MC-H: Multi-Granularity Clustering with Hyperspherical Determinantal Point Process

Yingjie Ge[1], **Kun Zhou**[2,3], **Jieyu Zhao**[1], **Xulun Ye**[1,†]
[1] School of Artificial Intelligence, Ningbo University, Ningbo, China
[2] School of Architecture & Urban Planning, Shenzhen University, Shenzhen, China
[3]Intelligent Computing Center, Shenzhen University
suyiz0910@gmail.com,zhoukun@szu.edu.cn,{zhao_jieyu,yexulun}@nbu.edu.cn

Abstract

Deep clustering struggles when coarse-grained and fine-grained semantics coexist in the same dataset, as a unified partitioning strategy tends to either over-separate coarse categories or miss subtle fine-grained differences. To address this problem, we propose the adaptive multi-granularity clustering framework in a hyperspherical space (MC-H). The framework combines Dynamic Prototype-Center Assignment (DPCA), a Hyperspherical Determinantal Point Process (H-DPP) module that adaptively regulates repulsion according to local structure and promotes stronger separation in dense regions while preserving flexibility in sparse ones, and a Semantic Subspace Denoising (SSD) module that suppresses non-semantic variations and improves cluster compactness. Extensive experiments across datasets with different semantic granularities show that our method consistently outperforms existing approaches and effectively unifies clustering across the full granularity spectrum. **Code is available at** https://github.com/suyifengchentan/MC-H.

1 Introduction

Deep clustering aims to jointly learn feature representations and cluster assignments in a fully unsupervised manner, and has achieved strong performance on standard benchmarks [51, 3, 47, 53]. Most existing studies [7, 19, 17, 28, 18, 52, 31, 6], however, implicitly assume that the data follows a single semantic granularity. In real-world visual data, this assumption is often violated: some categories are separated at a coarse level with clear semantic boundaries, whereas others differ only at a fine-grained level with subtle visual variations. More importantly, these different granularities may coexist within the same dataset, yielding heterogeneous local geometry in the representation space. This creates a fundamental challenge for deep clustering: a partitioning strategy that preserves broad semantic categories may fail to resolve subtle subcategory differences, while one that separates fine-grained categories well may over-fragment coarse-grained structure. Consequently, when the number of data categories is unknown a priori, clustering methods face an even greater challenge because they cannot determine the appropriate granularity level or anticipate the coexistence of coarse and fine structures without explicit supervision.

Existing methods are not designed for this mixed-granularity setting. Coarse-grained clustering approaches [27, 47, 20] typically rely on globally uniform partitioning mechanisms, which are effective when categories are well separated in feature space and the clustering geometry is relatively simple. Under such conditions, moderate and globally shared repulsion is often sufficient to partition data without introducing spurious fragmentation, which partly explains the strong empirical performance of deep clustering on standard benchmarks [9]. Fine-grained clustering methods [1, 29, 46, 53], in contrast, address settings where semantically similar subcategories are densely packed and often

†Corresponding author.

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

partially overlapping. Their entangled geometry and small inter-class margins demand significantly stronger separation to resolve subtle distinctions. However, such aggressive separation strategies can easily damage coarse-grained semantic structure when applied beyond the fine-grained regime. As a result, current methods are largely specialized to one granularity regime or the other, and their effectiveness often depends on prior knowledge of the dataset structure for model selection or hyperparameter tuning. At a deeper level, their limitation stems from applying a fixed clustering strategy to data whose local semantic geometry varies substantially across regions.

To address this issue, we propose the MC-H, an adaptive multi-granularity deep clustering framework in a shared hyperspherical space. The key idea is to replace globally uniform repulsion with a geometry-aware mechanism that adapts to local semantic density. Specifically, the framework combines Dynamic Prototype-Center Assignment (DPCA), a Hyperspherical Determinantal Point Process (H-DPP) module, and a Semantic Subspace Denoising (SSD) module. DPCA provides quality-aware soft assignments that stabilize optimization and support the hyperspherical formulation. The H-DPP module regulates repulsion according to local structure, enforcing stronger separation in dense regions while preserving flexibility in sparse ones. This allows the clustering objective to simultaneously accommodate fine-grained distinctions and coarse-grained integrity within a unified formulation. Additionally, the eigenspectrum of the H-DPP module provides a natural, data-driven spectral proxy for the effective number of occupied prototype components, which we use for dynamic prototype allocation when the cluster number K is unknown. SSD further improves intra-cluster consistency by suppressing non-semantic variations while preserving semantically meaningful structure. Together, these components establish a unified clustering mechanism for heterogeneous semantic granularities. Extensive experiments on coarse-grained, mixed-grained, and fine-grained benchmarks demonstrate that our method consistently outperforms existing approaches and effectively unifies clustering across the full granularity spectrum.

Our contributions are summarized as follows:

- We propose a unified adaptive multi-granularity deep clustering framework that addresses the coexistence of coarse-grained and fine-grained semantics within a shared hyperspherical space, providing a single solution for clustering across heterogeneous semantic granularities.
- Within the MC-H framework, DPCA provides quality-aware assignments, H-DPP adaptively regulates repulsion according to local structure, and SSD suppresses non-semantic variations to improve cluster compactness.
- Extensive experiments on coarse-grained, mixed-grained, and fine-grained benchmarks demonstrate that our method consistently outperforms existing approaches and effectively unifies clustering across the full semantic granularity spectrum.

2 Related Work

Deep Clustering. Early deep clustering methods jointly optimize representation learning and clustering via reconstruction, pseudo-label refinement, or mutual-information maximization [51, 7, 19, 17]. With the advent of self-supervised learning, subsequent approaches partially decouple feature learning from clustering through contrastive pretraining, nearest-neighbor consistency, and online cluster prediction [9, 47, 27, 28]. Prototype-based methods further make cluster structures explicit, e.g., SwAV, PCL, SPICE, and ProPos [8, 26, 38, 18], while recent works improve efficiency and assignment stability via aggregation strategies [52]. To capture more complex structures, some methods explore hyperspherical clustering with nonparametric models or prototype nodes [43, 31], while others incorporate hierarchical or multi-granularity learning [15, 50]. However, most existing approaches assume a fixed semantic scale or rely on globally specified objectives (e.g., temperature or prototype structure), limiting their ability to adapt to heterogeneous local densities. More recent works leverage large pretrained models for stronger semantic priors. TEMI highlights the effectiveness of frozen encoders, while CLIP-based methods incorporate textual semantics or cross-modal alignment [1, 40, 6, 29, 30, 20, 53]. Diffusion-based clustering has also emerged as an alternative formulation [46]. Although these methods improve semantic discriminability, they still face a fundamental trade-off between coarse-grained over-segmentation and fine-grained under-separation, especially under low inter-class margins and high intra-class variance [37, 39]. Moreover, most frameworks assume a pre-defined cluster number K, whereas real-world scenarios require data-driven cardinality estimation [41, 44, 36, 48].

Deep Determinantal Point Processes. Determinantal Point Processes (DPPs) model diversity and negative dependence via determinants that capture both item quality and pairwise repulsion [24]. Classical work focuses on learning and inference, including fixed-point optimization, Nyström approximation, and efficient sampling [33, 2, 25]. In deep learning, DPPs are typically used as auxiliary mechanisms for discrete subset selection or diversity control, such as document summarization, network compression, coreset construction, and k-means seeding [10, 34, 45, 5, 4]. In generative modeling, DPP-inspired objectives mitigate mode collapse and improve diversity [14, 35, 21], and have been extended to temporal mixture modeling, retrieval refinement, and discrete diffusion decoding [13, 42, 32]. However, existing uses treat DPPs as auxiliary components—samplers, priors, or regularizers—rather than as differentiable objectives that directly shape representation geometry. H-DPP addresses this limitation by promoting the log-determinant to a density-aware, assignment-weighted regularizer on the hypersphere, extending DPPs from discrete subset selection to continuous feature learning. Its eigenspectrum further provides a practical proxy for adaptive prototype cardinality when K is unknown.

3 Methodology

3.1 Problem Formulation and Overview

Problem Formulation: Given an unlabeled dataset $\mathcal{X} = \{x_i\}_{i=1}^N$, whose category structure may span multiple semantic granularities (with coarse- and fine-grained classes coexisting), and where the number of categories K is unknown a priori, our goal is to learn a unified clustering objective that can simultaneously model multi-granularity structures and enable adaptive estimation of a geometry-derived prototype cardinality proxy without requiring prior knowledge of K.

Overview: Our proposed multi-granularity clustering framework consists of three core modules, as illustrated in Fig. 1, forming a unified trainable clustering pipeline built on frozen CLIP features: Dynamic Prototype-Center Assignment (DPCA, Sec. 3.3). Taking frozen CLIP features as input, this module constructs assignment-aware prototype structures and maps samples of different granularities into a common hyperspherical space. Its quality head supplies sample weights for H-DPP, whose spectrum provides the assignment anchors. H-DPP (with Dynamic-K, Sec. 3.4). H-DPP constructs spectral anchors from the current quality-weighted representation and applies a log-determinant diversity constraint scaled by batch angular concentration. Meanwhile, its eigenspectrum supports adaptive estimation of an effective prototype cardinality proxy when K is unknown. SSD (Sec. 3.5). Building upon H-DPP, this module further refines cluster structures by anchoring samples to semantic subspaces and suppressing variance along non-semantic directions, thereby enhancing intra-cluster compactness and sharpening inter-cluster boundaries.

3.2 Hyperspherical Representation Learning with Frozen Semantic Priors

We leverage a pre-trained, frozen CLIP image encoder $g(\cdot)$ [40] to provide a robust semantic foundation. To adapt these general-purpose embeddings to the nuances of multi-granularity clustering, we introduce a learnable projection head $h_\phi(\cdot)$ that maps the backbone features onto a d-dimensional unit hypersphere $\mathcal{S}^{d-1}$. For an input x_i, we generate two stochastic augmentations x_i^a, x_i^b and compute adapted representations $\mathbf{z}_i^v = h_\phi(g(x_i^v))/\|h_\phi(g(x_i^v))\|_2 \in \mathcal{S}^{d-1}$ for $v \in \{a, b\}$. Here, *frozen semantic priors* means the pretrained image features $g(x_i^v)$, whose encoder parameters remain fixed throughout clustering. SSD uses these features to anchor its semantic subspaces. To ensure instance-level view invariance, we adopt a von Mises–Fisher (vMF)-inspired alignment–uniformity objective on the hypersphere. This objective combines an alignment term that pulls positive pairs together along the sphere with a log-sum-exp uniformity term that discourages representation collapse:

$$\mathcal{L}_{\text{inv}} = -\frac{1}{n\tau_{\text{inv}}} \sum_{i=1}^{n} \langle \mathbf{z}_i^a, \mathbf{z}_i^b \rangle + \frac{1}{2n} \sum_{i=1}^{n} \Big[\log \sum_{j=1}^{n} e^{\langle \mathbf{z}_i^a, \mathbf{z}_j^b \rangle / \tau_{\text{inv}}} + \log \sum_{j=1}^{n} e^{\langle \mathbf{z}_i^b, \mathbf{z}_j^a \rangle / \tau_{\text{inv}}} \Big], \tag{1}$$

where τ_{inv} is a temperature hyperparameter. This objective preserves the angular discrimination required by the subsequent prototype and density-adaptive H-DPP modules.

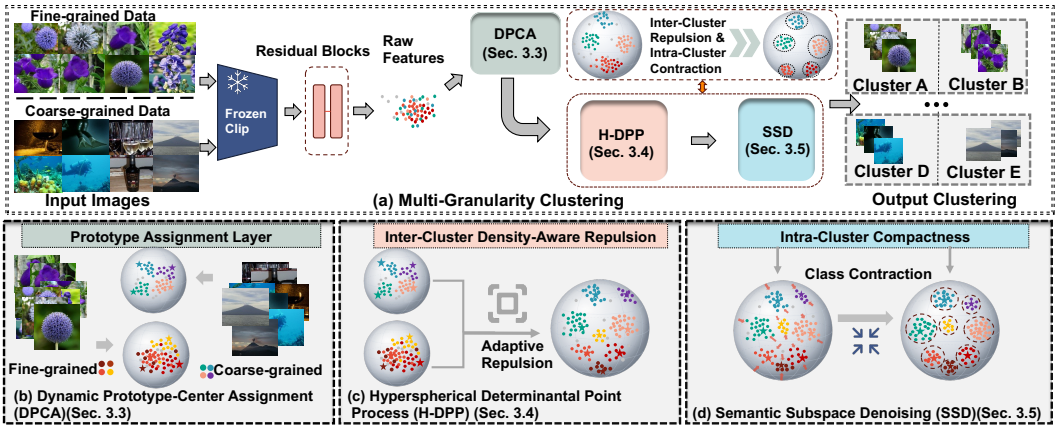


Figure 1: Framework overview. **(a) Overall multi-granularity clustering pipeline:** input images are encoded by frozen CLIP features, and then refined through DPCA, H-DPP, and SSD before producing the final clustering assignments. **(b) DPCA:** the prototype assignment layer converts the current representation into dynamic fine- and coarse-grained prototype distributions; five-pointed stars denote the soft cluster centers computed by DPCA. **(c) H-DPP:** quality-weighted spectral expansion uses a batch-level adaptive coefficient. **(d) SSD:** semantic subspace denoising suppresses covariance unsupported by frozen semantic features.

3.3 Dynamic Prototype-Center Assignment (DPCA)

DPCA acts as the partition interface between hyperspherical representations and the H-DPP geometry. It assigns each sample a relative confidence level and converts the current geometric structure into soft prototype-based cluster distributions. In this way, the partition is not fixed a priori, but induced by the geometry currently supported by the representation space.

Quality Weight Prediction. For each mini-batch, a learnable quality head f_ψ predicts the scalar log-quality score $r_i^v = f_\psi(\mathbf{z}_i^v)$ for sample i and view v. We center them within the batch and convert them into positive relative qualities $w_i^v = \exp\big(r_i^v - \frac{1}{n}\sum_{m=1}^{n} r_m^v\big)$, so that $w_i^v > 0$. These quality weights are passed to the H-DPP core to construct the quality-weighted kernel used for density-aware repulsion and dynamic-K estimation.

Soft Assignment to Dynamic Prototype Centers. As detailed in Section 3.4, the H-DPP spectrum yields a geometry-derived proxy cardinality $\hat{K}$ and a corresponding set of spectral prototypes $V = [\mathbf{v}_1, \ldots, \mathbf{v}_{\hat{K}}]^\top \in \mathbb{R}^{\hat{K}\times d}$. DPCA treats these prototypes as dynamic assignment anchors and computes logits $\ell_{ik}^v = \langle \mathbf{z}_i^v, \mathbf{v}_k \rangle / \tau_{\text{assign}}$ together with soft assignments $\mathbf{q}_i^v = \text{softmax}(\boldsymbol{\ell}_i^v)$ for $k = 1, \ldots, \hat{K}$. Thus, the assignment dimension is determined by the current H-DPP spectrum rather than by a pre-specified cluster count.

Symmetrized Cross-View Partition Consistency. To ensure that the induced latent partition is stable under stochastic augmentations, we enforce consistency between the view-specific prototype distributions through a symmetrized predictive alignment, which is used implicitly inside the DPCA refinement. Let $\sigma_\tau(\cdot)$ denote the softmax operator with temperature τ, and define $\tilde{\mathbf{q}}_i^v = \text{sg}[\sigma_{\tau_{\text{sharp}}}(\boldsymbol{\ell}_i^v)]$ as the sharpened assignment target for view $v \in \{a, b\}$, where $\text{sg}(\cdot)$ is the stop-gradient operation. The corresponding cross-view consistency quantity is computed as

$$\frac{1}{2n}\sum_{i=1}^{n}\sum_{(u,v)\in\mathcal{V}_{\text{pair}}} H\left(\tilde{\mathbf{q}}_i^u, \sigma_{\tau_{\text{part}}}(\boldsymbol{\ell}_i^v)\right). \tag{2}$$

where $\mathcal{V}_{\text{pair}} = \{(a,b),(b,a)\}$ denotes the set of cross-view pairs, and $H(p,q) = -p^\top \log q$ is the cross-entropy functional. DPCA uses this quantity for cross-view assignment consistency within its forward refinement. It is not an additional loss term in Equation 13; the backpropagated objective contains only invariance, H-DPP, and SSD.

Dynamic Refinement of Geometry and Assignments. Each iteration first constructs H-DPP anchors from current embeddings and quality weights, matches and sign-aligns them to an EMA prototype bank, and then computes DPCA assignments. Detached assignments route frozen features to SSD queues. The optimizer updates the projection and quality heads, and the next iteration recomputes the spectrum from the updated representation. This is a dependency across iterations, with no inner fixed-point loop. Algorithm 1 gives the complete update order.

3.4 H-DPP: Density-Aware Repulsion and Dynamic Prototype Cardinality

Building on the quality-weighted representations, H-DPP is the main inter-cluster geometry module in our framework. It turns the log-determinant objective of a hyperspherical DPP kernel into a differentiable criterion that expands the occupied spectral volume of the feature set. H-DPP combines quality-weighted pairwise geometry with a batch-level coefficient that increases with angular concentration. The coefficient is shared by the batch; it does not estimate a separate local density for each sample. In addition, the eigenspectrum of the same kernel provides a natural spectral extension for dynamic prototype cardinality when K is unknown.

Hyperspherical Kernel Construction. Let $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n]^\top \in \mathbb{R}^{n\times d}$ be the adapted representations and $\mathbf{w} \in \mathbb{R}^n$ be the corresponding positive relative quality weights produced by DPCA (view indices omitted). We define a quality-weighted feature matrix $\Phi = \mathrm{diag}(\tilde{\mathbf{w}})\, Z$ through a centered-log reparameterization, with $\tilde{w}_i = \exp\left(\log w_i - \frac{1}{n}\sum_{m=1}^{n} \log w_m\right)$. This reparameterization preserves the relative quality ratios while ensuring numerical stability across various scales. The resulting H-DPP kernel is $L = \Phi\Phi^\top \in \mathbb{R}^{n\times n}$, where $L_{ij} = \tilde{w}_i\tilde{w}_j\langle \mathbf{z}_i, \mathbf{z}_j\rangle$. This kernel intrinsically couples semantic confidence with pairwise geometric similarity, providing a principled basis for adaptive repulsion.

Log-Determinant Repulsion Loss. We instantiate DPP-based repulsion by minimizing the negative log-determinant of the quality-weighted kernel:

$$\mathcal{L}_{\mathrm{dpp}} = -\log\det((1+\epsilon)I_n + L), \tag{3}$$

where $\epsilon > 0$ is a small jitter for positive definiteness. Maximizing the log-determinant enlarges the volume spanned by the quality-weighted features and promotes diverse, non-collapsed directions in the representation space. When the feature dimension is smaller than the kernel size ($d < n$), Sylvester's determinant identity yields the equivalent feature-space form

$$\log\det((1+\epsilon)I_n + \Phi\Phi^\top) = \log\det((1+\epsilon)I_d + \Phi^\top\Phi) + (n-d)\log(1+\epsilon). \tag{4}$$

Thus the same objective can be evaluated from the feature-space Gram matrix without changing the induced spectral volume.

Proposition 3.1 (Spectral Volume Expansion)**.** *Let $G = (1+\epsilon)I_n + \Phi\Phi^\top$. The gradient of $\mathcal{L}_{\mathrm{dpp}} = -\log\det G$ with respect to the quality-weighted feature matrix Φ is*

$$\frac{\partial\mathcal{L}_{\mathrm{dpp}}}{\partial\Phi} = -2G^{-1}\Phi. \tag{5}$$

Equivalently, if $\Phi = U\Sigma V^\top$ with singular values $\{\sigma_k\}$, then

$$\mathcal{L}_{\mathrm{dpp}} = -\sum_k \log(1+\epsilon+\sigma_k^2) + \mathrm{C}. \tag{6}$$

up to the zero singular directions. Hence the log-determinant term increases the spectral volume occupied by the quality-weighted features and favors diverse, non-collapsed feature directions. The density adaptivity of H-DPP is therefore obtained by combining this volume-expanding objective with the batch-level coefficient schedule in Equation 7, which assigns larger effective repulsion to batches whose features exhibit higher average similarity.

Density-Adaptive Coefficient Scheduling. To calibrate the global strength of this geometric expansion across mini-batches, we compute the mean pairwise cosine similarity within the batch as a density proxy, $\rho_t = \frac{1}{n(n-1)}\sum_{i\neq j}\langle \mathbf{z}_i, \mathbf{z}_j\rangle$, and use a smoothed running estimate $\bar{\rho}_t$ to define the effective H-DPP coefficient:

$$\alpha_{\mathrm{div}}^{(t)} = \gamma(t)\cdot\min(\max(\alpha_{\mathrm{base}}\cdot\mathrm{sg}(\bar{\rho}_t), 0), 1), \tag{7}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. The nonnegative clipping prevents a negative density estimate from reversing the log-determinant objective; when the smoothed density proxy is non-positive, H-DPP simply applies no additional repulsive pressure. Combined with the spectral-volume objective in Proposition 3.1, this coefficient makes the effective geometric pressure stronger in denser regimes and milder in sparser regimes. Here ρ_t is an angular-concentration proxy, not a probability-density estimate: larger mean cosine similarity indicates a more concentrated batch. It can be negative, so the schedule clips its contribution to be nonnegative. This scalar cannot resolve heterogeneous local densities within a batch. Detailed scheduling choices are deferred to Appendix A.2.

Dynamic Prototype Cardinality from the H-DPP Spectrum. When K is unknown, we use the H-DPP eigenspectrum for dynamic prototype allocation. The expected cardinality of a DPP sample is $\mathbb{E}[|Y|] = \sum_k \lambda_k^L/(1+\lambda_k^L)$ [16], which we use as a smooth spectral measure of how many prototype-supporting directions are effectively active in the current kernel. We therefore set

$$\hat{K} = \left\lfloor \sum_{k=1}^{n} \frac{\lambda_k^L}{1+\lambda_k^L} \right\rceil, \tag{8}$$

and lift the $\hat{K}$ leading sample-space eigenvectors into feature-space prototypes via the Nyström-style relation

$$\mathbf{v}_k = \frac{1}{\sqrt{\lambda_k^L}} \Phi^\top \mathbf{u}_k, \qquad k = 1, \ldots, \hat{K}, \tag{9}$$

Here $\lambda_k^L \geq 0$ and $\mathbf{u}_k$ are the eigenvalue and normalized eigenvector of L, respectively; the lift uses positive eigenvalues and gives $\|\mathbf{v}_k\| = 1$. Each $\lambda_k^L/(1+\lambda_k^L)$ contributes between zero and one to the expected cardinality. This is an effective-direction count, not a guarantee of recovering the annotated number of classes. The resulting prototype set $\{\mathbf{v}_k\}_{k=1}^{\hat{K}}$ serves as a compact collection of assignment anchors, allowing the model to adapt its prototype inventory without explicit supervision on K or a separate model-selection step. Numerical stabilization details are given in Appendix A.2.

3.5 Semantic Subspace Denoising for Intra-cluster Compactness (SSD)

While H-DPP shapes inter-cluster separation, intra-cluster variation must also be controlled to maintain semantic precision. In many cases, within-cluster variation is dominated by nuisance factors such as lighting, background, or pose, which should not define the cluster identity. SSD addresses this by associating each cluster with a semantically admissible subspace derived from frozen semantic priors, and by suppressing residual covariance outside that subspace.

Semantic Subspace Identification. Let $\{\mathbf{t}_i\}$ denote semantic prior features associated with cluster k, and let $\tilde{q}_{ik} = q_{ik}/\sum_j q_{jk}$ be the normalized confidence weights. We form the confidence-weighted second-moment matrix

$$M_k = \sum_i \tilde{q}_{ik}\, \mathbf{t}_i \mathbf{t}_i^\top \in \mathbb{R}^{d\times d}, \tag{10}$$

and obtain the dominant rank-r semantic subspace $U_k \in \mathbb{R}^{d\times r}$ as the top-r eigenvectors of M_k, with orthogonal projector $P_k = U_k U_k^\top$. Using $\tilde{q}_{ik}$ as weights ensures that the subspace is anchored to the most semantically representative samples. The priors $\mathbf{t}_i$ are frozen CLIP image features from a temporal queue, and the weights use detached DPCA assignments. Subspace estimation, student covariance computation, and residual penalization run in this order every N_{sub} iterations; SSD is zero on intervening steps. The queue is updated after every optimizer step. Details are given in Appendix A.3.

Granularity-Comparable Residual Space. Fine-grained and coarse-grained clusters naturally exhibit different intra-cluster scales: fine-grained clusters are compact but visually entangled, whereas coarse-grained clusters may preserve a broader semantic spread. To make the SSD constraint comparable across these regimes, we normalize each cluster covariance by its total energy and evaluate residual variation relative to the semantic projector P_k. Let $\mathbf{m}_k = \sum_i \tilde{q}_{ik}\mathbf{z}_i$ and define the assignment-weighted covariance

$$\bar{\Sigma}_k = \frac{\sum_i \tilde{q}_{ik}(\mathbf{z}_i - \mathbf{m}_k)(\mathbf{z}_i - \mathbf{m}_k)^\top}{\mathrm{tr}(\sum_i \tilde{q}_{ik}(\mathbf{z}_i - \mathbf{m}_k)(\mathbf{z}_i - \mathbf{m}_k)^\top) + \epsilon}. \tag{11}$$

Table 1: Results on mixed-grained benchmarks (ImageNet and its subsets). Higher is better. Best results are in bold and second-best results are underlined. "–" indicates results not reported in the original paper or cases where the provided code does not converge on the dataset.

Method	Backbone	ImageNet-50			ImageNet-100			ImageNet-200			ImageNet-1K		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SIC (AAAI'23)	ViT-B/32	75.4	81.4	67.1	67.2	77.2	55.8	57.9	73.4	46.4	47.0	77.2	34.3
TEMI (BMVC'23)	ViT-B/16	80.1	86.1	71.0	75.0	85.6	65.4	73.1	85.2	62.1	34.6	63.2	26.6
TAC (ICML'24)	ViT-B/32	78.7	83.7	67.5	63.8	68.1	57.6	59.2	70.9	50.8	<u>58.2</u>	<u>79.9</u>	<u>43.5</u>
CLUDI (ICML'25)	ViT-B/16	<u>82.1</u>	<u>91.2</u>	<u>76.2</u>	<u>76.6</u>	<u>87.1</u>	<u>67.8</u>	<u>73.7</u>	<u>86.1</u>	<u>63.2</u>	45.5	67.0	34.8
CAE (AAAI'26)	ViT-B/16	–	–	–	–	–	–	–	–	–	53.1	79.3	39.3
Ours	ViT-B/16	**86.9**	**92.4**	**78.9**	**79.4**	**88.5**	**70.5**	**75.3**	**88.3**	**66.0**	**60.2**	**81.5**	**45.0**

This normalization makes SSD depend on the fraction of variation outside the semantic subspace rather than on the absolute cluster radius. The residual-space decomposition and its normalized interpretation are given in Appendix B.1.

Covariance-Aware Nuisance Suppression. Using the normalized covariance above, the SSD loss suppresses the covariance energy that lies orthogonal to the semantic prior subspace:

$$\mathcal{L}_{\text{ssd}} = \frac{1}{|\mathcal{A}|} \sum_{k \in \mathcal{A}} \|\bar{\Sigma}_k - P_k \bar{\Sigma}_k P_k\|_F, \tag{12}$$

where $\mathcal{A}$ denotes the active cluster set. Writing $P_k^{\perp} = I - P_k$, the residual contains $P_k \bar{\Sigma}_k P_k^{\perp}$, $P_k^{\perp} \bar{\Sigma}_k P_k$, and $P_k^{\perp} \bar{\Sigma}_k P_k^{\perp}$. Its Frobenius norm penalizes both covariance outside the semantic subspace and cross-subspace coupling, while leaving the within-subspace block unpenalized. By decoupling semantic variation from nuisance noise, SSD improves intra-cluster consolidation across different granularity regimes.

3.6 Unified Objective

The components of our framework—representation learning, prototype assignment, H-DPP geometry, and SSD—are integrated into a unified training objective. Together they define a density- and semantics-conditioned geometric learning principle: H-DPP expands spectral volume with a batch-adaptive strength, while SSD consolidates each cluster around its semantically admissible subspace. Crucially, SSD counteracts H-DPP's inherent intra-cluster repulsion to maintain cluster compactness while preserving inter-cluster separation. The final training objective is:

$$\mathcal{L}^{(t)} = \lambda_{\text{inv}} \mathcal{L}_{\text{inv}} + \underbrace{\lambda_{\text{geom}} \left(\alpha_{\text{div}}^{(t)} \mathcal{L}_{\text{dpp}} + \beta_{\text{ssd}} \mathcal{L}_{\text{ssd}} \right)}_{\text{inter-cluster diversity and intra-cluster compactness}} . \tag{13}$$

Here $\mathcal{L}_{\text{inv}}$ enforces hyperspherical view-invariance, while the geometric term combines H-DPP and SSD for inter-cluster separation and intra-cluster compactness. DPCA in Section 3.3 supplies the forward assignments used by these components; Equation 2 is not an addend to this objective. Gradients update h_ϕ and f_ψ; the encoder, assignment targets, semantic queues, and coefficient statistic are detached.

4 Experiments

We evaluate MC-H across coarse-, mixed-, and fine-grained regimes, and assess the spectral extension as an effective-cardinality estimate. Unless otherwise stated, we report ACC, NMI, and ARI (higher is better).

4.1 Experimental Setup

Datasets. We group the benchmarks into three regimes according to their semantic granularity: coarse-grained (CIFAR-10, CIFAR-100-20, ImageNet-10, STL-10), mixed-grained (ImageNet-50/100/200/1K), and fine-grained (Stanford Dogs, Oxford Flowers, Oxford-IIIT Pet, CUB-200)

Table 2: Results on fine-grained benchmarks. Higher is better. Best results are in bold and second-best results are underlined. "–" indicates results not reported in the original paper or cases where the provided code does not converge on the dataset.

Method	Backbone	Stanford Dogs			Oxford Flowers			Oxford-IIIT Pet			CUB-200-2011		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SCAN (ECCV'20)	ResNet-50	12.3	35.0	16.4	56.5	77.0	22.0	60.4	76.0	40.3	11.9	45.0	2.4
MixNMatch (CVPR'20)	–	10.3	30.0	0.6	39.0	57.0	4.2	42.3	56.0	1.3	10.2	41.0	0.4
SimCLR (ICML'20)	ResNet-50	6.8	19.0	0.8	12.5	29.0	4.0	13.8	30.0	1.2	8.4	40.0	0.7
CC (AAAI'21)	ResNet-18	7.6	22.2	1.5	27.7	56.5	7.5	28.7	55.8	2.7	5.2	25.1	1.0
C3-GAN (ICLR'22)	–	17.9	36.0	1.5	67.8	67.0	7.9	51.2	57.0	1.9	27.6	53.0	1.4
TEMI (BMVC'23)	ViT-B/16	19.2	43.5	5.9	39.8	66.9	23.8	36.6	62.6	12.4	24.4	60.9	6.9
DivClust (CVPR'23)	ResNet-50	9.5	27.6	1.0	30.3	59.0	6.3	33.5	58.6	1.4	10.8	39.7	1.2
TAC (ICML'24)	ViT-B/32	48.7	64.8	**41.7**	53.2	73.2	65.9	61.3	76.5	61.4	34.7	64.3	25.7
IDC (NeurIPS'24)	ResNet-50	18.1	30.5	1.1	27.5	43.3	6.4	28.7	46.2	2.5	17.8	29.3	0.6
CLUDI (ICML'25)	ViT-B/16	42.6	63.7	24.9	69.7	81.5	22.7	74.1	87.3	49.7	30.2	61.3	8.9
Ours	ViT-B/16	**49.2**	**67.1**	40.5	**76.6**	**92.0**	**73.1**	**77.2**	**88.0**	**81.8**	**42.0**	**72.3**	**38.6**

Table 3: Results on coarse-grained benchmarks. Higher is better. Best results are in bold and second-best results are underlined. "–" indicates results not reported in the original paper or cases where the provided code does not converge on the dataset.

Method	Backbone	CIFAR-10			CIFAR-100-20			ImageNet-10			STL-10		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SCAN (ECCV'20)	ResNet-34	88.3	79.7	77.2	41.1	58.1	29.9	–	–	–	80.9	69.8	64.6
CC (AAAI'21)	ResNet-34	79.0	70.5	63.7	38.1	33.0	24.9	89.3	85.9	82.2	85.0	76.4	72.6
ProPos (TPAMI'22)	ResNet-34	94.3	88.6	88.4	35.8	46.4	26.5	95.6	89.6	90.6	86.7	75.8	73.7
SPICE (TIP'22)	ResNet	92.6	86.5	85.2	–	32.9	26.6	95.9	90.2	91.2	93.8	87.2	87.0
TCL (IJCV'22)	ResNet-34	88.7	81.9	78.0	36.0	51.9	27.4	89.5	87.5	83.7	86.8	79.9	75.7
SIC (AAAI'23)	ViT-B/32	–	–	–	58.3	59.3	43.9	98.2	97.0	96.1	98.1	95.3	95.9
TAC (ICML'24)	ViT-B/32	92.3	84.1	83.9	59.0	68.6	42.7	99.5	98.5	98.8	98.3	95.7	96.3
CLUDI (ICML'25)	ViT-B/16	95.3	89.6	89.8	60.4	63.8	38.9	75.0	85.3	72.4	75.1	75.8	73.4
SEIC (AAAI'26)	ViT-B/16	95.8	90.5	90.2	71.5	73.2	59.8	99.7	**99.2**	**99.4**	**99.3**	**98.0**	**98.5**
SATC (ICLR'26)	ViT-B/32	94.5	88.9	88.3	63.4	70.5	48.1	**99.8**	**99.2**	**99.4**	99.0	97.3	97.9
CAE (AAAI'26)	ViT-B/16	90.9	81.7	80.5	–	–	–	98.8	98.1	97.4	98.2	95.5	96.7
Ours	ViT-B/16	**96.7**	**91.3**	**91.0**	**73.4**	**74.1**	**61.0**	**99.8**	**99.2**	99.2	**99.3**	97.5	97.9

[23, 12, 11, 22, 37, 39, 49]. We follow the protocols of TEMI [1] and CLUDI [46]. MC-H uses frozen CLIP ViT-B/16 features (implementation details in Appendix A.2); because published systems may use different backbones or protocols, Appendix C.4 provides shared-backbone controls.

Baseline methods. We compare with purely visual methods, including SCAN [47], CC [27], ProPos [18], SPICE [38], TCL [28], SIC [6], SEIC [30], SATC [20], and CAE [53], as well as large-model-assisted methods (TAC [29], CLUDI [46], TEMI [1]) that use pretrained vision–language or generative models for semantic guidance.

Unknown-K evaluation and reporting. Main MC-H results use estimated $\hat{K}$. Each sample is assigned to its highest-scoring prototype; the predicted-cluster/label contingency matrix is padded to $\max(\hat{K}, K)$ and Hungarian-matched, so dummy matches count as errors. NMI and ARI use the raw partitions, and class-count errors are thus penalized without post-hoc correction. The fixed-K controls are reported separately. Main results use seed 42; error-bar provenance is given in Appendix A.2.

Controlled CIFAR-100 hierarchy setup. CIFAR-100 has a native hierarchy of 20 superclasses, each containing five fine classes. Ten superclasses remain broad targets, while each of the other ten is represented by its five fine classes, giving $10 + 10 \times 5 = 60$ targets. We generate 20 fixed Mixed-60 target partitions with seed 42, score the same predictions against them, and report mean global

Table 4: Dynamic-K prototype cardinality.

	CIFAR-10	CIFAR-100-20	STL-10	Image Net-10	Image Net-50	Image Net 100	Image Net 200	Image Net-1K	CUB-200-2011	Oxford Flowers	Oxford-IIIT Pet	Stanford Dogs
Pred. $\hat{K}$	14	26	12	11	54	89	237	1272	243	102	38	152
Ground Truth K	10	20	10	10	50	100	200	1000	200	102	37	120

Table 5: ImageNet label-granularity audit under the shared direct-hypernym criterion.

Dataset	Fine labels	Coarse labels	Total	Fine proportion
ImageNet-50	35	15	50	70.0%
ImageNet-100	69	31	100	69.0%
ImageNet-200	144	56	200	72.0%
ImageNet-1K	677	323	1,000	67.7%

Table 6: Controlled CIFAR-100 hierarchy evaluation. ACC / NMI / ARI (%) are reported at coarse, mixed, and fine targets.

Repulsion (DPCA and SSD fixed)	Superclass ($K = 20$) ACC / NMI / ARI	Mixed-60 ($K = 60$) ACC / NMI / ARI	Fine class ($K = 100$) ACC / NMI / ARI
Uniform ($\lambda = 0.05$)	60.6 / 65.1 / 47.7	42.6 / 67.4 / 37.1	46.0 / 72.2 / 40.9
Fixed H-DPP ($\alpha_{\text{base}} = 0.5$)	66.6 / 69.1 / 53.0	44.4 / 69.3 / 39.8	52.3 / 73.9 / 46.4
Adaptive H-DPP (MC-H)	**71.9 / 73.3 / 60.9**	**46.1 / 71.8 / 40.8**	**64.5 / 80.3 / 52.9**

ACC/NMI/ARI. These vary the evaluation hierarchy, not the training seed; labels are evaluation-only and no external images are added.

We compare uniform repulsion ($\lambda = 0.05$), fixed-strength H-DPP ($\alpha_{\text{base}} = 0.5$), and adaptive H-DPP, holding DPCA, SSD, encoder, augmentations, optimizer, training budget, and evaluation fixed. Dynamic-K is disabled, with $K = 60$ supplied for the mixed setting. Separate fixed-K runs at $K = 20$ and $K = 100$ evaluate the coarse and fine endpoints on the same data; these endpoints do not represent simultaneous recovery of multiple resolutions from one prediction.

4.2 Experimental Results

Tables 1–3 show strong performance across all three regimes. MC-H reaches **60.2%** ACC on ImageNet-1K, versus 47.0% for SIC, and **92.0%** NMI on Oxford Flowers. These comparisons assess the complete system; Section 4.3 and Appendix C.3 isolate adaptive repulsion from DPCA and SSD.

Table 4 reports effective prototype cardinality. Estimates match the annotated count for Oxford Flowers and are close for ImageNet-50 and Oxford-IIIT Pet, but substantial errors remain: ImageNet-1K is overestimated by 272 (27.2%) and Stanford Dogs by 32 (26.7%), while ImageNet-100 is underestimated by 11 (11.0%). We interpret $\hat{K}$ as a signal of active feature directions rather than an exact semantic class count. Appendix C.6 gives all absolute and relative errors and a post-hoc audit of excess clusters; only some excess clusters admit an interpretable semantic explanation.

4.3 Within-Dataset Mixed-Granularity Evidence

Native ImageNet granularity audit. To establish that multiple granularities occur within a dataset, we audit ImageNet-1K and its subsets using the label hierarchy. A label is fine-grained if it shares a direct WordNet hypernym with at least one other ImageNet-1K label, and coarse-grained otherwise; subsets use the same criterion. This identifies related subclasses, such as dog breeds, alongside broader categories such as volcano. Table 5 shows that every subset contains both groups, with 67.7–72.0% fine labels. The audit establishes coexistence, not clustering quality or failure of competing methods.

Controlled CIFAR-100 hierarchy evaluation. Under the shared controls of Section 4.1, adaptive H-DPP performs best at all three targets (Table 6). On Mixed-60, it improves ACC/NMI/ARI over fixed H-DPP by 1.7/2.5/1.0 points and over uniform repulsion by 3.5/4.4/3.7 points; its ACC gain over fixed H-DPP is 5.3 points at $K = 20$ and 12.2 at $K = 100$. These controls support adaptive geometric

Table 7: Ablation analysis of the proposed components (ACC, %).

Clustering Configuration	CIFAR-10	ImageNet-50	Oxford Flowers
Baseline (Frozen CLIP + $\mathcal{L}_{\text{inv}}$)	24.9	28.0	37.0
Baseline + Dynamic Prototype-Center Assignment (DPCA)	86.7	77.2	63.0
Baseline + DPCA + H-DPP Core (via $\mathcal{L}_{\text{dpp}}$)	87.0	78.0	68.8
Baseline + DPCA + SSD (via $\mathcal{L}_{\text{ssd}}$)	90.8	81.8	70.2
Ours (full model)	**96.7**	**86.9**	**76.6**

regularization when coarse and fine targets coexist, although the coefficient remains batch-level rather than a local density estimator.

4.4 Ablation and Hyperparameter Analysis

Ablation. Table 7 reports a component-wise ablation in ACC (%). Starting from frozen CLIP features with the invariance objective $\mathcal{L}_{\text{inv}}$, we add DPCA (Section 3.3), then probe the geometric module through its H-DPP-only and SSD-only variants before reporting the full model. The three datasets span coarse-, mixed-, and fine-grained regimes, separating prototype formation, diversity control, and semantic denoising.

DPCA yields the largest standalone improvement by forming geometry-aware prototypes. SSD contributes more than H-DPP when added alone, and combining both improves all three datasets. Appendix C.1 provides uniformity comparisons and the base-coefficient sweep (Table 10).

Figure 2 examines the invariance weight λ_{inv}, geometric weight λ_{geom}, and SSD weight β_{ssd}. Moderate changes around the defaults generally retain competitive performance. These full-objective sweeps assess sensitivity; they do not isolate individual components or establish the cause of stability. Controlled evidence for adaptive repulsion appears in Section 4.3 and Appendix C.3. Configurations and additional sweeps are given in Appendices A.2 and C.1.

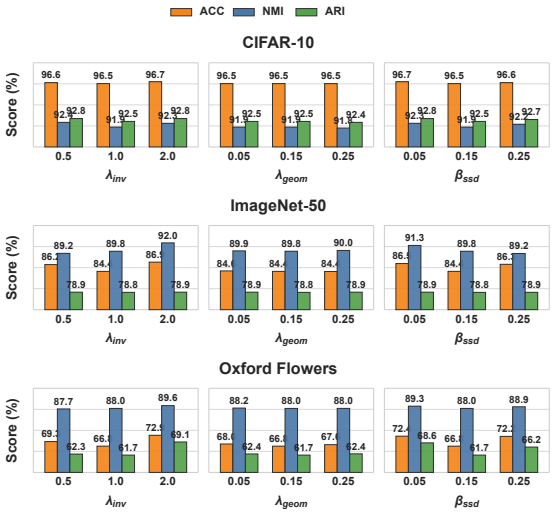


Figure 2: Hyperparameter analysis.

5 Conclusion

We have presented the MC-H framework for multi-granularity image clustering. Our main contribution is a unified geometric objective that combines density-adaptive H-DPP repulsion with SSD-based semantic denoising: H-DPP improves separability in the hyperspherical representation space, while SSD encourages compact and semantically coherent subspaces. DPCA provides quality-aware assignments and enables a spectral extension for estimating the cluster cardinality when K is unavailable. Experiments on coarse-grained, mixed-grained, and fine-grained benchmarks show that the proposed framework achieves stable performance across different semantic granularities and outperforms globally uniform repulsion baselines.

Limitations. MC-H depends on frozen-encoder semantic quality; ResNet-50 transfer yields lower performance. Batch size and feature dimension affect concentration and spectral estimates (Appendix C.5); batches of 128–256 are a practical range. The coefficient is batch-level and averages heterogeneous local geometry. Dynamic-K errors can merge or split annotated classes, without guaranteed semantic discovery. Published-system comparisons may differ in backbone or protocol, motivating matched controls. The current formulation uses a single-modality feature space.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 62606327, 62471266, 62006131, 62071260; the National Natural Science Foundation of Zhejiang Province under Grant LQ24F020001; the Ningbo Major Research and Development Plan Project (No.2023Z224); and the Intelligent Computing Center of Shenzhen University.

References

[1] N. Adaloglou, F. Michels, H. Kalisch, and M. Kollmann. Exploring the limits of deep image clustering using pretrained models, 2023. URL https://arxiv.org/abs/2303.17896.

[2] R. H. Affandi, A. Kulesza, E. Fox, and B. Taskar. Nystrom approximation for large-scale determinantal processes. In C. M. Carvalho and P. Ravikumar, editors, *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 85–98, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR. URL https://proceedings.mlr.press/v31/affandi13a.html.

[3] Y. M. Asano, C. Rupprecht, and A. Vedaldi. Self-labelling via simultaneous clustering and representation learning. *CoRR*, abs/1911.05371, 2019. URL http://arxiv.org/abs/1911.05371.

[4] N. Bajpai, J. H. Paik, and S. Sarkar. Balanced seed selection for k-means clustering with determinantal point process. *Pattern Recognition*, 164:111548, 2025. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2025.111548. URL https://www.sciencedirect.com/science/article/pii/S0031320325002080.

[5] R. Bardenet, S. Ghosh, H. Simon-Onfroy, and H.-S. Tran. Small coresets via negative dependence: DPPs, linear statistics, and concentration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=jd3msHMtTL.

[6] S. Cai, L. Qiu, X. Chen, Q. Zhang, and L. Chen. Semantic-enhanced image clustering, 2023. URL https://arxiv.org/abs/2208.09849.

[7] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part XIV*, page 139–156, Berlin, Heidelberg, 2018. Springer-Verlag. ISBN 978-3-030-01263-2. doi: 10.1007/978-3-030-01264-9_9. URL https://doi.org/10.1007/978-3-030-01264-9_9.

[8] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2021. URL https://arxiv.org/abs/2006.09882.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

[10] S. Cho, C. Li, D. Yu, H. Foroosh, and F. Liu. Multi-document summarization with determinantal point processes and contextualized representations, 2019. URL https://arxiv.org/abs/1910.11411.

[11] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database . In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pages 248–255, Los Alamitos, CA, USA, June 2009. IEEE Computer Society. doi: 10.1109/CVPR.2009.5206848. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2009.5206848.

[13] Y. Dong, S. Ye, Q. Han, Y. Cao, H. Xu, and H. Yang. A bayesian mixture model of temporal point processes with determinantal point process prior. *IEEE Transactions on Signal Processing*, 73:2216–2226, 2025. doi: 10.1109/TSP.2025.3575175.

[14] M. Elfeki, C. Couprie, M. Riviere, and M. Elhoseiny. GDPP: Learning diverse generations using determinantal point processes. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1774–1783. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/elfeki19a.html.

[15] Y. Guo, M. Xu, J. Li, B. Ni, X. Zhu, Z. Sun, and Y. Xu. HCSC: Hierarchical Contrastive Selective Coding . In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9696–9705, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.00948. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00948.

[16] J. B. Hough, M. Krishnapur, Y. Peres, and B. Virág. Determinantal Processes and Independence. *Probability Surveys*, 3(none):206 – 229, 2006. doi: 10.1214/154957806000000078. URL https://doi.org/10.1214/154957806000000078.

[17] J. Huang, S. Gong, and X. Zhu. Deep semantic clustering by partition confidence maximisation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8846–8855, 2020. doi: 10.1109/CVPR42600.2020.00887.

[18] Z. Huang, J. Chen, J. Zhang, and H. Shan. Learning representation for clustering via prototype scattering and positive sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7509–7524, 2023. doi: 10.1109/TPAMI.2022.3216454.

[19] X. Ji, A. Vedaldi, and J. Henriques. Invariant information clustering for unsupervised image classification and segmentation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9864–9873, Oct 2019. doi: 10.1109/ICCV.2019.00996.

[20] Z. Jiu, F. Li, J. Wang, Y. Qian, and L. Chen. Spatial structure and selective text jointly facilitate image clustering. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=3D0gmfZ2k6.

[21] T. Kazimi, C. Dunlop, and P. Yanardag. Diverse video generation with determinantal point process-guided policy optimization, 2025. URL https://arxiv.org/abs/2511.20647.

[22] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization : Stanford dogs. 2012. URL https://api.semanticscholar.org/CorpusID:3181866.

[23] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[24] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2-3):123–286, 12 2012. ISSN 1935-8237. doi: 10.1561/2200000044. URL https://doi.org/10.1561/2200000044.

[25] C. Li, S. Jegelka, and S. Sra. Efficient sampling for k-determinantal point processes, 2016. URL https://arxiv.org/abs/1509.01618.

[26] J. Li, P. Zhou, C. Xiong, and S. Hoi. Prototypical contrastive learning of unsupervised representations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KmykpuSrjcq.

[27] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng. Contrastive clustering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):8547–8555, May 2021. doi: 10.1609/aaai.v35i10.17037. URL https://ojs.aaai.org/index.php/AAAI/article/view/17037.

[28] Y. Li, M. Yang, D. Peng, T. Li, J. Huang, and X. Peng. Twin contrastive learning for online clustering. *International Journal of Computer Vision*, 130(9):2205–2221, Sep 2022. ISSN 1573-1405. doi: 10.1007/s11263-022-01639-z. URL https://doi.org/10.1007/s11263-022-01639-z.

[29] Y. Li, P. Hu, D. Peng, J. Lv, J. Fan, and X. Peng. Image clustering with external guidance. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=JSYN891WnB.

[30] Z. Li, W. Sun, J. Hu, J. Yin, J. Wu, and L. Nie. Self-enhanced image clustering with cross-modal semantic consistency, 2025. URL https://arxiv.org/abs/2508.01254.

[31] J. Lu, D. Wu, F. Nie, R. Wang, and X. Li. Hyperspherical prototype node clustering. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=z3ZlnaOM0d.

[32] J. Lys, V. Gripon, B. Pasdeloup, A. Marmoret, L. Mauch, F. Cardinaux, and G. B. Hacene. D5p4: Partition determinantal point process for diversity in parallel discrete diffusion decoding, 2026. URL https://arxiv.org/abs/2603.19146.

[33] Z. Mariet and S. Sra. Fixed-point algorithms for learning determinantal point processes. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2389–2397, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/mariet15.html.

[34] Z. Mariet and S. Sra. Diversity networks: Neural network compression using determinantal point processes, 2017. URL https://arxiv.org/abs/1511.05077.

[35] M. M. Morshed and V. Boddeti. Diverseflow: Sample-efficient diverse mode coverage in flows, 2025. URL https://arxiv.org/abs/2504.07894.

[36] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000. doi: 10.1080/10618600.2000.10474879. URL https://doi.org/10.1080/10618600.2000.10474879.

[37] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, ICVGIP '08, page 722–729, USA, 2008. IEEE Computer Society. ISBN 9780769534763. doi: 10.1109/ICVGIP.2008.47. URL https://doi.org/10.1109/ICVGIP.2008.47.

[38] C. Niu, H. Shan, and G. Wang. Spice: Semantic pseudo-labeling for image clustering. *IEEE Transactions on Image Processing*, 31:7264–7278, 2022. doi: 10.1109/TIP.2022.3221290.

[39] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. URL https://api.semanticscholar.org/CorpusID:279027499.

[40] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[41] S. Sarfraz, V. Sharma, and R. Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[42] N. Sogi, T. Shibata, M. Terao, M. Suganuma, and T. Okatani. Ms-dpps: multi-source determinantal point processes for contextual diversity refinement of composite attributes in text to image retrieval. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/207. URL https://doi.org/10.24963/ijcai.2025/207.

[43] J. Straub, T. Campbell, J. P. How, and J. W. Fisher. Small-variance nonparametric clustering on the hypersphere. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 334–342. IEEE, 2015. doi: 10.1109/cvpr.2015.7298630. URL http://dx.doi.org/10.1109/CVPR.2015.7298630.

[44] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001. doi: https://doi.org/10.1111/1467-9868.00293. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00293.

[45] N. Tremblay, S. Barthelmé, and P.-O. Amblard. Determinantal point processes for coresets, 2020. URL https://arxiv.org/abs/1803.08700.

[46] R. Uziel, I. Chelly, O. Freifeld, and A. Pakman. Clustering via self-supervised diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9uPCP3j62m.

[47] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool. Scan: Learning to classify images without labels. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, pages 268–285, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58607-2.

[48] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7492–7501, June 2022.

[49] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[50] X. Wei, Y. Chen, F. Nie, H. Liu, Q. Song, and L. Xiao. View-adaptive multi-granularity anchor learning for multi-view clustering. *IEEE Transactions on Image Processing*, 35:3113–3126, 2026. doi: 10.1109/TIP.2026.3674007.

[51] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 478–487, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/xieb16.html.

[52] Y. Yan, N. Lu, and R. Yan. Deep online probability aggregation clustering. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, editors, *Computer Vision – ECCV 2024*, pages 37–54, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-73668-1.

[53] X. Zhu, B. Zhu, Y. Li, J. Fang, S. Wang, K. Zhao, and H. Zhang. Hierarchical semantic alignment for image clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29177–29185, 2026.

A Implementation Details and Hyperparameter Settings

This section provides comprehensive details on the experimental setup, training protocols, and the practical implementation of the Semantic Subspace Denoising (SSD) module.

A.1 MC-H Training Procedure

Algorithm 1 specifies the order of the coupled operations. The learnable projection head h_ϕ and quality head f_ψ are updated by Equation 13; the CLIP encoder g remains frozen. We use $\mathcal{B}$ for the prototype bank, reserving P_k for the SSD semantic projector. The spectral bank is maintained by matching, sign alignment, and EMA, rather than by an inner DPCA–H-DPP fixed-point iteration.

Algorithm 1 One MC-H training iteration at step t

Require: Unlabeled batch $\{x_i\}_{i=1}^n$; frozen encoder g; heads h_ϕ, f_ψ; bank $\mathcal{B}$; per-cluster queues $\mathcal{Q}_k$; running statistic $\bar{\rho}_{t-1}$; refresh interval N_{sub}

Generate views x_i^a, x_i^b; compute frozen features $\mathbf{t}_i^v = g(x_i^v)$
Compute $\mathbf{z}_i^v = h_\phi(\mathbf{t}_i^v)/\|h_\phi(\mathbf{t}_i^v)\|_2$ and $r_i^v = f_\psi(\mathbf{z}_i^v)$
Center r_i^v within each view batch and exponentiate to obtain $w_i^v > 0$
Stack both views into Z and w; form $\Phi = \mathrm{Diag}(w)Z$ and $L = \Phi\Phi^\top$
Obtain the nonnegative spectrum $\{\lambda_j^L, \mathbf{u}_j\}$ of L
Set $\hat{K}_t = \mathrm{Round}(\sum_j \lambda_j^L/(1+\lambda_j^L))$ ▷ Use supplied K in fixed-K controls
Lift leading positive-eigenvalue directions: $\mathbf{v}_j = \Phi^\top \mathbf{u}_j/\sqrt{\lambda_j^L}$
if $\mathcal{B}$ is empty **then**
    Initialize $\mathcal{B}$ with the spectral prototypes
else
    Match prototypes to $\mathcal{B}$, align matched signs, and EMA-update matched directions
    Initialize unmatched new directions and retain the current $\hat{K}_t$ anchors
end if
Normalize bank rows; compute both-view logits and DPCA assignments using Section 3.3 and Equation 2
Detach assignment targets and queue-routing confidences
Compute $\mathcal{L}_{\text{inv}}$ (Equation 1) and $\mathcal{L}_{\text{dpp}}$ (Equation 3)
Compute batch angular concentration ρ_t; EMA-update $\bar{\rho}_t$ and set $\alpha_{\text{div}}^{(t)}$ by Equation 7
Set $\mathcal{L}_{\text{ssd}} \leftarrow 0$
if $t \bmod N_{\text{sub}} = 0$ **then**
    Estimate active semantic subspaces from current frozen features and $\mathcal{Q}_k$, using detached confidences
    Compute current student covariances and $\mathcal{L}_{\text{ssd}}$ (Equation 12); use zero if no cluster is active
end if
Backpropagate Equation 13 and update ϕ, ψ
Enqueue detached frozen features and confidences by highest-probability cluster; retain the most recent Q entries per queue
Retain normalized $\mathcal{B}$, queues, and $\bar{\rho}_t$ for step $t+1$

The cross-view DPCA quantity supplies forward assignment consistency and SSD routing; it is not a separate term in the backpropagated objective. The queues are updated every iteration, whereas subspace projectors and the SSD loss are refreshed every N_{sub} iterations. When bank cardinality changes, queue identities follow the prototype matching; newly introduced anchors start with empty queues. The student covariance uses current differentiable embeddings and detached assignments, so SSD gradients affect the student representation rather than the frozen semantic features.

A.2 Detailed Experimental Setup

Implementation environment. Unless a backbone comparison is specified, experiments employ a frozen CLIP ViT-B/16 image encoder. The codebase is implemented in Python 3.12 with PyTorch 2.8.0 and CUDA 12.6; training uses mixed precision (FP16). Most experiments run on a single

NVIDIA RTX 2080Ti GPU, while ImageNet subsets and ImageNet-1K-scale experiments use two NVIDIA RTX 4090 GPUs. Following official protocols, we partition the dataset into training and test sets. From the training set, we reserve 10% as a validation set to evaluate the performance of different loss functions. For datasets like ImageNet that do not provide a public test set, we utilize the official validation set as our test set.

Training schedule. The effective DPP strength $\alpha_{\text{div}}^{(t)}$ follows a linear warmup schedule:

$$\gamma(t) = \min\left(1, \frac{t}{T_{\text{warm}}}\right), \tag{14}$$

where T_{warm} is set to 10% of total training epochs. This allows initial cluster anchors to form before density-aware repulsion is fully activated. The partition temperature τ_{part} is annealed as:

$$\tau_{\text{part}}^{(t)} = \tau_{\text{part}}^{\max} - (\tau_{\text{part}}^{\max} - \tau_{\text{part}}^{\min}) \min\left(1, \frac{t}{T_{\text{anneal}}}\right), \tag{15}$$

where $\tau_{\text{part}}^{\max} = 0.5$, $\tau_{\text{part}}^{\min} = 0.1$, and T_{anneal} matches the training horizon. The sharpening temperature τ_{sharp} remains fixed at 0.1.

Prototype stabilization. For spectral prototype stabilization, we align current eigenvectors to the historical prototype bank using bipartite matching. We enforce sign consistency by ensuring matched directions maintain positive inner products before applying the EMA update with $\beta = 0.99$. This reduces prototype permutation and sign-flip instability when dynamic-K estimation is active.

Statistical reporting provenance. The main MC-H benchmark results are seed-42 results. The $\pm$ values in Tables 19, 20, and 21 are the originally reported hardware-repeat summaries: three seed-42 runs on each of RTX 2080 Ti, RTX 3080, and RTX 4090 are averaged per device, and the reported standard error is $s/\sqrt{3}$ across those three device averages. These values describe device/FP16 nondeterminism under a fixed seed, not variability across independent random initializations. The 20 seed-2026 Mixed-60 evaluation partitions in Section 4.3 vary target granularity and likewise do not constitute a multi-seed training study. Independent-seed results for ImageNet-50, ImageNet-100, ImageNet-200, and ImageNet-1K, as well as the coarse- and fine-grained benchmarks, are reported separately in Appendix C.7 as mean $\pm$ sample standard deviation across five seeds. These results assess variation across random seeds rather than the fixed-seed hardware variation summarized above.

Hyperparameter summary. The main objective in Equation 13 combines $\mathcal{L}_{\text{inv}}$ with the geometric module $\mathcal{L}_{\text{geom}}^{(t)}$, for which we use the default weight $\lambda_{\text{geom}} = 0.15$. Within the geometric term, we set the repulsion coefficient $\alpha_{\text{base}} = 0.5$ and the SSD coefficient $\beta_{\text{ssd}} = 0.15$. For the invariance objective, we use $\tau_{\text{inv}} = 0.2$. All values stated here serve as defaults; deviations are noted explicitly where applicable. Table 8 summarizes the dataset-wise loss hyperparameters used in the released training configurations. All datasets share the same default settings, which is a deliberate choice: rather than tuning per dataset, we fix hyperparameters across all benchmarks to demonstrate that the density-adaptive mechanism and semantic denoising naturally accommodate different granularity regimes without dataset-specific adjustments.

Table 8: Dataset-wise loss hyperparameters used in the released training configurations.

Dataset	λ_{inv}	τ_{inv}	β_{ssd}	α_{base}	λ_{geom}
CIFAR-10	1.0	0.2	0.15	0.5	0.15
CIFAR-100-20	1.0	0.2	0.15	0.5	0.15
STL-10	1.0	0.2	0.15	0.5	0.15
ImageNet-10	1.0	0.2	0.15	0.5	0.15
ImageNet-50	1.0	0.2	0.15	0.5	0.15
ImageNet-100	1.0	0.2	0.15	0.5	0.15
ImageNet-200	1.0	0.2	0.15	0.5	0.15
ImageNet-1K	1.0	0.2	0.15	0.5	0.15
CUB-200-2011	1.0	0.2	0.15	0.5	0.15
Oxford Flowers	1.0	0.2	0.15	0.5	0.15
Oxford-IIIT Pet	1.0	0.2	0.15	0.5	0.15
Stanford Dogs	1.0	0.2	0.15	0.5	0.15

A.3 Implementation Details of Semantic Subspace Denoising (SSD)

Semantic subspace definition. While the main text defines SSD at an abstract level, the practical instantiation relies on a temporally smoothed estimate of cluster-conditional semantic subspaces. The semantic prior feature $\mathbf{t}_i$ is instantiated as the frozen CLIP image feature of sample x_i. At each training step, frozen teacher features are computed for two augmented views, concatenated into a batch-level matrix, and subsequently mapped to student features through the learnable adapter. For each cluster, SSD maintains a bounded temporal feature bank of recent teacher features weighted by their confidence scores, derived from the current soft assignments. After each optimizer step, every sample is assigned to its highest-probability cluster, and the corresponding teacher feature is appended to that cluster's feature bank, which retains only the Q most recent entries (where Q denotes the queue size). Consequently, the semantic subspace employed by SSD is the confidence-weighted principal subspace of frozen CLIP image features that have been consistently routed to the same cluster.

Active cluster selection and temporal basis estimation. Let q_{ik} denote the detached soft assignment of sample i to cluster k in the current batch. For each cluster with sufficient current and queued teacher features to support a rank-r estimate, the basis U_k is estimated from the union of current-batch teacher features and the temporally retained teacher features for that cluster, via weighted low-rank PCA with a small oversampling margin Δq. The covariance term in Equation 12 is then computed solely from the current-batch student features, weighted by the detached assignments q_{ik}. Clusters whose available feature count is smaller than r are omitted from the SSD loss for that step. This defines the operational meaning of the active set $\mathcal{A}$ and the projector P_k in Equation 12.

Update cadence and computational efficiency. The SSD loss is not evaluated at every optimization step. Instead, it is computed every N_{sub} iterations and set to zero on intervening steps, consistent with its role as a geometric correction term rather than the primary training signal. The temporal feature bank is nevertheless updated after every optimizer step, ensuring that the semantic basis remains smooth even when the loss is evaluated sparsely. The projectors are refreshed together with the SSD loss every N_{sub} iterations. This design reduces the cost of repeated PCA while the queue collects features at every step.

Parameter settings and rationale. The released configurations follow a compact set of design rules. First, the subspace rank $r = 8$ and queue size $Q = 64$ are fixed across nearly all datasets. The rank is intentionally low so that SSD captures only dominant semantic directions rather than fitting nuisance variation; the temporal window is large enough to smooth mini-batch noise yet short enough to prevent stale cluster identities. Second, the PCA oversampling margin Δq is kept small (typically 2), providing a modest accuracy buffer for low-rank PCA at negligible additional cost. Third, the released configurations use a fixed SSD coefficient $\beta_{\text{ssd}} = 0.15$ across datasets, which provided a stable trade-off between semantic denoising and geometric separation in the final runs.

B Methodological Derivations and Geometric Analysis

This section provides supplementary mathematical derivations for the Semantic Subspace Denoising (SSD) module and formalizes the density-aware verification protocols.

B.1 Supplementary Methodological Derivations

Residual-space normalization for SSD. To compare fine- and coarse-grained clusters in a common coordinate system, we decompose each centered feature into semantic and residual components:

$$\mathbf{m}_k = \sum_i \tilde{q}_{ik}\mathbf{z}_i, \qquad \boldsymbol{\delta}_{ik} = \mathbf{z}_i - \mathbf{m}_k, \qquad \boldsymbol{\delta}_{\text{sem}} = P_k\boldsymbol{\delta}_{ik}, \quad \boldsymbol{\delta}_{\text{res}} = (I - P_k)\boldsymbol{\delta}_{ik}. \tag{16}$$

The corresponding soft-assignment-weighted covariance is

$$\Sigma_k = \sum_i \tilde{q}_{ik}\,\boldsymbol{\delta}_{ik}\boldsymbol{\delta}_{ik}^{\top}. \tag{17}$$

Normalizing by total cluster energy yields

$$\bar{\Sigma}_k = \frac{\Sigma_k}{\mathrm{tr}(\Sigma_k) + \epsilon}, \qquad r_{k,\mathrm{res}} = \mathrm{tr}\big((I - P_k)\bar{\Sigma}_k(I - P_k)\big) = \frac{\sum_i \tilde{q}_{ik}\|\boldsymbol{\delta}_{\mathrm{res}}\|_2^2}{\sum_i \tilde{q}_{ik}\|\boldsymbol{\delta}_{ik}\|_2^2 + \epsilon}. \tag{18}$$

Thus, SSD measures the fraction of cluster variation lying outside the semantic subspace rather than the absolute cluster radius.

Why the residual covariance penalty suppresses nuisance variation. Let $P = P_k$ and $P^\perp = I - P$. The covariance decomposes into four orthogonal blocks:

$$\bar{\Sigma}_k = P\bar{\Sigma}_k P + P\bar{\Sigma}_k P^\perp + P^\perp\bar{\Sigma}_k P + P^\perp\bar{\Sigma}_k P^\perp. \tag{19}$$

Consequently,

$$\|\bar{\Sigma}_k - P\bar{\Sigma}_k P\|_F^2 = \|P\bar{\Sigma}_k P^\perp\|_F^2 + \|P^\perp\bar{\Sigma}_k P\|_F^2 + \|P^\perp\bar{\Sigma}_k P^\perp\|_F^2. \tag{20}$$

Minimizing the residual norm penalizes covariance fully outside the semantic subspace as well as its coupling to the semantic block. The within-subspace covariance $P\bar{\Sigma}_k P$ is not directly penalized. Trace normalization makes this a relative covariance penalty, rather than a penalty on absolute cluster radius.

B.2 Density-Aware Geometry Verification

This section formalizes the diagnostic quantities used to characterize the density-adaptive behavior of H-DPP. The analysis complements the objective-component ablations by relating the adaptive repulsion coefficient to measurable feature-space redundancy.

Density statistic. The batch-level density statistic ρ measures instantaneous feature redundancy via the average pairwise cosine similarity. For a batch of ℓ_2-normalized features $\tilde{\mathbf{z}}_i = \mathbf{z}_i/\|\mathbf{z}_i\|_2$, we define

$$\rho_t = \frac{1}{B(B-1)} \sum_{i \neq j} \tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j, \tag{21}$$

where B denotes the batch size. Larger ρ_t indicates greater batch angular concentration; smaller ρ_t indicates a more dispersed batch. This is a batch statistic, not a sample-level or probability-density estimate. Short-term mini-batch fluctuations are attenuated via exponential moving average smoothing:

$$\bar{\rho}_t = \beta\,\bar{\rho}_{t-1} + (1-\beta)\rho_t, \tag{22}$$

with $\beta = 0.85$.

Cross-dataset density comparison. To verify that semantic granularity is reflected in the learned geometry, we track $\bar{\rho}_t$ during training on representative coarse-grained, mixed-grained, and fine-grained datasets. The expected ordering is

$$\bar{\rho}_{\mathrm{fine}} > \bar{\rho}_{\mathrm{mixed}} > \bar{\rho}_{\mathrm{coarse}}, \tag{23}$$

since fine-grained categories contain more visually similar classes and thus produce denser hyperspherical neighborhoods. This diagnostic characterizes representation concentration across benchmark groups. It does not isolate the effect of adaptive repulsion on clustering accuracy; the fixed-strength controls in Table 6 and Table 13 address that comparison.

Scheduling and effective repulsion. The density statistic is not itself the final strength of the geometric constraint. Instead, it serves as the input signal used to schedule the adaptive coefficient:

$$\alpha_{\mathrm{div}}^{(t)} = \gamma(t) \cdot \min\left(\max\left(\alpha_{\mathrm{base}} \cdot \mathrm{sg}(\bar{\rho}_t), 0\right), 1\right). \tag{24}$$

Accordingly, $\bar{\rho}_t$ provides the density signal, while $\alpha_{\mathrm{div}}^{(t)}$ applies the theory-driven warmup, clipping, and scaling needed to transform that signal into a usable optimization coefficient. We refer to the resulting realized geometric pressure as the *effective repulsion*: it is the final strength with which the H-DPP term acts on the representation after density estimation and coefficient scheduling have both been applied.

Granularity-dependent DPP repulsion. The H-DPP objective maximizes the log-volume spanned by quality-weighted features. In dense regions, high pairwise similarities yield a small determinant, producing stronger repulsive gradients. In sparse regions, feature directions are already well-separated and the determinant penalty is correspondingly weaker. Combined with the adaptive coefficient $\alpha_{\text{div}}^{(t)}$, the resulting repulsion follows the qualitative ordering

$$\text{repulsion}_{\text{fine}} > \text{repulsion}_{\text{mixed}} > \text{repulsion}_{\text{coarse}}. \tag{25}$$

This mechanism explains geometrically why H-DPP improves fine-grained separation without unnecessarily fragmenting coarse-grained clusters.

B.3 Granularity-Wise Density and Repulsion Analysis

To provide a complementary cross-dataset perspective on the density-adaptive behavior of H-DPP, we aggregate the post-warmup mean values of three key quantities—the smoothed density signal $\bar{\rho}_t$, the scheduled coefficient α_{div}, and the resulting effective repulsion—across all datasets in each granularity regime. The aggregation yields a single summary value per granularity level per quantity, visualized in Figure 3. The coarse-grained group comprises CIFAR-10, CIFAR-100-20, ImageNet-10 and STL-10; the mixed-grained group comprises ImageNet-50, ImageNet-100, ImageNet-200 and ImageNet-1K; the fine-grained group comprises CUB-200-2011, Stanford Dogs, Oxford Flowers and Oxford-IIIT Pet.

Aggregated angular concentration. The left panel of Figure 3 reports post-warmup mean $\bar{\rho}_t$. Fine-grained datasets have the highest mean (1.173×10^{-3}), compared with coarse-grained (4.035×10^{-4}) and mixed-grained (4.566×10^{-4}) datasets. This is consistent with greater feature concentration for visually similar categories, while remaining an aggregate cross-dataset observation.

Scheduled coefficient and effective repulsion. The mean coefficient is 5.864×10^{-4} for fine-grained, 2.018×10^{-4} for coarse-grained, and 2.283×10^{-4} for mixed-grained datasets. Its ordering follows from Equation 7: after warmup, positive unclipped concentration values are scaled by α_{base}. The coefficient ordering therefore follows by definition, rather than providing independent evidence of a clustering advantage. The effective-repulsion panel similarly describes the response of the implemented geometric term. These diagnostics complement, but do not replace, the controlled comparisons in Section 4.3 and Appendix C.3.

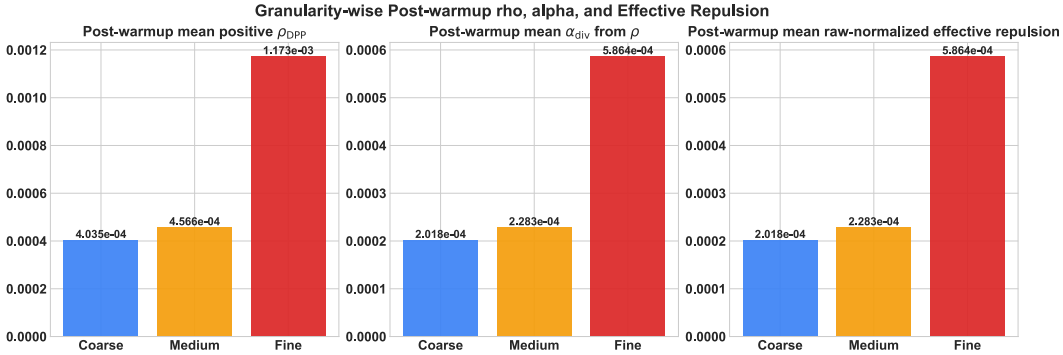


Figure 3: Post-warmup mean values of the smoothed density signal $\bar{\rho}_t$, the scheduled coefficient α_{div}, and the effective repulsion, aggregated across coarse-grained, mixed-grained, and fine-grained datasets.

C Additional Experimental Results and Visualizations

This section collects ablation studies, batch-size sensitivity analysis, and extensive visualizations that corroborate the main experimental claims.

C.1 Additional Ablations

Objective-component ablation. Table 7 studies a five-way transition from a baseline clustering setup built on frozen CLIP features and $\mathcal{L}_{\text{inv}}$ to the full model. Adding DPCA introduces geometry-

aware prototype structure on top of the baseline representation, while the H-DPP-only and SSD-only variants separately expose the contributions of inter-cluster diversity and subspace-based precision. Taken together, the table shows that DPCA is the main structural bridge from view-invariant features to cluster formation, that SSD provides the larger standalone geometric gain in this comparison, and that combining SSD with H-DPP yields the best overall performance.

Table 9: End-to-end uniformity versus geometry comparison. Uniformity variants omit SSD; the geometric variant includes H-DPP and SSD.

Dataset	Baseline + DPCA + $\mathcal{L}_{\text{uniform}}$ ($\lambda = 0.05$)			Baseline + DPCA + $\mathcal{L}_{\text{uniform}}$ ($\lambda = 0.10$)			Baseline + DPCA + $\mathcal{L}_{\text{uniform}}$ ($\lambda = 0.20$)			Baseline + DPCA + $\mathcal{L}_{\text{geom}}$		
	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
CIFAR-10	**96.8**	**92.7**	**93.2**	96.8	92.6	93.1	96.4	91.8	92.3	96.7	91.3	91.0
ImageNet-50	84.0	89.9	78.5	84.0	90.0	78.7	84.2	90.1	78.8	**86.9**	**92.4**	**78.9**
Oxford Flowers	67.1	87.5	61.7	69.4	87.3	62.2	68.3	87.8	62.4	**76.6**	**92.0**	**73.1**

Uniformity versus H-DPP geometry. Table 9 is an end-to-end comparison with the baseline representation and DPCA fixed. The uniformity variants omit SSD, whereas $\mathcal{L}_{\text{geom}}$ includes both H-DPP and SSD. Uniformity performs better on CIFAR-10; the full geometry variant performs better on ImageNet-50 and Oxford Flowers. Because SSD also changes, this table cannot isolate the advantage of adaptive over uniform repulsion. Table 13 supplies that control with SSD held fixed.

Sensitivity to the base H-DPP coefficient. We further examine the sensitivity of the model to the base repulsion coefficient α_{base} on six representative datasets. The results indicate that the preferred setting is dataset-dependent: CIFAR-10 is essentially flat across the tested range, STL-10 favors weaker repulsion, ImageNet-50 favors the default setting, and ImageNet-100 performs best under a stronger coefficient. Oxford Flowers and CUB-200-2011 show comparatively small to moderate variation across the tested range, suggesting that the method is reasonably stable once α_{base} is chosen within a sensible interval. Overall, the sweep supports the practical use of $\alpha_{\text{base}} = 0.5$ as a robust default, while also indicating that some datasets can benefit from lighter or stronger repulsion depending on their local semantic density.

Table 10: Sensitivity to $\alpha_{\text{base}} \in \{0.3, 0.5, 1.0\}$ across six datasets. ACC (%) reported on the official evaluation split.

Dataset	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 1.0$	Best α	ΔACC
CIFAR-10	**96.7**	**96.7**	**96.7**	0.3 / 0.5 / 1.0	0.0
STL-10	**99.4**	99.3	99.0	0.3	0.4
ImageNet-50	86.8	**86.9**	86.8	0.5	0.1
ImageNet-100	79.4	79.4	**79.5**	1.0	0.1
Oxford Flowers	76.2	76.6	**77.0**	1.0	0.8
CUB-200-2011	41.8	42.0	**42.5**	1.0	0.7

Backbone comparison. Table 11 evaluates the effect of the visual encoder across six representative datasets. ViT-B/16 achieves the highest ACC, NMI, and ARI on all datasets, indicating that its denser patch resolution supplies stronger semantic features for the proposed geometric objective. The gains are most pronounced on fine-grained datasets (Oxford Flowers, Oxford-IIIT Pet), where local visual cues are critical for disambiguating visually similar categories. At the same time, our method generalizes to ResNet-50 and ViT-B/32 features, demonstrating that the method is not tied to a single backbone architecture.

Table 11: Backbone comparison across six representative datasets.

Backbone	CIFAR10			Image50			Oxford Flowers			STL10			Image100			Oxford Pet		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
ResNet-50	68.7	58.0	49.2	80.8	84.7	73.4	62.1	79.8	54.6	95.9	91.4	91.4	74.3	83.2	66.1	60.8	71.2	48.6
ViT-B/32	93.9	86.9	87.2	84.5	88.3	76.8	69.8	86.7	61.9	97.5	94.3	94.6	77.8	86.9	68.8	66.5	78.1	57.4
ViT-B/16	**96.7**	**91.3**	**91.0**	**86.9**	**92.4**	**78.9**	**76.6**	**92.0**	**73.1**	**99.3**	**97.5**	**97.9**	**79.4**	**88.5**	**70.5**	**77.2**	**88.0**	**81.8**

C.2 Batch-Size Sensitivity

H-DPP relies on mini-batch statistics for density estimation, kernel construction, and spectral estimation, so its geometric behavior depends on how much of the feature distribution is represented in each batch. Table 12 reports ACC under batch sizes 32, 128, and 256 on six representative datasets spanning coarse-, mixed-, and fine-grained regimes.

Table 12: Batch-size sensitivity across six representative datasets. We report clustering ACC (%) on the dataset-specific evaluation split for training batch sizes 32, 128, and 256. The best result in each row is boldfaced.

Dataset	BS=32	BS=128	BS=256	Best BS
CIFAR-10	96.6	**96.7**	96.7	128
STL-10	**98.9**	98.8	98.8	32
ImageNet-50	85.6	86.8	**86.9**	256
ImageNet-100	77.5	79.0	**79.4**	256
Oxford Flowers	75.1	**76.6**	74.9	128
CUB-200-2011	39.3	38.6	**42.0**	256

Moderate or large batch sizes are generally preferable for H-DPP. Batch sizes 128 and 256 yield the best ACC on five of six datasets, with clear gains on ImageNet-50, ImageNet-100, and CUB-200-2011. CIFAR-10 is nearly invariant across the tested range, while STL-10 slightly favors 32, indicating that coarse-grained datasets with simpler geometry are less sensitive to the batch-level density estimate. These results support the practical recommendation of batch sizes 128–256 when applying H-DPP to mixed- or fine-grained benchmarks.

We apply EMA smoothing to ρ_t to attenuate short-term noise. When memory is constrained, two strategies are practical: constructing the H-DPP kernel from a memory bank of recent features, and accumulating the $d \times d$ scatter matrix $S = \Phi^\top \Phi$ across multiple mini-batches before eigendecomposition. Both approaches increase the effective sample size without enlarging the physical batch.

C.3 Matched Repulsion Ablation with SSD Held Fixed

The end-to-end comparison in Table 9 changes both the repulsion and the presence of SSD. To isolate the repulsion mechanism, Table 13 keeps DPCA, SSD, the compactness weight, and the tuning budget fixed, varying only uniform repulsion, fixed-strength H-DPP, and density-adaptive H-DPP. These are separate controlled runs, so their full-model values need not equal the original benchmark results.

Table 13: SSD-held-fixed repulsion ablation. All entries are ACC / NMI / ARI (%). Bold denotes the best value, including ties.

Repulsion (DPCA and SSD fixed)	CIFAR-10 ACC / NMI / ARI	ImageNet-50 ACC / NMI / ARI	Oxford Flowers ACC / NMI / ARI
Uniform ($\lambda = 0.05$)	96.5 / 93.1 / 93.4	84.9 / 90.2 / 79.0	67.8 / 88.0 / 62.0
Fixed H-DPP ($\alpha_{\text{base}} = 0.5$)	97.0 / 93.2 / 93.4	85.6 / 91.7 / **79.2**	68.3 / 86.7 / 61.4
Adaptive H-DPP (MC-H)	**97.4 / 94.0 / 94.5**	**87.1 / 92.4 / 79.2**	**77.2 / 88.1 / 81.9**

Adaptive H-DPP improves ACC over fixed-strength H-DPP by 0.4, 1.5, and 8.9 points on CIFAR-10, ImageNet-50, and Oxford Flowers, respectively. The ImageNet-50 ARI is tied at 79.2. Together with the hierarchy controls in Table 6, these results support a complementary contribution from adaptive repulsion while recognizing the substantial contributions of DPCA and SSD.

C.4 Matched-Backbone Baselines on CIFAR-100 Mixed-60

Table 14 fixes frozen CLIP ViT-B/16, the data split, augmentations, $K = 60$, training budget, and evaluation protocol. Frozen-feature baselines directly cluster the encoder features. Adapted-

feature baselines apply the corresponding post-hoc clustering to adapted embeddings, separating representation adaptation from the full prototype–geometry loop. DPCA-only retains assignment construction while disabling H-DPP and SSD gradients. This control is distinct from comparisons against published systems with different encoders or protocols.

Table 14: Matched CIFAR-100 Mixed-60 baselines. ACC, NMI, and ARI are percentages; all methods use the same frozen backbone and fixed $K = 60$.

Method	ACC	NMI	ARI
Frozen-feature Euclidean k-means	40.7	65.4	30.9
Frozen-feature spherical k-means	41.3	65.6	32.5
Adapted-feature Euclidean k-means	42.7	68.7	35.3
Adapted-feature spherical k-means	43.2	68.8	35.8
DPCA-only	43.4	68.3	37.8
Full MC-H	**46.1**	**71.8**	**40.8**

Full MC-H improves ACC/NMI/ARI over frozen-feature spherical k-means by 4.8/6.2/8.3 points and over DPCA-only by 2.7/3.5/3.0 points. Adaptation alone improves the frozen-feature baselines, but the complete model gives the best reported metrics. The backbone comparison in Table 11 provides an additional transfer check: with ResNet-50, MC-H obtains 80.8/84.7/73.4 on ImageNet-50 and 74.3/83.2/66.1 on ImageNet-100. Lower absolute performance under weaker features remains a limitation.

C.5 Batch-Size and Feature-Dimension Controls on Mixed-60

Table 15 holds the dataset, frozen ViT-B/16 backbone, $K = 60$, optimizer, 30-epoch budget, and evaluation fixed. Batch-size rows use dimension 512; feature-dimension rows use batch size 192. At dimension 512, batch 256 improves ACC/NMI/ARI by 1.3/1.3/3.6 points over batch 32. At batch 192, dimension 512 improves by 2.1/1.0/3.7 points over dimension 256, while increasing the dimension to 1024 does not improve the metrics.

Table 15: Batch-size and feature-dimension sensitivity on CIFAR-100 Mixed-60 (%). Bold identifies the best result within each control.

Control	Batch	Dimension	ACC	NMI	ARI
Batch size	32	512	45.1	71.0	37.3
Batch size	128	512	45.5	71.4	37.7
Batch size	256	512	**46.4**	**72.3**	**40.9**
Feature dimension	192	256	44.0	71.1	37.1
Feature dimension	192	512	**46.1**	**72.1**	**40.8**
Feature dimension	192	1024	45.2	71.4	37.2

The dimension-512 result here reports NMI 72.1, whereas the hierarchy and matched-baseline runs report 71.8; we preserve the separately reported control results. These controls show sensitivity to the batch and feature configuration rather than invariance to either choice.

C.6 Effective Cardinality Errors and Excess-Cluster Audit

Class-count error. Table 16 expands Table 4 with absolute and relative errors. Here the relative error is $|\hat{K} - K|/K$. Estimates can deviate substantially from the annotated class count: CIFAR-10 has 40.0% relative error, CIFAR-100-20 has 30.0%, and ImageNet-1K has 27.2%. The spectral estimate measures effective directions in the frozen-feature geometry, and should not be interpreted as guaranteed recovery of semantic categories.

Post-hoc semantic audit. Table 17 reports the number of excess clusters judged interpretable in a manual audit. On ImageNet-50, the audit identifies Table Lamp (633 samples, 87.7% purity) and Goose (631 samples, 60.1% purity). These examples show that some excess directions align with recognizable semantic structure; they do not validate every excess cluster or establish a general class-discovery capability. The audit is descriptive, uses labels only after clustering, and does not modify the predictions or the unknown-K evaluation.

Table 16: Absolute and relative class-count errors derived from the reported effective-cardinality estimates.

Dataset	K	$\hat{K}$	$\lvert\hat{K}-K\rvert$	$\lvert\hat{K}-K\rvert/K$
CIFAR-10	10	14	4	40.0%
CIFAR-100-20	20	26	6	30.0%
STL-10	10	12	2	20.0%
ImageNet-10	10	11	1	10.0%
ImageNet-50	50	54	4	8.0%
ImageNet-100	100	89	11	11.0%
ImageNet-200	200	237	37	18.5%
ImageNet-1K	1,000	1,272	272	27.2%
CUB-200-2011	200	243	43	21.5%
Oxford Flowers	102	102	0	0.0%
Oxford-IIIT Pet	37	38	1	2.7%
Stanford Dogs	120	152	32	26.7%

Table 17: Manual audit of excess clusters. The final column counts excess clusters judged meaningful, rather than all excess clusters.

Dataset	K	$\hat{K}$	$\lvert\hat{K}-K\rvert$	Relative error	Interpretable excess clusters
ImageNet-50	50	54	4	8.0%	2
Stanford Dogs	120	152	32	26.7%	6
CIFAR-100-20	20	26	6	30.0%	1

Other excess clusters may reflect pose, illumination, background, or small-sample variation in the frozen representation. Underestimation can merge annotated categories and overestimation can split them. Padded Hungarian matching counts unmatched clusters as errors rather than correcting the estimate; NMI and ARI also evaluate the resulting partition directly. We do not claim robustness to large cardinality errors.

C.7 Multi-Seed Evaluation

We evaluate MC-H with five random seeds, 42–46, on the official validation split of ImageNet-50, ImageNet-100, ImageNet-200, and ImageNet-1K. Table 18 reports ACC, NMI, and ARI as mean $\pm$ sample standard deviation across the five runs, with the standard deviation computed using denominator $5-1$. The error bars measure variation across random seeds and are distinct from the fixed-seed hardware-repeat standard errors reported in the complete benchmark tables.

We also train MC-H for 30 epochs with five independent seeds (42–46) on four coarse-grained and four fine-grained benchmarks. Each run uses the existing dataset configuration with a fixed seed-42 validation partition, selects its best checkpoint by the maximum validation ACC + NMI + ARI, and evaluates it on a separate evaluation split. Table 18 reports mean $\pm$ sample standard deviation across seeds (percent). ImageNet-10 uses a fixed class-stratified seed-42 holdout from its original images; Oxford Flowers uses `flower_data/valid` as the labeled evaluation split.

C.8 Supplementary Visualizations and Complete Results

t-SNE visualizations. Figures 4–6 contrast the raw feature distribution with the feature representations learned by our method. These visualizations illustrate the qualitative effect of our framework: compared to the raw feature distribution, the clusters become clear and well-separable after applying our method, where coarse-grained categories become cleanly separated, mixed-grained structures remain stable under semantic overlap, and fine-grained clusters become more distinguishable after adaptive repulsion.

Complete benchmark tables. Tables 19, 20, and 21 provide the complete fine-grained, coarse-grained, and mixed-grained comparisons against additional baselines. Across all three granularity

Table 18: MC-H multi-seed results across coarse-, mixed-, and fine-grained benchmarks. Metrics are percentages, reported as mean ± sample standard deviation over seeds 42–46. Evaluation protocols are described above.

Granularity	Dataset	ACC	NMI	ARI
Coarse	CIFAR-10	97.0 ± 0.1	92.8 ± 0.1	91.4 ± 0.1
	CIFAR-100-20	71.8 ± 2.8	72.3 ± 1.9	59.6 ± 3.3
	ImageNet-10	99.6 ± 0.2	98.9 ± 0.4	99.0 ± 0.4
	STL-10	99.0 ± 0.1	98.8 ± 0.1	97.6 ± 0.1
Mixed	ImageNet-50	86.6 ± 0.1	92.4 ± 0.1	79.2 ± 0.2
	ImageNet-100	79.6 ± 0.2	89.4 ± 0.1	70.4 ± 0.2
	ImageNet-200	75.6 ± 0.5	88.2 ± 0.4	66.5 ± 0.5
	ImageNet-1K	59.9 ± 0.2	81.7 ± 0.1	44.6 ± 0.3
Fine	Stanford Dogs	49.2 ± 0.6	67.2 ± 0.3	40.8 ± 0.5
	Oxford-IIIT Pet	77.2 ± 0.2	88.0 ± 0.3	81.6 ± 0.4
	CUB-200-2011	42.5 ± 0.4	72.9 ± 0.2	38.8 ± 0.4
	Oxford Flowers	76.8 ± 2.5	91.1 ± 0.9	75.1 ± 2.8

regimes, our method consistently achieves the best or second-best results on nearly every dataset–backbone combination. These supplementary comparisons confirm that the proposed unified geometric objective generalizes across granularity regimes without requiring per-dataset hyperparameter tuning.

Table 19: Additional fine-grained results (ACC, NMI, ARI). MC-H error bars describe fixed-seed hardware repeats; see Appendix A.2.

Method	Backbone	Stanford Dogs			Oxford Flowers			Oxford-IIIT Pet			CUB-200-2011		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SCAN (ECCV'20)	ResNet-50	12.3	35.0	16.4	56.5	77.0	22.0	60.4	76.0	40.3	11.9	45.0	2.4
MixNMatch (CVPR'20)	–	10.3	30.0	0.6	39.0	57.0	4.2	42.3	56.0	1.3	10.2	41.0	0.4
SimCLR (ICML'20)	ResNet-50	6.8	19.0	0.8	12.5	29.0	4.0	13.8	30.0	1.2	8.4	40.0	0.7
CC (AAAI'21)	ResNet-18	7.6	22.2	1.5	27.7	56.5	7.5	28.7	55.8	2.7	5.2	25.1	1.0
C3-GAN (ICLR'22)	–	17.9	36.0	1.5	67.8	67.0	7.9	51.2	57.0	1.9	27.6	53.0	1.4
TEMI (BMVC'23)	ViT-B/16	19.2	43.5	5.9	39.8	66.9	23.8	36.6	62.6	12.4	24.4	60.9	6.9
DivClust (CVPR'23)	ResNet-50	9.5	27.6	1.0	30.3	59.0	6.3	33.5	58.6	1.4	10.8	39.7	1.2
TAC (ICML'24)	ViT-B/32	48.7	64.8	**41.7**	53.2	73.2	65.9	61.3	76.5	61.4	34.7	64.3	25.7
IDC (NeurIPS'24)	ResNet-50	18.1	30.5	1.1	27.5	43.3	6.4	28.7	46.2	2.5	17.8	29.3	0.6
CLUDI (ICML'25)	ViT-B/16	42.6	63.7	24.9	69.7	81.5	22.7	74.1	87.3	49.7	30.2	61.3	8.9
Ours	ViT-B/16	**49.2±0.3**	**67.1±0.2**	40.5±0.4	**76.6±0.5**	**92.0±0.1**	**73.1±0.6**	**77.2±0.2**	**88.0±0.3**	**81.8±0.4**	**42.0±0.5**	**72.3±0.1**	**38.6±0.2**

Table 20: Additional coarse-grained results (ACC, NMI, ARI). MC-H error bars describe fixed-seed hardware repeats; see Appendix A.2.

Method	Backbone	CIFAR-10			CIFAR-100-20			ImageNet-10			STL-10		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SCAN (ECCV'20)	ResNet-34	88.3	79.7	77.2	41.1	58.1	29.9	–	–	–	80.9	69.8	64.6
CC (AAAI'21)	ResNet-34	79.0	70.5	63.7	38.1	33.0	24.9	89.3	85.9	82.2	85.0	76.4	72.6
ProPos (TPAMI'22)	ResNet-34	94.3	88.6	88.4	35.8	46.4	26.5	95.6	89.6	90.6	86.7	75.8	73.7
SPICE (TIP'22)	ResNet	92.6	86.5	85.2	–	32.9	26.6	95.9	90.2	91.2	93.8	87.2	87.0
TCL (IJCV'22)	ResNet-34	88.7	81.9	78.0	36.0	51.9	27.4	89.5	87.5	83.7	86.8	79.9	75.7
SIC (AAAI'23)	ViT-B/32	–	–	–	58.3	59.3	43.9	98.2	97.0	96.1	98.1	95.3	95.9
TAC (ICML'24)	ViT-B/32	92.3	84.1	83.9	59.0	68.6	42.7	99.5	98.5	98.8	98.3	95.7	96.3
CLUDI (ICML'25)	ViT-B/16	95.3	89.6	89.8	60.4	63.8	38.9	75.0	85.3	72.4	75.1	75.8	73.4
SEIC (AAAI'26)	ViT-B/16	95.8	90.5	90.2	71.5	73.2	59.8	99.7	**99.2**	**99.4**	**99.3**	**98.0**	**98.5**
SATC (ICLR'26)	ViT-B/32	94.5	88.9	88.3	63.4	70.5	48.1	**99.8**	**99.2**	**99.4**	99.0	97.3	97.9
CAE (AAAI'26)	ViT-B/16	90.9	81.7	80.5	–	–	–	98.8	98.1	97.4	98.2	95.5	96.7
Ours	ViT-B/16	**96.7±0.1**	**91.3±0.2**	**91.0±0.1**	**73.4±0.2**	**74.1±0.3**	**61.0±0.2**	**99.8±0.1**	**99.2±0.2**	99.2±0.1	**99.3±0.4**	97.5±0.3	97.9±0.2

Table 21: Additional mixed-grained results (ACC, NMI, ARI). MC-H error bars describe fixed-seed hardware repeats; see Appendix A.2.

Method	Backbone	ImageNet-50			ImageNet-100			ImageNet-200			ImageNet-1K		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
SCAN (ECCV'20)	ResNet-50	–	82.2	66.1	–	80.8	57.6	–	77.2	47.0	–	–	–
ProPos (TPAMI'22)	ResNet-50	–	82.8	69.1	–	83.5	63.5	–	80.6	53.8	–	–	–
TCL (IJCV'22)	ResNet-50	–	–	–	–	–	–	–	–	–	37.9	67.1	26.6
SIC (AAAI'23)	ViT-B/32	75.4	81.4	67.1	67.2	77.2	55.8	57.9	73.4	46.4	47.0	77.2	34.3
TEMI (BMVC'23)	ViT-B/16	80.1	86.1	71.0	75.0	85.6	65.4	73.1	85.2	62.1	34.6	63.2	26.6
TAC (ICML'24)	ViT-B/32	78.7	83.7	67.5	63.8	68.1	57.6	59.2	70.9	50.8	58.2	79.9	43.5
CLUDI (ICML'25)	ViT-B/16	82.1	91.2	76.2	76.6	87.1	67.8	73.7	86.1	63.2	45.5	67.0	34.8
CAE (AAAI'26)	ViT-B/16	–	–	–	–	–	–	–	–	–	53.1	79.3	39.3
Ours	ViT-B/16	**86.9±0.2**	**92.4±0.1**	**78.9±0.3**	**79.4±0.2**	**88.5±0.4**	**70.5±0.2**	**75.3±0.3**	**88.3±0.1**	**66.0±0.5**	**60.2±0.3**	**81.5±0.2**	**45.0±0.4**

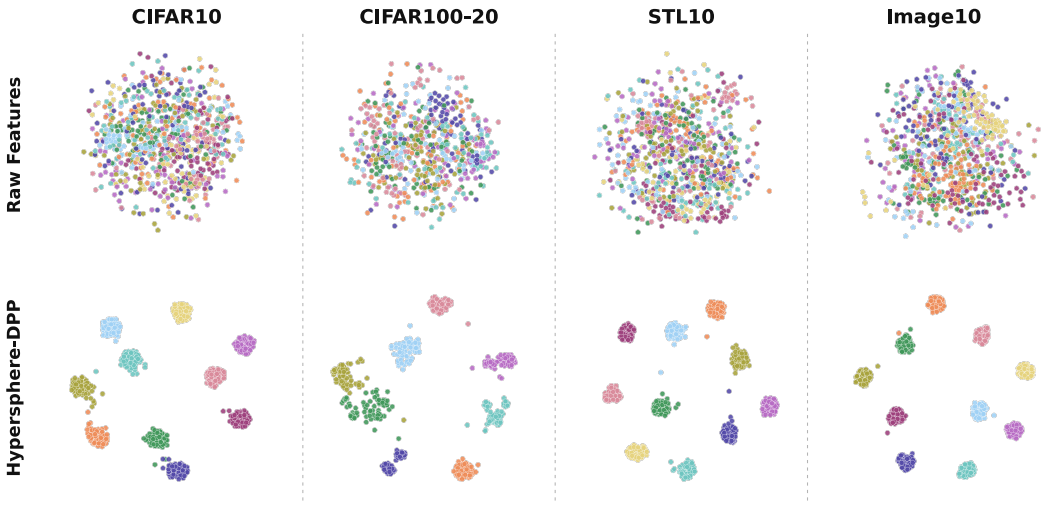


Figure 4: t-SNE on coarse-grained benchmarks (top: raw features; bottom: learned features).

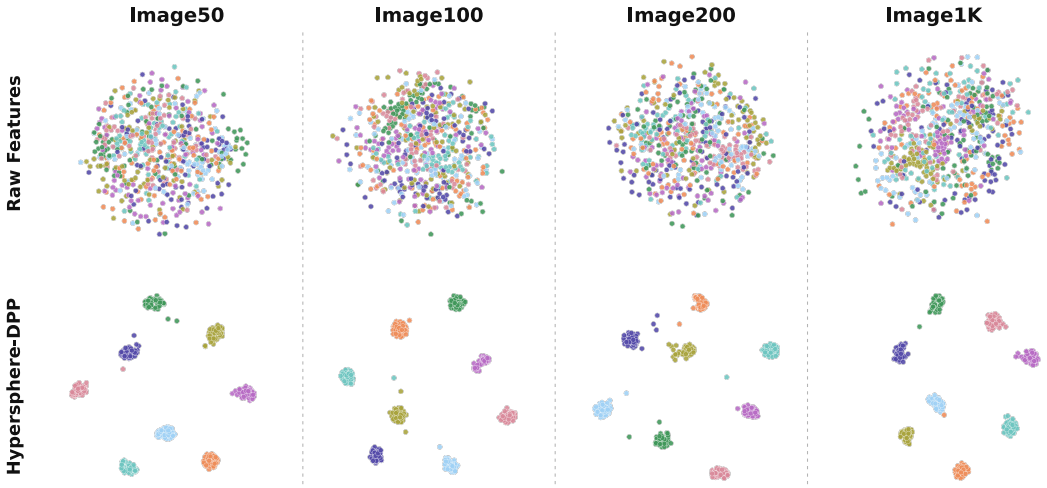


Figure 5: t-SNE on mixed-grained benchmarks (top: raw features; bottom: learned features).

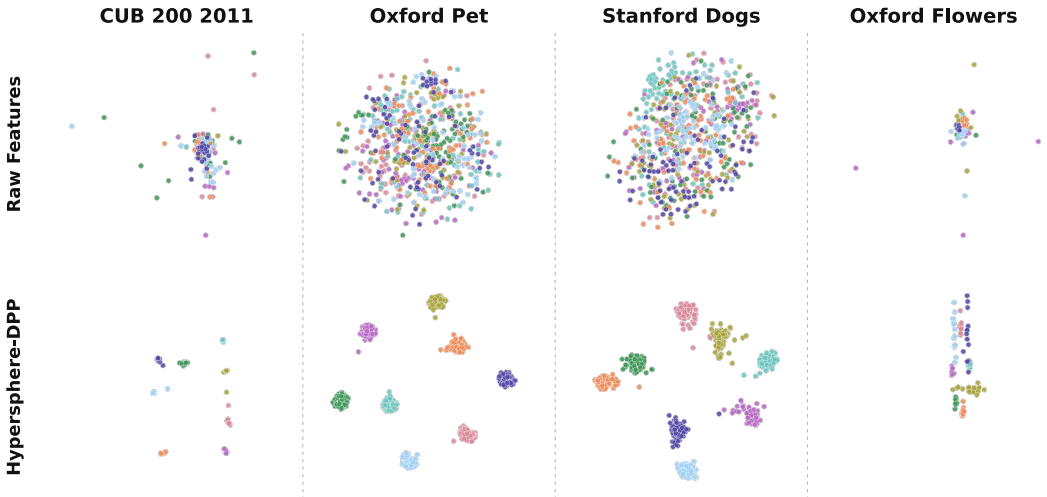


Figure 6: t-SNE on fine-grained benchmarks (top: raw features; bottom: learned features).

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: In the Methodology and Experiments of this paper, we provide a detailed elaboration and validation of the MC-H framework theory and our core contributions as proposed in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We address the limitations of our proposed method in the Conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We provide detailed theoretical proofs in the Methodology section and Appendix B.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Detailed parameters for reproducing all experimental results are provided in the Methodology section and Appendix A.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We plan to release the anonymized version of the code and data upon publication

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We provide detailed hyperparameters and the experimental environment in Appendix A.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: Appendix A.2 defines the reported fixed-seed hardware-repeat standard errors and their limitations. These are not estimates of variability across random initializations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We report the hardware and experimental environment used in Appendix A.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our work does not involve any human subjects, sensitive data, or potential risks; it fully conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: Our research prioritizes theoretical and performance-based advancements over domain-specific practical applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our work does not involve the release of models or datasets with high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets used in this work are publicly available and the corresponding references are included in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: Our work utilizes only established public datasets and does not involve the creation of any new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our work does not involve crowdsourcing or human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our work does not involve human subjects, hence IRB approval is not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were used only for language editing and did not contribute to the core theory or experimental results.

    Guidelines:

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