC achieves the best or tied-best result in 22 cases, including the best or tied-best ACC on 9 of the 11 datasets. CPC also achieves the best result on all four metrics for Mm, Pt, and Sf, indicating consistent improvements rather than gains restricted to a single evaluation criterion. The improvements are particularly pronounced on datasets with more challenging cluster structures. On Pt, which contains 21 clusters, CPC achieves 82.76% ACC, improving by more than 40 percentage points over the strongest baseline (40.46%), while also obtaining substantial gains in NMI (74.68%), ARI (65.97%), and F1 (75.31%). On Sf, CPC reaches 90.32% ACC, 78.61% NMI, and 80.35% ARI, substantially exceeding all competing methods on these metrics. Similarly, on Mm, CPC achieves 85.71% ACC, 40.20% NMI, 47.09% ARI, and 85.42% F1. These results suggest that jointly learning a shared prototype space can capture structural patterns that are difficult to recover when clustering each dataset independently, while RPD further reduces prototype redundancy to form compact cluster structures.

CPC does not achieve the best performance on every dataset. For example, VaDE is best on Hr, DEC leads on Ps and several metrics of Hd and Ls, and SDSGC achieves the best NMI and ARI on Lc. These results indicate that dataset-specific clustering objectives can remain effective when the intrinsic structure of an individual dataset is well captured by a specialized model. Nevertheless, the overall results show that CPC maintains competitive performance across heterogeneous datasets while providing a shared prototype-based representation and dataset-specific cluster discovery within a unified framework.

Table 2: Clustering performance of different methods on 11 heterogeneous datasets. All results are reported as mean ± standard deviation (%) over 20 runs. The best mean result for each dataset and metric is highlighted in bold.

ACC

Dataset	KMeans	DEC	VaDE	SpectralNet	GMM	SDSGC	SSAE	CPC
Bs	69.57 ± 4.46	50.06 ± 1.06	67.68 ± 4.34	54.74 ± 7.88	51.86 ± 1.75	54.56 ± 3.50	52.50 ± 2.35	**75.00 ± 2.81**
Ca	54.67 ± 0.00	54.82 ± 0.00	75.80 ± 8.76	58.94 ± 8.15	62.28 ± 7.20	53.29 ± 1.16	54.00 ± 2.90	**76.19 ± 1.47**
Hr	38.64 ± 0.00	35.83 ± 0.99	**54.55 ± 5.33**	40.30 ± 4.36	42.08 ± 5.05	38.71 ± 0.92	47.27 ± 4.62	46.67 ± 1.56
Hd	57.49 ± 0.86	48.87 ± 4.05	40.74 ± 3.88	33.15 ± 4.71	38.32 ± 6.18	45.45 ± 3.77	33.62 ± 1.89	**63.33 ± 0.85**
Ls	63.96 ± 2.72	73.75 ± 1.91	58.33 ± 6.23	54.37 ± 5.81	50.42 ± 5.42	62.50 ± 6.68	48.75 ± 5.61	**80.00 ± 2.53**
Lc	**56.25 ± 4.84**	55.78 ± 0.82	40.62 ± 4.48	54.53 ± 6.52	51.41 ± 7.36	53.74 ± 1.68	46.88 ± 3.74	**56.25 ± 3.34**
Mm	79.76 ± 0.00	77.76 ± 0.21	78.55 ± 6.83	67.45 ± 7.58	75.60 ± 4.65	51.45 ± 4.47	62.90 ± 6.32	**85.71 ± 1.45**
Pt	40.46 ± 0.83	21.30 ± 0.64	28.74 ± 1.89	24.47 ± 1.71	25.22 ± 1.66	31.27 ± 2.06	25.01 ± 2.30	**82.76 ± 2.49**
Ps	74.43 ± 7.82	**89.25 ± 4.39**	73.65 ± 6.74	72.36 ± 11.79	72.97 ± 9.69	52.83 ± 3.75	57.59 ± 4.36	68.18 ± 3.16
Sf	88.85 ± 0.00	80.50 ± 0.00	47.99 ± 0.45	51.84 ± 6.61	53.90 ± 6.12	72.76 ± 5.26	41.92 ± 2.41	**90.32 ± 2.52**
Tt	65.34 ± 0.00	57.85 ± 0.09	63.99 ± 3.81	55.25 ± 4.65	54.21 ± 3.80	57.20 ± 2.43	55.17 ± 2.74	**73.53 ± 2.40**

NMI

Dataset	KMeans	DEC	VaDE	SpectralNet	GMM	SDSGC	SSAE	CPC
Bs	18.96 ± 10.09	12.79 ± 0.38	13.58 ± 6.89	19.06 ± 9.67	11.40 ± 2.25	23.61 ± 5.98	13.80 ± 4.46	**36.20 ± 2.37**
Ca	2.27 ± 0.00	2.53 ± 0.00	25.87 ± 8.83	6.41 ± 8.53	8.10 ± 6.32	1.05 ± 1.17	0.70 ± 0.61	**32.91 ± 1.04**
Hr	0.13 ± 0.11	0.86 ± 0.47	**23.72 ± 7.18**	2.83 ± 2.84	3.53 ± 3.09	2.51 ± 1.66	8.58 ± 4.69	11.27 ± 2.49
Hd	18.67 ± 1.09	**23.89 ± 0.99**	14.87 ± 2.73	13.64 ± 5.24	14.47 ± 2.66	6.35 ± 1.23	14.18 ± 2.84	20.05 ± 1.69
Ls	10.85 ± 10.08	**44.50 ± 3.91**	19.93 ± 8.35	25.13 ± 8.98	20.62 ± 9.31	19.65 ± 8.23	14.25 ± 5.97	34.96 ± 4.65
Lc	27.47 ± 6.36	21.40 ± 0.00	5.21 ± 2.13	23.27 ± 6.28	16.27 ± 8.92	**30.33 ± 2.41**	9.66 ± 3.07	27.95 ± 1.43
Mm	29.59 ± 0.00	23.64 ± 0.37	24.97 ± 7.54	13.53 ± 8.30	23.02 ± 6.69	2.73 ± 2.23	8.44 ± 2.83	**40.20 ± 1.94**
Pt	33.26 ± 0.62	28.88 ± 0.93	12.93 ± 0.48	30.83 ± 1.36	32.26 ± 1.19	29.41 ± 1.08	28.50 ± 3.57	**74.68 ± 0.78**
Ps	20.93 ± 12.04	**55.64 ± 6.93**	21.57 ± 5.17	20.76 ± 16.04	19.29 ± 13.57	4.87 ± 2.34	2.25 ± 2.37	13.94 ± 3.81
Sf	3.00 ± 0.13	4.17 ± 0.00	0.92 ± 0.30	2.13 ± 1.12	4.48 ± 0.52	1.36 ± 0.44	1.32 ± 0.78	**78.61 ± 0.49**
Tt	0.74 ± 0.00	0.75 ± 0.07	**8.43 ± 2.84**	0.99 ± 0.87	1.59 ± 1.02	7.88 ± 1.73	0.60 ± 0.38	6.41 ± 1.26

ARI

Dataset	KMeans	DEC	VaDE	SpectralNet	GMM	SDSGC	SSAE	CPC
Bs	**20.89 ± 10.43**	14.83 ± 0.56	18.47 ± 9.22	19.85 ± 10.30	13.47 ± 2.30	19.56 ± 3.77	15.18 ± 2.73	18.33 ± 2.15
Ca	0.46 ± 0.00	0.57 ± 0.00	**26.39 ± 8.79**	5.57 ± 9.08	7.78 ± 6.38	0.39 ± 0.54	0.82 ± 0.83	24.57 ± 1.83
Hr	-1.41 ± 0.07	-0.99 ± 0.28	**14.67 ± 7.82**	0.52 ± 2.36	2.32 ± 2.99	0.88 ± 0.38	5.69 ± 2.61	6.57 ± 1.50
Hd	12.83 ± 1.77	**28.61 ± 4.12**	11.26 ± 3.52	6.45 ± 6.21	12.35 ± 7.32	8.59 ± 2.68	9.15 ± 2.86	21.29 ± 0.65
Ls	2.50 ± 10.03	**45.07 ± 2.37**	7.25 ± 4.55	7.77 ± 8.46	7.57 ± 8.93	14.57 ± 5.19	1.94 ± 0.56	37.51 ± 2.25
Lc	14.15 ± 5.85	12.33 ± 0.00	3.75 ± 2.87	12.67 ± 6.50	7.86 ± 8.00	**20.02 ± 3.32**	2.48 ± 1.81	11.45 ± 1.76
Mm	35.35 ± 0.00	30.74 ± 0.47	32.53 ± 10.73	14.39 ± 11.51	27.00 ± 7.83	0.15 ± 0.08	10.81 ± 3.29	**47.09 ± 2.56**
Pt	8.59 ± 0.46	4.95 ± 0.60	15.74 ± 2.66	6.80 ± 1.32	7.41 ± 1.25	6.85 ± 1.16	5.75 ± 2.44	**65.97 ± 1.19**
Ps	25.63 ± 14.70	**62.04 ± 11.68**	26.18 ± 8.43	24.85 ± 19.46	24.14 ± 17.03	0.22 ± 0.16	2.14 ± 1.23	9.82 ± 0.93
Sf	-4.03 ± 2.13	17.81 ± 0.00	1.55 ± 0.23	1.07 ± 0.87	2.96 ± 0.80	2.18 ± 0.99	-0.37 ± 0.82	**80.35 ± 3.24**
Tt	1.95 ± 0.00	1.97 ± 0.08	7.65 ± 4.27	0.78 ± 0.34	1.06 ± 0.74	4.88 ± 1.90	0.88 ± 0.41	**11.22 ± 0.92**

F1

Dataset	KMeans	DEC	VaDE	SpectralNet	GMM	SDSGC	SSAE	CPC
Bs	47.90 ± 3.24	43.09 ± 0.60	46.87 ± 3.17	46.83 ± 6.63	45.33 ± 1.49	50.32 ± 1.40	46.30 ± 2.35	**74.83 ± 2.06**
Ca	35.35 ± 0.00	52.51 ± 0.00	**73.54 ± 4.60**	56.70 ± 8.94	58.18 ± 12.63	35.35 ± 3.57	53.79 ± 2.86	72.15 ± 1.38
Hr	18.58 ± 0.00	35.73 ± 1.04	**53.92 ± 5.68**	38.51 ± 5.41	40.97 ± 5.36	16.50 ± 1.38	47.31 ± 0.98	38.66 ± 1.07
Hd	23.95 ± 2.13	**34.61 ± 1.24**	28.90 ± 2.57	25.39 ± 3.18	28.97 ± 3.08	13.70 ± 1.22	27.31 ± 1.65	32.75 ± 0.91
Ls	33.04 ± 10.36	**63.28 ± 4.03**	55.77 ± 6.90	48.75 ± 8.87	47.77 ± 4.99	25.64 ± 3.17	46.06 ± 3.70	54.74 ± 1.77
Lc	48.75 ± 5.87	**57.00 ± 0.85**	19.20 ± 2.56	52.06 ± 8.91	51.65 ± 7.66	53.47 ± 2.18	46.65 ± 1.65	53.09 ± 2.16
Mm	79.66 ± 0.00	77.67 ± 0.22	78.53 ± 8.08	65.47 ± 9.03	75.23 ± 5.11	33.97 ± 3.50	62.70 ± 3.28	**85.42 ± 2.79**
Pt	16.05 ± 1.18	16.86 ± 0.99	22.74 ± 1.67	18.13 ± 1.75	18.59 ± 0.87	18.43 ± 1.35	14.58 ± 3.41	**75.31 ± 3.34**
Ps	74.29 ± 7.93	**89.07 ± 4.84**	73.14 ± 7.81	72.21 ± 11.86	72.97 ± 9.69	39.33 ± 4.20	57.55 ± 4.35	66.45 ± 2.09
Sf	31.37 ± 0.00	39.57 ± 0.00	27.27 ± 0.89	28.83 ± 2.63	32.61 ± 2.11	32.93 ± 2.32	25.78 ± 1.98	**89.72 ± 1.68**
Tt	39.52 ± 0.00	54.84 ± 0.18	**63.60 ± 3.76**	50.90 ± 5.14	53.78 ± 3.71	36.39 ± 2.15	52.66 ± 2.60	57.20 ± 3.15

4.3 Ablation Study

We conduct ablations on the same 11 datasets to examine the two key components of CPC: CPA and RPD. `CPC_wo_CPA` removes cross-dataset joint prototype learning and learns an independent prototype space for each dataset; `CPC_wo_RPD` directly treats all activated prototypes as clusters; and `CPA+KMeans` replaces RPD with KMeans on the activated prototypes, using the ground-truth number of classes only for this diagnostic comparison. All variants use the same 20-run protocol, with complete results reported in Appendix Tables 4, 5, and 6.

Cross-Dataset Prototype Alignment. As shown in Table 4, CPC outperforms `CPC_wo_CPA` on all 11 datasets and all four metrics, yielding an average ACC improvement of 6.93 percentage points. For example, ACC on Pt increases from 74.56% to 82.76%. This consistent improvement sup-

Table 3: Number of ground-truth classes, activated prototypes, and representative prototypes per dataset.

Dataset	Classes	Activated Prototypes	Representative Prototypes
Bs	3	12	3
Ca	2	21	3
Hr	3	15	3
Hd	5	30	6
Ls	3	15	3
Lc	3	12	3
Mm	2	14	3
Pt	21	29	22
Ps	2	22	3
Sf	6	28	6
Tt	2	34	4

ports the role of the shared prototype space: jointly optimizing the prototype space across heterogeneous datasets allows different datasets to activate common prototypes and thereby provides implicit prototype-level structural correspondence.

Representative Prototype Discovery. Removing RPD exposes the redundancy of the shared prototype space. `CPC_wo_RPD` can remain competitive on several datasets, but directly treating every activated prototype as a cluster can substantially fragment the final structure; on Pt, ACC drops from 82.76% to 43.66%. The prototype statistics in Table 3 further show that activated prototypes are generally more numerous than ground-truth classes (e.g., 28 vs. 6 on Sf and 34 vs. 2 on Tt). RPD therefore serves as a prototype-level selection stage that reduces redundancy and extracts representative prototypes as compact cluster anchors.

Why not conventional prototype clustering? Replacing RPD with KMeans leads to substantial degradation on most datasets, even when the ground-truth cluster number is provided. For instance, ACC decreases from 90.32% to 31.59% on Sf and from 82.76% to 24.97% on Pt. This comparison indicates that the final clustering structure cannot be obtained by simply partitioning the shared prototypes; RPD explicitly models prototype interactions and learns representative probabilities to select structurally informative anchors.

4.4 Shared Prototype Space Size Sensitivity

The shared space determines the capacity of the prototype space used to capture recurring structures across heterogeneous datasets. We therefore examine the effect of K on four representative datasets (Bs, Mm, Ps, Tt) with $K \in \{32, 48, 64, 96, 128\}$, while keeping all other settings fixed.

As shown in Fig. 2, CPC remains relatively stable across a broad range of space sizes, with $K = 64$ providing a consistently strong operating point. When K is too small, the shared prototype space has limited capacity to represent the diverse structural patterns contributed by different datasets, which can cause multiple patterns to be represented by overly coarse prototypes. Increasing K initially improves or preserves performance by providing finer structural coverage. However, further increasing the codebook introduces additional prototypes that are not necessarily supported by distinct structures and may increase prototype redundancy, leading to performance degradation on some datasets such as Bs and Mm.

This behavior is consistent with the role of the codebook in CPC: it is designed as a shared structural vocabulary rather than a direct estimate of the number of clusters. The prototype space therefore needs sufficient capacity to cover heterogeneous structures, while excessive capacity can make the prototype space unnecessarily redundant. The relatively stable performance around $K = 64$ suggests that CPC does not depend on a narrowly tuned codebook size. We consequently use $K = 64$ as the dataset-independent default in all experiments.

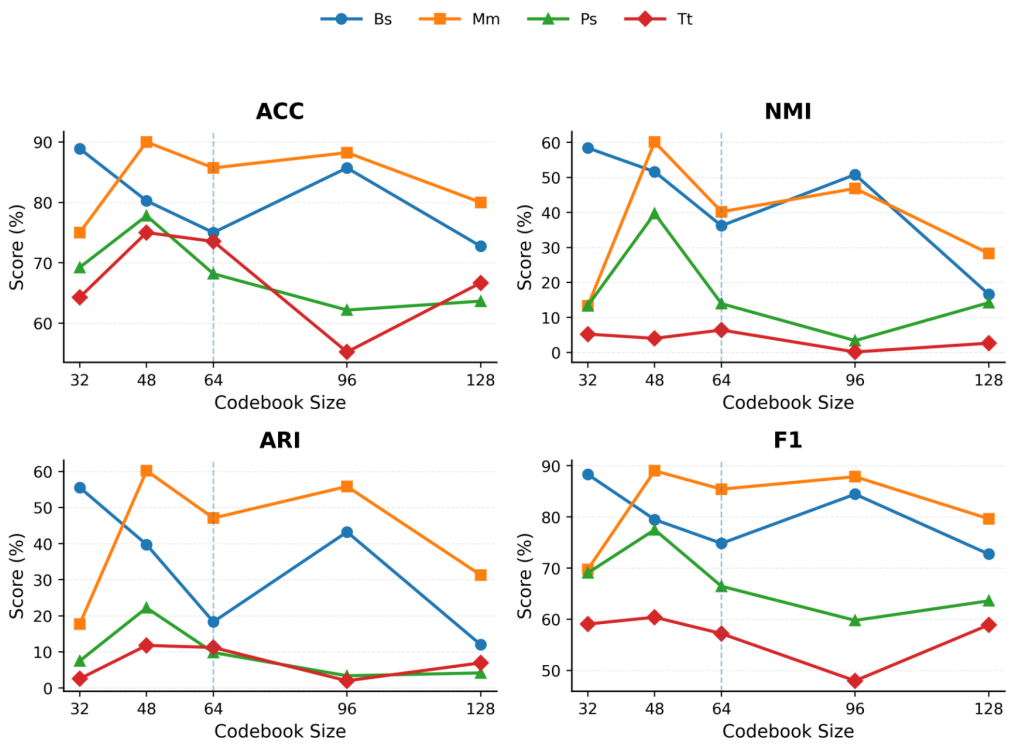

Figure 2: Sensitivity of CPC to the shared prototype space size K on four representative datasets.

5 Conclusion

We presented CPC, a general domain-adaptive clustering approach based on cross-dataset prototype alignment. CPC jointly learns a shared prototype space from multiple heterogeneous datasets rather than relying on a designated source–target relationship. Through its two-stage *Learn–Select* framework, CPA maps heterogeneous data into a common representation space and learns a shared codebook, allowing different datasets to activate common prototypes and establish implicit prototype-level structural correspondence. RPD then models interactions among activated prototypes and selects a compact subset as dataset-specific cluster anchors.

Experiments on 11 heterogeneous real-world datasets show that CPC achieves the best or tied-best ACC on 9 datasets and substantially improves over existing clustering methods on several challenging datasets. Ablation studies further demonstrate the complementary roles of the two stages: removing CPA consistently weakens performance, while removing RPD exposes the redundancy of the activated prototype space. Replacing RPD with conventional KMeans also leads to substantial degradation, indicating that shared prototype learning and representative prototype discovery address distinct aspects of clustering. Overall, the results support the use of shared prototype spaces as a means of capturing structural regularities across heterogeneous datasets while retaining dataset-specific cluster discovery.

CPC also has limitations. Its shared prototype space benefits from the presence of reusable structural patterns across the jointly trained datasets, and the representative selection process still involves manually specified hyperparameters. Future work will investigate adaptive construction of the shared prototype space and more flexible representative selection mechanisms to reduce these dependencies.

AI use statement

We have not used generative AI tools in this work. We take responsibility for the final content of this work.

Reproducibility statement

The code related to the paper is now available on GitHub, and we have also submitted the experimental code as an attachment.

Acknowledgments

The author thanks all the reviewers and area chairs for their efforts.

References

Hossein Abbasimehr and Ali Noshad. Localized global models using autoencoder-based clustering to forecast related time series. *International Journal of Data Science and Analytics*, 20(4):4067–4088, 2025.

Mashaan Alshammari, John Stavrakakis, and Masahiro Takatsuka. A parameter-free graph reduction for spectral clustering and spectralnet. *Array*, 15:100192, 2022.

Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. Efficient deep embedded subspace clustering. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 21–30. IEEE, 2022.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European conference on computer vision*, pp. 139–156. Springer, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

D Dua and C Graff. Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california, school of information and computer science. *IEEE transactions on pattern analysis and machine intelligence*, 1(1):1–29, 2019.

Delbert Dueck and Brendan J Frey. Non-metric affinity propagation for unsupervised image categorization. In *2007 IEEE 11th international conference on computer vision*, pp. 1–8. IEEE, 2007.

Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Ijcai*, volume 17, pp. 1753–1759, 2017.

Xiaofei He, Deng Cai, Yuanlong Shao, Hujun Bao, and Jiawei Han. Laplacian regularized gaussian mixture model for data clustering. *IEEE transactions on knowledge and data engineering*, 23(9): 1406–1418, 2010.

Zhizhong Huang, Jie Chen, Junping Zhang, and Hongming Shan. Learning representation for clustering via prototype scattering and positive sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7509–7524, 2022.

Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: A generative approach to clustering. *CoRR, abs/1611.05148*, 1, 2016.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Shuo Li, Fang Liu, Licheng Jiao, Puhua Chen, and Lingling Li. Self-supervised self-organizing clustering network: A novel unsupervised representation learning method. *IEEE transactions on neural networks and learning systems*, 35(2):1857–1871, 2022.

Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8547–8555, 2021.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

Yuwu Lu, Haoyu Huang, Wai Keung Wong, Xue Hu, Zhihui Lai, and Xuelong Li. Adaptive dispersal and collaborative clustering for few-shot unsupervised domain adaptation. *IEEE Transactions on Image Processing*, 2025.

Zihe Lv, Zhengda Wu, and Jinghua Zhu. Clustering-guided contrastive prototype learning: Towards semi-supervised medical image segmentation. *Pattern Recognition*, pp. 112321, 2025.

Louis Mahon and Thomas Lukasiewicz. Efficient deep clustering of human activities and how to improve evaluation. In *Asian Conference on Machine Learning*, pp. 722–737. PMLR, 2023.

I Made Putrama and Péter Martinek. Heterogeneous data integration: Challenges and opportunities. *Data in Brief*, 56:110853, 2024.

Chendong Qin, Yongxiong Wang, Fubin Zeng, Jiapeng Zhang, Yangsen Cao, Xiaolan Yin, Shuai Huang, Di Chen, Huojun Zhang, and Zhiyong Ju. Active domain adaptation based on probabilistic fuzzy c-means clustering for pancreatic tumor segmentation. *IEEE Transactions on Fuzzy Systems*, 33(6):2016–2026, 2025.

Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019.

Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S Yu, and Lifang He. Deep clustering: A comprehensive survey. *IEEE transactions on neural networks and learning systems*, 36(4):5858–5878, 2024.

Amelie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. Scalarization for multi-task and multi-domain learning at scale. *Advances in Neural Information Processing Systems*, 36: 16917–16941, 2023.

Peeyush Singhal, Rahee Walambe, Sheela Ramanna, and Ketan Kotecha. Domain adaptation: challenges, methods, datasets, and applications. *IEEE access*, 11:6973–7020, 2023.

Douglas Steinley. K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1):1–34, 2006.

Georgios Vardakas, Ioannis Papakostas, and Aristidis Likas. Deep clustering using the soft silhouette score: Towards compact and well-separated clusters. *Machine Learning*, 115(4):81, 2026.

Nian Wang, Zhigao Cui, Aihua Li, Yihang Lu, Rong Wang, and Feiping Nie. Structured doubly stochastic graph-based clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11064–11077, 2025.

Xiang Wang, Liping Jing, Huafeng Liu, and Jian Yu. Structure-driven representation learning for deep clustering. *ACM Transactions on Knowledge Discovery from Data*, 18(1):1–25, 2023.

Viola Wenz, Arno Kesper, and Gabriele Taentzer. Clustering heterogeneous data values for data quality analysis. *ACM Journal of Data and Information Quality*, 15(3):1–33, 2023.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487. PMLR, 2016.

Hui Yin, Amir Aryani, Stephen Petrie, Aishwarya Nambissan, Aland Astudillo, and Shengyuan Cao. A rapid review of clustering algorithms. *Array*, pp. 100904, 2026.

Xingyu Zhu, Beier Zhu, Yunfan Li, Junfeng Fang, Shuo Wang, Kesen Zhao, and Hanwang Zhang. Hierarchical semantic alignment for image clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29177–29185, 2026.

A Additional Figures

The overall architectures of the two stages of CPC are illustrated in Figs. 3 and 4.

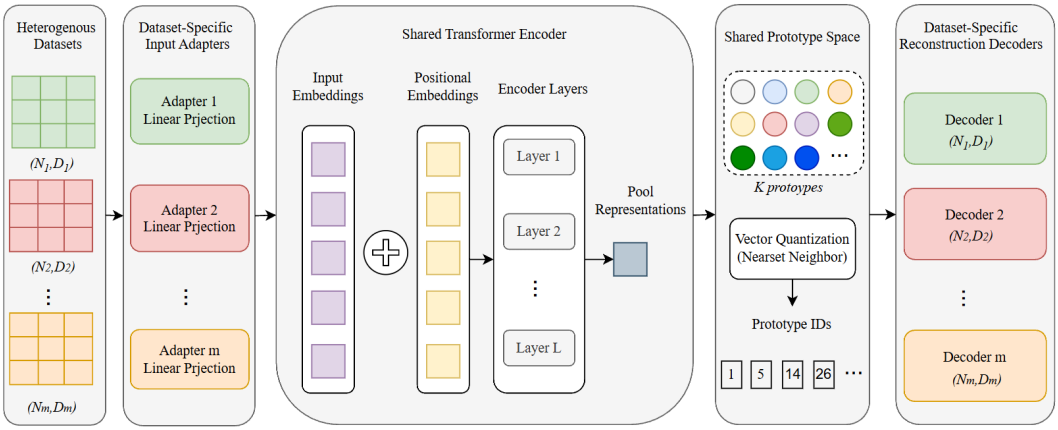

Figure 3: Overview of the Cross-Dataset Prototype Alignment module. Heterogeneous datasets are projected into a shared latent space through dataset-specific adapters and a shared Transformer encoder, followed by prototype quantization and reconstruction.

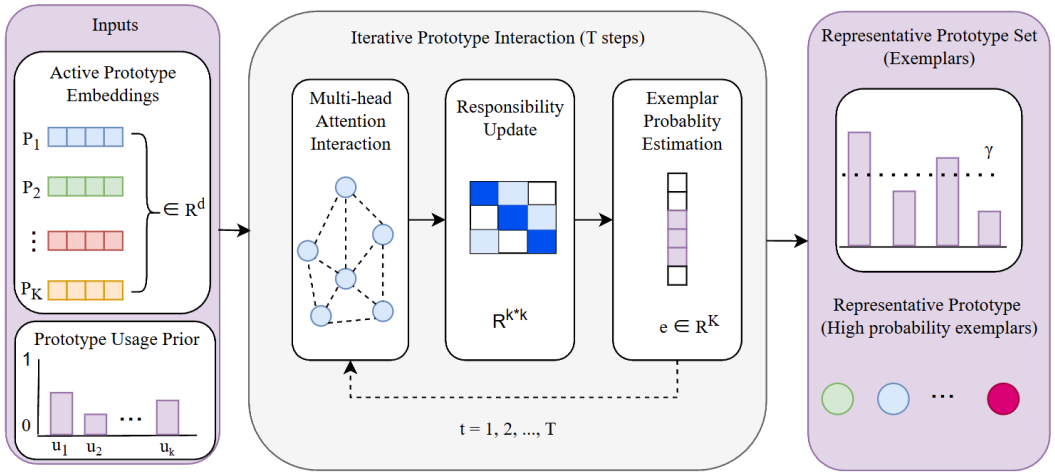

Figure 4: Overview of the RPD module. The module iteratively models prototype interactions through multi-head attention and predicts representative prototypes for clustering.

B Computational Complexity

Taking the Bs dataset as an example, the trainable parameter counts are 128 for the dataset-specific adapter, 8,836 for the decoder, 4,096 for the shared codebook, 132,864 for the shared encoder, and 25,282 for RPD, totaling approximately 171.2K parameters. The shared encoder and codebook account for the majority of the parameters and are jointly learned across datasets, while dataset-specific components add only a small overhead. During the RPD stage the shared space is fixed and only the lightweight RPD module is optimized.

C Full Ablation Results

Tables 4, 5, and 6 report the complete ablation results over 20 runs (mean $\pm$ std).

Table 4: Ablation results of Cross-Dataset Prototype Alignment. All values are reported as mean ± standard deviation over 20 runs.

Dataset	CPC				CPC_wo_CPA			
	ACC	NMI	ARI	F1	ACC	NMI	ARI	F1
Bs	75.00 ± 2.81	36.20 ± 2.37	18.33 ± 2.15	74.83 ± 2.06	68.42 ± 3.12	30.15 ± 2.89	14.87 ± 2.43	68.91 ± 2.54
Ca	76.19 ± 1.47	32.91 ± 1.04	24.57 ± 1.83	72.15 ± 1.38	69.83 ± 1.92	26.74 ± 1.56	19.33 ± 2.07	65.47 ± 1.87
Hr	46.67 ± 1.56	11.27 ± 2.49	6.57 ± 1.50	38.66 ± 1.07	41.23 ± 2.01	8.56 ± 2.13	4.91 ± 1.82	33.82 ± 1.56
Hd	63.33 ± 0.85	20.05 ± 1.69	21.29 ± 0.65	32.75 ± 0.91	57.89 ± 1.24	15.32 ± 2.01	16.78 ± 0.92	28.13 ± 1.23
Ls	80.00 ± 2.53	34.96 ± 4.65	37.51 ± 2.25	54.74 ± 1.77	72.15 ± 3.01	27.43 ± 5.12	30.22 ± 2.89	48.56 ± 2.11
Lc	56.25 ± 3.34	27.95 ± 1.43	11.45 ± 1.76	53.09 ± 2.16	49.87 ± 3.78	21.09 ± 1.89	8.73 ± 2.13	46.32 ± 2.54
Mm	85.71 ± 1.45	40.20 ± 1.94	47.09 ± 2.56	85.42 ± 2.79	78.34 ± 1.98	33.67 ± 2.45	39.85 ± 2.98	78.91 ± 3.12
Pt	82.76 ± 2.49	74.68 ± 0.78	65.97 ± 1.19	75.31 ± 3.34	74.56 ± 2.87	66.23 ± 1.12	57.44 ± 1.56	67.82 ± 3.67
Ps	68.18 ± 3.16	13.94 ± 3.81	9.82 ± 0.93	66.45 ± 2.09	60.12 ± 3.45	10.78 ± 4.02	7.15 ± 1.24	58.73 ± 2.45
Sf	90.32 ± 2.52	78.61 ± 0.49	80.35 ± 3.24	89.72 ± 1.68	83.47 ± 2.98	70.12 ± 0.78	72.89 ± 3.67	82.15 ± 2.01
Tt	73.53 ± 2.40	6.41 ± 1.26	11.22 ± 0.92	57.20 ± 3.15	65.78 ± 2.67	4.89 ± 1.54	8.56 ± 1.13	50.34 ± 3.34

Table 5: Ablation results of Representative Prototype Discovery. All values are reported as mean ± standard deviation over 20 runs.

Dataset	ACC	NMI	ARI	F1
Bs	76.88 ± 1.07	38.19 ± 0.94	20.64 ± 0.77	76.34 ± 0.72
Ca	74.04 ± 1.35	30.10 ± 1.08	21.69 ± 0.85	70.11 ± 0.93
Hr	65.37 ± 1.12	24.63 ± 0.79	8.84 ± 0.72	67.20 ± 1.08
Hd	62.21 ± 1.04	18.27 ± 0.65	19.40 ± 0.73	31.40 ± 0.89
Ls	82.56 ± 1.14	38.35 ± 1.23	40.65 ± 1.19	57.64 ± 1.03
Lc	63.20 ± 1.20	32.36 ± 1.24	15.93 ± 1.21	59.82 ± 0.95
Mm	83.24 ± 0.88	38.10 ± 0.61	45.17 ± 0.54	83.28 ± 0.71
Pt	43.66 ± 0.73	35.84 ± 0.82	5.36 ± 0.68	18.16 ± 0.57
Ps	71.32 ± 1.16	16.51 ± 0.89	12.16 ± 0.79	69.47 ± 0.95
Sf	92.92 ± 0.87	81.26 ± 0.92	83.64 ± 0.93	92.58 ± 0.74
Tt	73.47 ± 1.06	6.35 ± 0.58	10.55 ± 0.42	66.68 ± 0.86

Table 6: Clustering results obtained by applying KMeans to the shared prototype space. The ground-truth number of classes is used as the KMeans cluster number. All values are reported as mean ± standard deviation over 20 runs.

Dataset	ACC	NMI	ARI	F1
Bs	65.28 ± 0.00	31.29 ± 0.00	29.40 ± 0.00	47.09 ± 0.00
Ca	56.29 ± 1.45	2.44 ± 0.67	1.41 ± 0.71	55.35 ± 1.72
Hr	40.04 ± 3.18	2.14 ± 1.61	0.38 ± 1.20	38.57 ± 3.11
Hd	38.11 ± 5.72	6.95 ± 1.14	9.31 ± 4.24	25.76 ± 2.23
Ls	58.75 ± 4.25	27.78 ± 2.26	15.18 ± 5.88	53.44 ± 6.74
Lc	52.03 ± 4.09	15.98 ± 4.42	10.71 ± 3.64	49.42 ± 5.72
Mm	62.82 ± 10.85	8.79 ± 10.43	10.95 ± 14.02	57.74 ± 14.63
Pt	24.97 ± 0.77	28.14 ± 0.74	5.18 ± 0.81	16.15 ± 0.75
Ps	55.42 ± 3.49	1.21 ± 1.25	0.73 ± 1.71	55.04 ± 3.70
Sf	31.59 ± 5.22	3.41 ± 0.84	1.71 ± 1.15	20.24 ± 2.35
Tt	56.74 ± 0.21	1.29 ± 0.02	1.71 ± 0.11	55.46 ± 0.10