Part C

UL task: Clustering

CI.1

Motivation: The goal of clustering is to group data samples into clus ters such that samples within the same cluster are more similar to each other than to those in different clusters. This task is suitable for exploring the intrinsic structure of unlabeled data and identifying natural groupings or patterns within it.

Metrics:

1. **Silhouette Coefficien**t: It measures how similar each sample is to its own cluster compared to other clusters, based solely on distance information. It ranges from -1 to 1, where higher values indicate that samples are well matched to their own cluster and poorly matched to neighboring clusters.
   Motivation: This metric is unsupervised, which means it does not require ground-truth labels, making it suitable for internal evaluation when only feature similarity is known. It helps quantify cluster compactness and separation.
2. **Adjusted Rand Index (ARI)**: It compares the clustering assignments with ground-truth labels, adjusting for chance. It evaluates how consistent the predicted clusters are with the true categories, ranging from -1 (poor agreement) to 1 (perfect match). 0 means the clustering result is random.
   Motivation: This metric is supervised, which means it can be used when reference labels are available. It provides an objective measure of clustering accuracy.

CI.3

Model 1: HDBSCAN

Motivation: HDBSCAN is an extension of the DBSCAN algorithm that combines density-based clustering with a hierarchical framework. Unlike algorithms that require a fixed number of clusters (e.g., K-means) or are sensitive to global density thresholds, HDBSCAN can automatically discover clusters of variable densities and identify noise points without manual tuning of the number of clusters.

Key characteristics of HDBSCAN:

1. No need to predefine the number of clusters
2. Robust to noise and outliers
3. Ability to handle clusters of varying density
4. Scalability and efficiency with complexity $O(nlogn)$

CI.4

How HDBSCAN works: HDBSCAN extends the classical DBSCAN algorithm by transforming the problem of clustering into one of hierarchical density analysis.

Step 1 – Mutual Reachability Distance: HDBSCAN starts by defining a mutual reachability distance between each two points in the dataset,

$$d_{mreach}(a,b) = \max\{d_c(a), d_c(b), d(a,b)\}, \tag{1}$$

while core distance is defined as:

$$d_c(x) = distance_{x-k}(x, k = min_{samples}). \tag{2}$$

Step 2 – Minimum Spanning Tree (MST): Using these mutual reachability distances, HDBSCAN constructs a minimum spanning tree (MST) of the data points. The MST encodes the connectivity structure of the dataset in terms of density-based reachability.

Step 3 – Hierarchical Cluster Tree Construction: The MST is then transformed into a hierarchical cluster tree by progressively removing edges in order of decreasing mutual reachability distance (i.e., increasing density). Each edge removal may cause clusters to split into smaller subclusters. This process naturally produces a hierarchy of clusters at multiple density levels.

Step 4 – Condensed Tree and Cluster Stability Analysis: HDBSCAN condenses this hierarchy tree by pruning small clusters below a minimum cluster size and calculating a stability score for each cluster. This score measures how long a cluster persists across different density thresholds, and the longer the persistence, the more stable and meaningful the cluster is.

$$stability(C) = \sum_{x \in C} (\lambda^C_{leave}(x) - \lambda^C_{join}(x)) \tag{3}$$

Step 5 – Cluster Selection via EOM (Excess of Mass): The Excess of Mass (EOM) algorithm selects the final flat clustering by maximizing total cluster stability (densest and most persistent) under the constraint that clusters do not overlap.

Optimization method: EOM is a stability-based optimization method, aiming to maximize overall cluster persistence across different density levels. This corresponds to an optimization problem of selecting clusters that maximize total stability mass (EOM). HDBSCAN optimizes cluster selection in a hierarchical density space, rather than through iterative parameter updates.

$$\text{maximize} \sum_{C_i \in S} stability(C_i) \tag{4}$$

CI.5

Results for HDBSCAN:

min_cluster_size	min_samples	Clusters	Clustered Samples	Clustered	Silhouette	ARI
3	2	68	341	85.25%	0.2305	0.3957
5	2	39	293	73.25%	0.1982	0.2508
5	3	30	236	59.00%	0.2259	0.1201

We applied **HDBSCAN** on the **Olivetti Faces** dataset (**400 samples**, **40 ground-truth subjects**) to study how its clustering behavior changes with different hyperparameters.

Overall, **HDBSCAN** produced varying results depending on its **density parameters**. Smaller *min_cluster_size* values (e.g., *3*) yielded **more clusters** and **higher ARI** (≈ 0.40), while larger values produced **fewer**, **more stable clusters** but **lower coverage** and **ARI**.
Silhouette scores (≈ 0.2) remained moderate, suggesting reasonable but imperfect separation in the **high-dimensional facial feature space**.
The **ARI <** 0.4 indicates that purely density-based methods still struggle with **high-dimensional, facial image data**.

We tried to use **PCA** to reduce the dimension of the raw data and extract features. However, the **clustering performance did not improve**: both **Silhouette** and **ARI** scores remained similar to or worse than those obtained from the raw data.

This outcome suggests that **PCA failed to capture the semantic features** necessary for distinguishing different identities.
As a *linear method*, **PCA** preserves **global variance** but not the **non-linear manifold structure** of facial images.
Consequently, the resulting **lower-dimensional representation** does not provide a more meaningful structure for **density-based clustering methods** such as **HDBSCAN**.

CI.6

In summary, the results suggest that **density-based clustering methods**, such as **HDBSCAN**, are **not well-suited** for **raw, high-dimensional facial image data**.
The **pixel space** of face images lacks **distinct density boundaries** and exhibits **strong non-linear and continuous variations**, which violate the **core assumptions** of density-based clustering.

To make **HDBSCAN** more effective for this type of data, it is necessary to first obtain a **more meaningful representation** through **non-linear dimensionality reduction** or **deep feature extraction**.
These approaches can transform the **original pixel data** into a **lower-dimensional manifold** where samples from the same identity are **closer together**, thereby providing a **feature space** that better reflects **semantic similarity** and allows **HDBSCAN** to form **more coherent clusters**.

UL Task: Spectral Clustering

Motivation

Group Olivetti face images into identity-consistent clusters without labels. The dataset is high-dimensional and lies on a curved manifold (pose, lighting, expression). Clustering exposes this intrinsic structure directly from features.

Datasets (Design Choice & Rationale)

Olivetti Faces (real-world, images):
400 images (64×64 → 4096-D). High-D, small-n; benefits from PCA and is a realistic testbed for face embeddings.

Model fit:

- **HDBSCAN:** after PCA (≈100–200), can find identity pockets and mark outliers (`min_cluster_size` tunes strictness).
- **OPTICS:** density ordering shows structure; reasonable on PCA features.
- **Spectral (ours):** with `affinity='nearest_neighbors'` often recovers ≈40 subjects on PCA features; captures non-linear separations.

(We initially considered the generated Two-Moons dataset to illustrate nonconvex structure, but we ultimately report only Olivetti for a purely real-world evaluation.)

Metrics

1. **Silhouette Coefficient (unsupervised)**
   Average "closer to own cluster than the next best" score using distances in feature space; range [−1,1], higher is better.
   - **Why here:** lets us tune Spectral's `n_neighbors` and `n_clusters` without labels on the PCA space.
   - **Caveat:** sensitive to cluster-size imbalance and large *k*; we report overall values.
2. **Adjusted Rand Index (ARI) (external)**
   Chance-adjusted agreement with ground-truth identities; range [−1,1] with 0 ≈ random and 1 = perfect.
   - **Why here:** Olivetti ships labels; ARI quantifies identity consistency post-hoc and enables a fair comparison to density methods.
   - **Caveat:** ARI tends to prefer matching the true number of clusters; we sweep *k* around ≈40 and report both ARI and Silhouette.

CI.3 — Model: Spectral Clustering

Motivation

Spectral converts clustering into graph partitioning. It builds a k-nearest-neighbors (kNN) similarity graph and clusters in the space of the graph Laplacian's leading eigenvectors, which captures non-linear structure (pose/lighting/expression) better than centroid-based models.

Key Characteristics

1. Handles non-linear cluster geometry via a neighborhood graph.
2. Less biased toward spherical clusters than k-means/GMM; tolerates shape/size imbalance.
3. Requires `n_clusters` — useful here because prior knowledge is ≈40 subjects.

CI.4 — How Spectral Clustering Works

(**Method, Optimization, Implementation**)

Step 1 – Affinity Graph:
Build sparse kNN graph **W** (edges between nearest neighbors).

Step 2 – Normalized Laplacian:

$$L_{sym} = I - D^{-1/2} W D^{-1/2} \tag{5}$$

with $D_{ii} = \sum_j W_{ij}$.

Step 3 – Spectral Embedding:
Take the *k* eigenvectors of L_{sym} with smallest eigenvalues; row-normalize to get $U \in \mathbb{R}^{n\times k}$.

Step 4 – Discretization:
Run k-means on rows of **U** to obtain *k* clusters.

Optimization view:
Spectral is the relaxed solution of Normalized Cuts

$$Ncut(A_1, \ldots, A_k) = \sum_{r=1}^{k} \frac{cut(A_r, \bar{A}_r)}{assoc(A_r, V)} \tag{6}$$

The exact minimization is NP-hard; the spectral relaxation reduces it to an eigenproblem on the Laplacian, then k-means discretizes the relaxed solution.

Our implementation (scikit-learn):

Preprocessing: $PCA(100, whiten = True)$ on flattened images (denoise + stable distances).

Model: $SpectralClustering(affinity =' nearest\_neighbors', assign\_labels =' kmeans')$

Swept hyper-params:

$n\_clusters \in 30, 35, 38, 40, 42, 45, 50, n\_neighbors \in 8, 10, 12, 15, 20, random\_state = 42.$

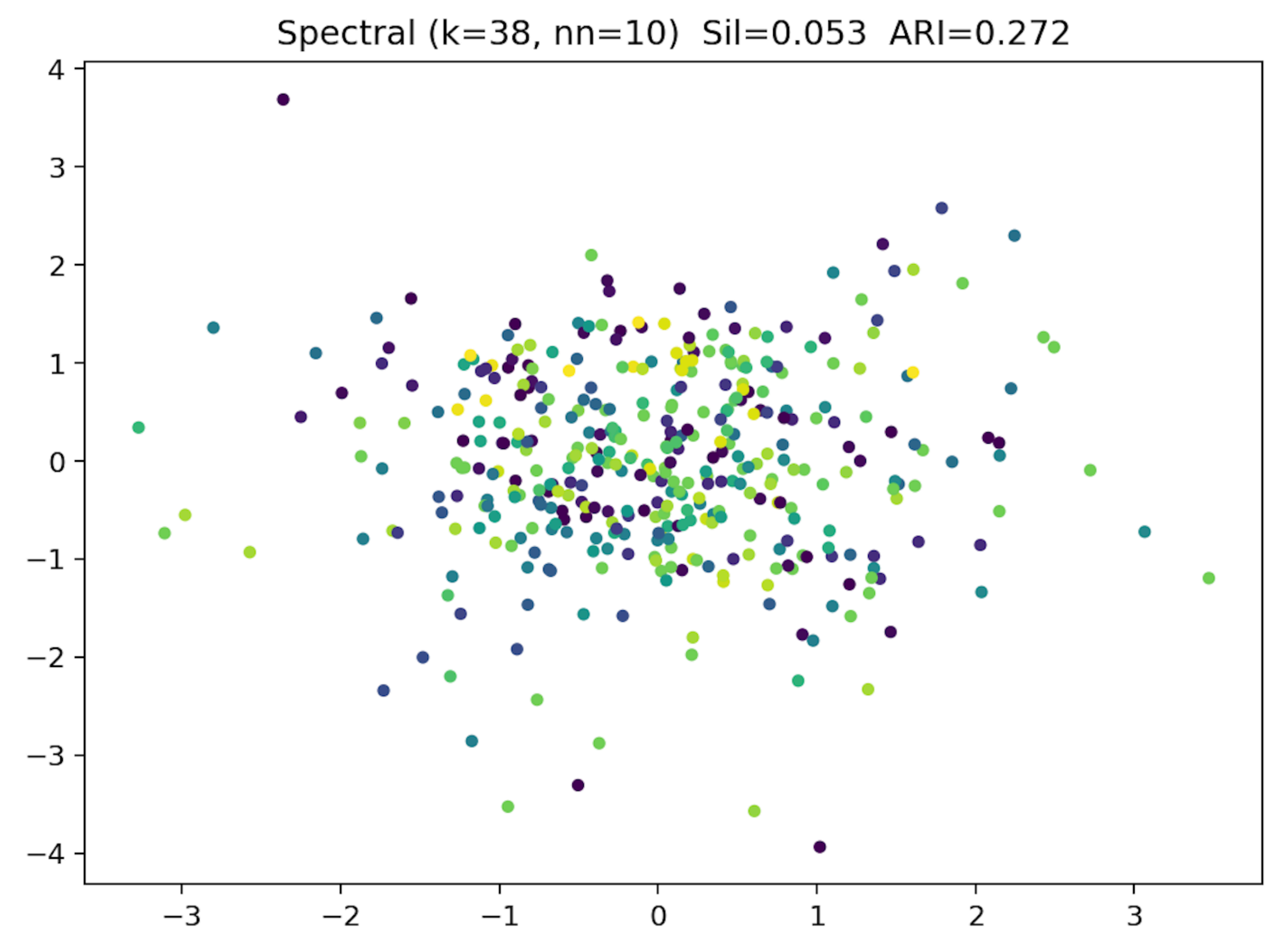
CI.5 — Results for Spectral Clustering (Olivetti Faces)

Input: n=400, 40 ground-truth subjects
Features used: $PCA(100) \to X \in R^{100 \cdot 400}$

Results

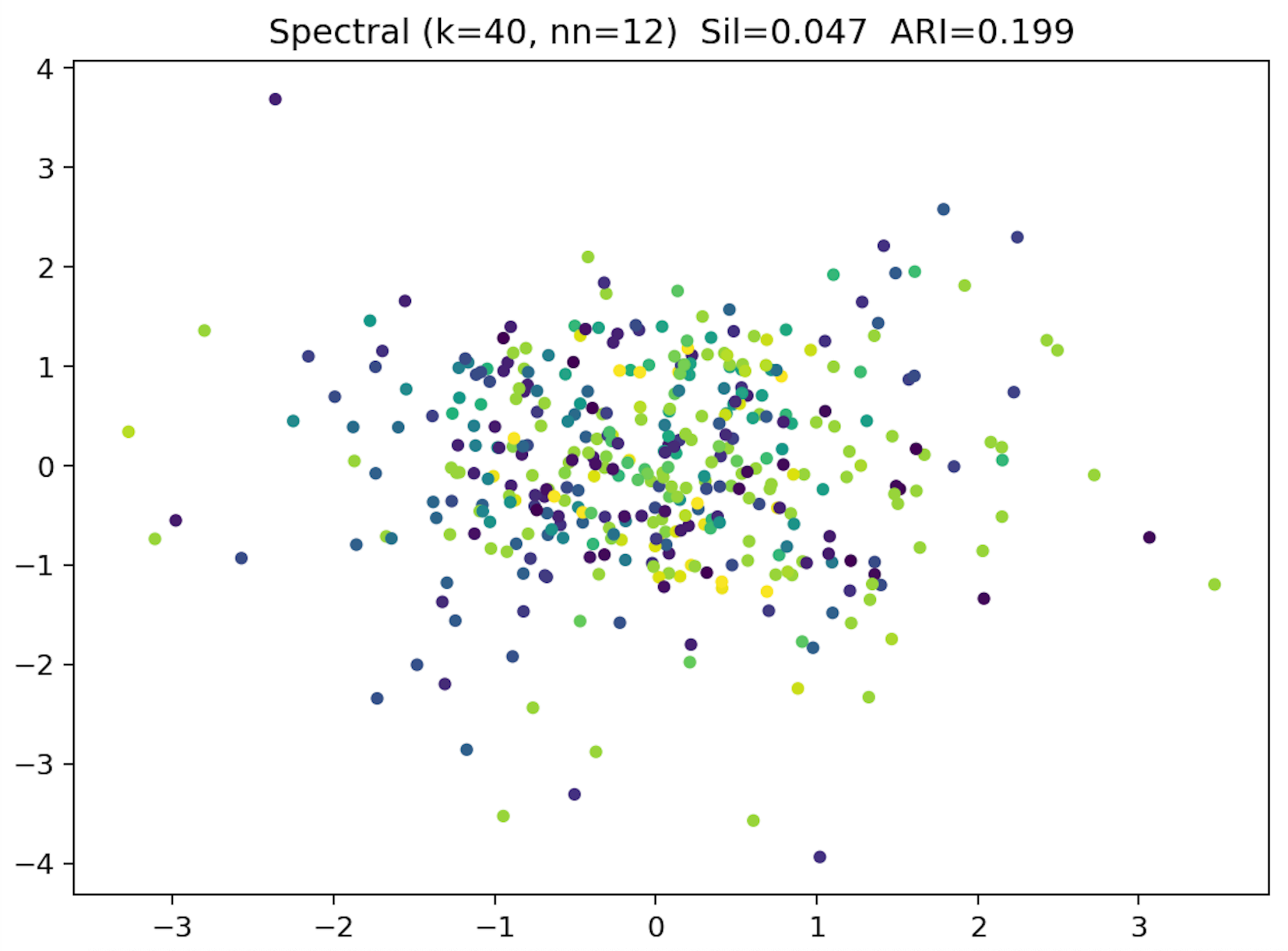
Parameters: `n_clusters=38`, `n_neighbors=10`
Evaluation metrics:

- `silhouette_score: 0.053`
- `adjusted_rand_score: 0.272` *(best ARI in our sweep)*



Parameters: `n_clusters=40`, `n_neighbors=12`
Evaluation metrics:

- `silhouette_score: 0.047`
- `adjusted_rand_score: 0.199`



- **Best setting by ARI:** (k=38, nn=10) slightly under-specifies the nominal 40 identities, avoiding over-splitting and yielding the best identity consistency (ARI ≈ 0.272).
- **Silhouette magnitudes:** low (~0.05) across settings ⇒ clusters are partly overlapping in PCA(100) with Euclidean distance; identity separation exists but is weak in this space.
- **Effect of n_neighbors:** Too small fragments the graph; too large over-smooths and merges identities. The sweet spot is 10–12 neighbors.
- **Full assignment:** Spectral labels every sample (no "noise"), which simplifies Silhouette reporting and contrasts with density methods that may leave hard cases as noise.

CI.6 — Summary & Takeaways (Spectral on Olivetti)

Spectral Clustering on PCA(100) features provides a principled non-linear partition with moderate identity consistency (best ARI ≈ 0.272, Silhouette ≈ 0.05).
The model is most effective with a kNN affinity (≈10–12 neighbors) and *k* near—but not necessarily equal to—the subject count.The low Silhouette indicates that Euclidean geometry on PCA(100) remains blurry for identities; stronger embeddings (more PCs, non-linear DR,

or learned features) are likely to improve separation.

Overall, Spectral offers a clean, deterministic baseline for comparison against density-based methods on real-world face data.