MAPPING-BASED ADAPTIVELY WEIGHTED ENSEMBLE CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensemble clustering fuses multiple base clusterings into a partition that is more accurate and more robust than any single one. Existing methods, however, fix the weights before the fusion and apply them at a single granularity, split representation learning and partition solving into two stages, are prone to trivial solutions, and rely on the expensive $n \times n$ co-association matrix. We therefore propose Mapping-based Adaptively Weighted Ensemble Clustering (MAWEC). A continuous weight rates each base clustering and a mapping matrix merges each base cluster into a target cluster, so the two granularities are optimized jointly. Consistency learning, structured learning and weight learning are coupled through one soft label matrix, whose row-wise maximum gives the labels directly. A structural regularizer promotes row sparsity and column balance, and a weight regularizer keeps the weights from collapsing, so three trivial solutions are excluded. Since the consensus is carried by the $n \times m$ association matrix, the fusion is linear in the number of samples. Experiments on eight real-world datasets show that MAWEC attains the highest ACC on all of them and the shortest running time.

1 INTRODUCTION

Clustering partitions a dataset by the similarity between samples and reveals its latent structure without manual annotation (Oyewole & Thopil, 2023), and has been applied to image segmentation (Minaee et al., 2022), text mining (Allahyari et al., 2017), recommendation systems (Xu et al., 2023) and bioinformatics (Zou et al., 2020). However, every algorithm rests on its own assumptions and fits only one kind of structure: k-means (Lloyd, 1982) favors convex clusters and often fails on non-convex shapes (Bai et al., 2020), while spectral clustering is sensitive to the similarity graph (Ng et al., 2001). The result also varies with the initialization, and choosing a suitable algorithm without prior knowledge is difficult (Hao et al., 2024), so a single algorithm rarely performs stably on diverse real-world data.

Ensemble clustering relieves these limitations (Golalipour et al., 2021). It generates diverse base clusterings by varying the algorithm, the parameters or the feature subset (Fern & Brodley, 2003), and fuses them by a consensus function. Since the base clusterings make errors at different places, their fusion is usually more accurate and more robust (Topchy et al., 2005). The fusion uses only the labels, so the method suits any data type (Boongoen & Iam-On, 2018) and has been applied to medical diagnostics (Greene et al., 2004), internet security (Zhuang et al., 2012) and remote sensing (Zhang et al., 2022).

Many consensus functions have been proposed, yet four limitations remain. First, the weights are fixed before the fusion by a preset criterion (Huang et al., 2018) and act on a single granularity (Xu & Ding, 2021). Second, the two-stage pipeline loses information, since the intermediate representation cannot be revised once formed (Jia et al., 2024) and the discretization brings a further loss (Wang et al., 2016). Third, the models are prone to trivial solutions (Yang et al., 2025a), namely uniform memberships, imbalanced cluster sizes, and weights collapsing onto a few base clusterings. Fourth, the $n \times n$ co-association matrix grows quadratically with the number of samples (Huang et al., 2019). Existing methods usually relieve only one or two of these problems.

We therefore propose Mapping-based Adaptively Weighted Ensemble Clustering (MAWEC), which replaces the co-association matrix by a base-cluster mapping matrix, and learns the weights, the mapping and the labels in one objective. The main contributions are as follows:

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

1. We propose an adaptive weighting mechanism with two granularities. A continuous weight rates each base clustering and a mapping matrix reassigns each base cluster, and both are updated together with the clustering result.
2. We build a unified optimization framework. The three kinds of learning are coupled through one soft label matrix, whose row-wise arg max gives the labels directly, so no two-stage processing is needed.
3. We design two regularizers that rule out three trivial solutions. The structural regularizer promotes the row sparsity and the column balance of the soft label matrix, and the weight regularizer keeps the weights spread out.
4. We reduce the complexity through the mapping matrix. The consensus is modeled on the $n \times m$ association matrix, so time and space become linear.

2 Related Work

According to how the base clusterings are represented and fused, existing methods mainly fall into three categories: pairwise-similarity-based, graph-partitioning-based and unified-optimization-based (Bai et al., 2019). A further group aligns the labels of different base clusterings and fuses them by voting (Zhou & Tang, 2006); such methods are cheap, but require every base clustering to have nearly the target number of clusters.

Pairwise-similarity-based methods count how often two samples fall into the same cluster and cluster the resulting $n \times n$ co-association matrix, as in EAC (Fred & Jain, 2005). PTA (Huang et al., 2016) redefines the similarity by random-walk trajectories, LWEA (Huang et al., 2018) weights it locally by an entropy criterion, and ECPCS (Huang et al., 2021) propagates cluster-wise similarities. They are simple and stable, but the matrix grows quadratically and is never revised once generated, so its quality caps the final result.

Graph-partitioning-based methods take samples, clusters or both as graph nodes and solve ensemble clustering by graph cuts (Fern & Brodley, 2004). LWGP (Huang et al., 2018) applies local weighting to graph partitioning, CEHM (Li et al., 2025) discovers k-hyperedge medoids, and MDEC-BG and MDEC-SC (Huang et al., 2022) enrich the diversity with many metrics and subspaces. They find clusters of arbitrary shape, but graph construction and graph partitioning remain separate stages, and spectral methods still need a discretization.

Unified-optimization-based methods write weight learning, structured learning and partition solving into one objective and solve them by alternating optimization (Bezdek & Hathaway, 2002). SCAC (Yang et al., 2025b) refines the weights to the cluster level and solves the discrete labels by coordinate descent, and CDEC (Zhang et al., 2025b) adds a balance term. They relieve the fixed weights and the two-stage pipeline, but their weights still act on only one granularity, either the partition or the cluster level, each trivial solution needs its own constraint, and most of them still take an $n \times n$ matrix as the core structure. MAWEC differs on all three points.

3 Methodology

3.1 Generation of Base Clusterings

Let $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n \times d}$ be the data matrix of n samples with d features, c the target number of clusters and T the number of base clusterings. To make the base clusterings differ, we randomize both the granularity of the partition and the feature view. The number of clusters m_k of the k-th base clustering ($k = 1, \ldots, T$) is drawn from an integer interval:

$$m_k \sim \mathcal{U}\{c,\ c+1,\ \ldots,\ \lfloor \sqrt{n} \rfloor\}, \tag{1}$$

where $\mathcal{U}\{\cdot\}$ is the uniform distribution over a set of integers. The lower bound keeps a base clustering from being coarser than the target partition, and the upper bound balances quality and diversity (Xie et al., 2024). When the interval is empty, we replace it by $\{c, \ldots, c+20\}$. We then draw a feature subset at random, with a sampling ratio drawn uniformly from $[0.8, 1.0]$, run k-means (Arthur

& Vassilvitskii, 2007) on this subset, and write the resulting hard partition as a one-hot indicator matrix $Y_k \in \{0,1\}^{n\times m_k}$:

$$(Y_k)_{ij} = \begin{cases} 1, & \text{if } x_i \text{ belongs to the } j\text{-th cluster of the } k\text{-th base clustering,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The clusters inside a base clustering are called base clusters, and their total number is

$$m = \sum_{k=1}^{T} m_k. \tag{3}$$

By (2), every row of Y_k has exactly one entry equal to 1, so $Y_k\mathbf{1} = \mathbf{1}$ and $Y_k \geq 0$, where $\mathbf{1}$ is the all-ones column vector whose dimension follows from the context.

The randomness makes the base clusterings uneven in quality, and the poor ones drag down the consensus if all are treated equally. We therefore introduce a weight vector $\alpha \in \mathbb{R}^{T\times 1}$, whose k-th component is the weight of the k-th base clustering, and require

$$\alpha \geq 0, \qquad \alpha^\top \mathbf{1} = 1. \tag{4}$$

Here α is learned from the data during the optimization rather than preset, and acts on a whole partition. Since the base clusters inside one base clustering are not equally reliable either, the next subsection adds a second granularity at the base cluster level.

3.2 Clustering Model

In Section 3.1, each base clustering gives a partition in its own cluster space, and no natural correspondence exists among these spaces. To unify them into the c target clusters, we introduce a mapping matrix $Z_k \in \{0,1\}^{m_k\times c}$ for the k-th base clustering and require $Z_k \in \mathrm{Ind}$, where Ind is the set of indicator matrices with exactly one entry equal to 1 in each row. Here $(Z_k)_{jl} = 1$ means that the j-th base cluster is merged into the l-th target cluster. Let $Y \in \mathbb{R}^{n\times c}$ denote the soft label matrix of the samples. We then propose the following model:

$$\min_{\substack{Z_k\in\mathrm{Ind},\, Y\mathbf{1}=\mathbf{1},\, Y\geq 0 \\ \alpha\geq 0,\, \alpha^\top\mathbf{1}=1}} \left\| Y - \sum_{k=1}^{T} \alpha_k Y_k Z_k \right\|_F^2 - \lambda \operatorname{Tr}\big(Y^\top H Y\big) + \gamma\|\alpha\|_2^2, \tag{5}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\lambda > 0$ and $\gamma > 0$ are regularization coefficients, and H is the centering matrix

$$H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top, \tag{6}$$

which satisfies $H = H^\top = H^2$ and is positive semidefinite.

We explain (5) starting from $Y_k Z_k$. This product is the sample partition obtained when each of the m_k base clusters of the k-th base clustering is assigned as a whole to one target cluster, which amounts to one round of label mapping (Huang et al., 2017), and $\sum_k \alpha_k Y_k Z_k$ is the weighted consensus of the T rounds.

The weight vector α and the mapping matrices Z_k form the two granularities: α_k is a continuous weight on a whole base clustering and each row of Z_k is a discrete assignment of a single base cluster, so the contribution of the j-th base cluster of the k-th base clustering toward the l-th target cluster is $\alpha_k (Z_k)_{jl}$. The three terms of (5) carry out consistency learning, structured learning and weight learning, and Y, Z_k and α all enter the fidelity term, so the three kinds of learning are coupled through the same Y. Since Y is itself the final soft label matrix, the class of the i-th sample is read off the largest entry of its row:

$$\ell_i = \operatorname*{arg\,max}_{1\leq j\leq c} Y_{ij}, \qquad i = 1, 2, \ldots, n, \tag{7}$$

so no two-stage pipeline is needed. Figure 1 illustrates the framework of MAWEC, which consists of five stages: base clustering generation, adaptive mapping, adaptive weighting, unified optimization, and output.

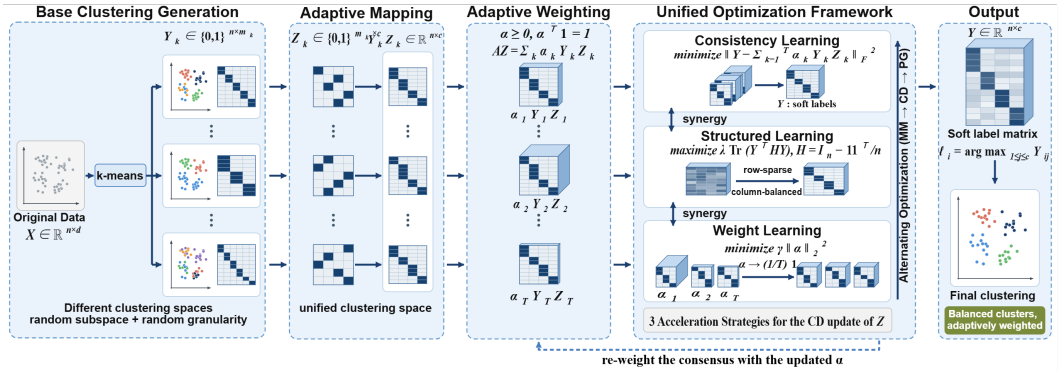


Figure 1: Framework of MAWEC. Stage 1 generates T base clusterings by k-means on random subspaces with random granularities, each encoded as a one-hot matrix Y_k. Stage 2 maps the heterogeneous cluster spaces into a unified one through Z_k. Stage 3 weights each base clustering by α_k and forms the weighted consensus. Stage 4 optimizes the three kinds of learning in one framework, updating Y, Z and α by MM, CD and PG. Stage 5 reads the labels off the row-wise maximum of Y.

For brevity, the T base clusterings and their weights are concatenated as

$$A = [\alpha_1 Y_1, \ \ldots, \ \alpha_T Y_T] \in \mathbb{R}^{n \times m}, \quad Z = [Z_1^\top, \ldots, Z_T^\top]^\top \in \{0,1\}^{m \times c}, \tag{8}$$

so that $\sum_k \alpha_k Y_k Z_k = AZ$ and (5) becomes

$$\min_{\substack{Z \in \mathrm{Ind},\ Y\mathbf{1}=\mathbf{1},\ Y \geq 0 \\ \alpha \geq 0,\ \alpha^\top \mathbf{1}=1}} \mathcal{L}(Y, Z, \alpha) = \|Y - AZ\|_F^2 - \lambda \operatorname{Tr}\big(Y^\top H Y\big) + \gamma \|\alpha\|_2^2, \tag{9}$$

where $\mathcal{L}$ denotes the objective function.

From the row-stochasticity of Y_k and (4) we get $A\mathbf{1} = \mathbf{1}$ and $A \geq 0$, so A is the association matrix between the samples and the base clusters (Li et al., 2012). The weighted co-association matrix $AA^\top = \sum_k \alpha_k^2 Y_k Y_k^\top$ carries no information beyond A, yet is quadratic in n in both time and memory, whereas only A and Z appear in (9). Moreover, for $n \geq c^2$, (1) and (3) give $m \leq T\sqrt{n}$, hence $m^2 \leq T^2 n$, and the scale of the model drops from quadratic to linear (Zhang et al., 2025a).

The two regularizers are needed because $\|Y - AZ\|_F^2$ alone is driven to zero by $Y = AZ$ for any Z and α, so it rules out none of the three trivial solutions: memberships close to $1/c$, severely imbalanced cluster sizes, and weights kept only on a few base clusterings.

3.3 Analysis of the Regularization Terms

This subsection shows how the two regularizers rule out the three trivial solutions. The structural regularizer is characterized by the following theorem.

Theorem 1. *Let $Y \in \mathbb{R}^{n \times c}$ satisfy $Y\mathbf{1} = \mathbf{1}$ and $Y \geq 0$. Then*

$$\operatorname{Tr}\big(Y^\top H Y\big) \leq \frac{n(c-1)}{c}, \tag{10}$$

with equality if and only if Y is a 0–1 indicator matrix and every cluster contains exactly n/c samples.

The proof is given in Appendix A. Both of the first two trivial solutions give $\operatorname{Tr}(Y^\top H Y) = 0$: setting $Y_{ij} = 1/c$, or placing all samples in one cluster. Minimizing (9) drives this term toward the maximum in (10) instead, which produces row sparsity and column balance at once, so the membership of each sample concentrates on one cluster and the cluster sizes do not differ wildly.

Unlike the structural regularizer, which concentrates Y, the weight regularizer spreads α out. Expanding the square with $\alpha^\top \mathbf{1} = 1$ gives

$$\|\alpha\|_2^2 = \left\|\alpha - \tfrac{1}{T}\mathbf{1}\right\|_2^2 + \frac{1}{T}, \tag{11}$$

where the second term is a constant, so minimizing $\|\alpha\|_2^2$ amounts to minimizing the distance from α to the equal-weight vector. Together with $\|\alpha\|_2^2 \le \|\alpha\|_1 \|\alpha\|_\infty \le 1$, we obtain

$$\frac{1}{T} \le \|\alpha\|_2^2 \le 1. \tag{12}$$

The lower bound is attained at equal weights and the upper bound at a vertex of the simplex, where only one base clustering is retained, so penalizing $\|\alpha\|_2^2$ excludes the third trivial solution.

It also makes the weight subproblem well posed. With $\gamma = 0$ the coefficient matrix is the Gram matrix G of Section 3.4, which is only positive semidefinite, so the optimum need not be unique and can collapse onto a few base clusterings; with $\gamma > 0$ it is strictly positive definite and the subproblem has a unique solution.

3.4 Optimization Algorithm

The objective (9) has three blocks, Y, Z and α. We solve it by alternating optimization, updating one block while the others are fixed. The superscript t denotes the outer iteration.

Updating Y. Dropping the constants independent of Y gives

$$\mathcal{L}(Y) = \|Y - AZ\|_F^2 - \lambda \operatorname{Tr}\big(Y^\top H Y\big). \tag{13}$$

The quadratic part of (13) is $\operatorname{Tr}\big(Y^\top (I_n - \lambda H) Y\big)$, and the eigenvalues of $I_n - \lambda H$ are $1-\lambda$ and 1, so (13) is nonconvex when $\lambda > 1$, and H couples the rows of Y. We therefore adopt majorization–minimization (MM) (Sun et al., 2016). Since H is positive semidefinite, $\operatorname{Tr}(Y^\top H Y)$ is convex with gradient $2HY$; replacing it by its first-order lower bound at Y^t and completing the square give a surrogate that touches $\mathcal{L}(Y)$ at Y^t:

$$g(Y, Y^t) = \big\|Y - \big(AZ + \lambda H Y^t\big)\big\|_F^2 + \text{const}, \tag{14}$$

so the subproblem in Y becomes

$$Y^{t+1} = \underset{Y\mathbf{1}=\mathbf{1},\ Y \ge 0}{\arg\min} \ \big\|Y - \big(AZ + \lambda H Y^t\big)\big\|_F^2. \tag{15}$$

The constraints are separable across rows, so (15) projects each row of $AZ + \lambda H Y^t$ onto the probability simplex $\Delta = \{x : \sum_j x_j = 1,\ x_j \ge 0\}$. For a single row v, the projection has the closed form (Duchi et al., 2008)

$$x_j = \max(v_j - \theta,\, 0), \qquad j = 1, \dots, c, \tag{16}$$

where $v_{(1)} \ge \cdots \ge v_{(c)}$ are the entries of v in descending order, $\theta_r = \frac{1}{r}\big(\sum_{j \le r} v_{(j)} - 1\big)$, ρ is the largest r with $v_{(r)} - \theta_r > 0$, and $\theta = \theta_\rho$. We denote this operator by $\Pi_\Delta(\cdot)$. In practice H need not be formed explicitly, since $HY^t = Y^t - \frac{1}{n}\mathbf{1}(\mathbf{1}^\top Y^t)$ costs $O(nc)$.

Updating Z. Let $Q = A^\top A \in \mathbb{R}^{m \times m}$ and $R = A^\top Y \in \mathbb{R}^{m \times c}$. Dropping the constants gives

$$\min_{Z \in \mathrm{Ind}} \ F(Z) = \operatorname{Tr}\big(Z^\top Q Z\big) - 2 \operatorname{Tr}\big(Z^\top R\big). \tag{17}$$

The (k,l)-th block of Q is $\alpha_k \alpha_l Y_k^\top Y_l$, the weighted contingency table between two base clusterings, so the partition-level weights enter the update at the base cluster level. Since each row of A has T nonzero entries, Q costs $O(nT^2)$. Problem (17) is combinatorial, and we solve it row by row with coordinate descent (CD) (Wright, 2015). Incremental evaluation and caching, detailed in Appendix B, reduce a single row update to $O(m + c)$ and one epoch to $O\big(m(m + c)\big)$. Every row takes its minimum and Z has finitely many feasible values, so CD stops within finitely many epochs; we stop when $F(Z)$ changes by less than 10^{-6}.

Updating α. With Z fixed, the set $\{Y_k Z_k\}_{k=1}^T$ is known. The part of the objective that involves α is then quadratic, with coefficient matrix $G + \gamma I_T$ and linear term h, where

$$G_{kl} = \operatorname{Tr}\big((Y_k Z_k)^\top (Y_l Z_l)\big), \qquad h_k = \operatorname{Tr}\big(Y^\top Y_k Z_k\big). \tag{18}$$

Since Y_k and Z_k are one-hot, G and h cost $O(T^2 n)$ and $O(Tn)$ with no $n \times c$ intermediate matrix. The matrix G is a Gram matrix, so $G + \gamma I_T$ is strictly positive definite and the subproblem

$$\alpha^{t+1} = \underset{\alpha \ge 0,\ \alpha^\top \mathbf{1} = 1}{\arg\min} \ \alpha^\top \big(G + \gamma I_T\big)\alpha - 2h^\top \alpha \tag{19}$$

is a strictly convex quadratic program on the simplex. Since T is of the order of tens, projected gradient descent (PG) (Lin, 2007) solves it efficiently:

$$\alpha^{(s+1)} = \Pi_{\Delta}\left(\alpha^{(s)} - \frac{1}{L}\left[\left(G + \gamma I_T\right)\alpha^{(s)} - h\right]\right), \qquad L = \left\|G + \gamma I_T\right\|_2, \tag{20}$$

where s is the inner index, $\alpha^{(0)} = \alpha^t$ and $1/L$ is the step size. Each step costs $O(T^2)$. A larger h_k means better agreement with the current labels and pushes α_k up. The learned α re-weights A and enters the next update of Y and Z, which closes the loop between the two levels.

Initialization and stopping. We set $\alpha^0 = \frac{1}{T}\mathbf{1}$, the minimizer of the third term, obtain Z^0 by k-means on the m base cluster centers, each computed in the full feature space, and set $Y^0 = A^0 Z^0$. The outer loop stops when $|\mathcal{L}^{t+1} - \mathcal{L}^t| < \varepsilon$ or after $t_{\max}$ iterations; Algorithm 1 in Appendix C summarizes the procedure.

3.5 Complexity Analysis

Generating the base clusterings runs k-means T times, which costs $O(ndm)$ in total per k-means iteration. The three subproblems are broken down in Appendix D, and each outer iteration costs

$$O\big(nT^2 + Tnc + nc\log c + m(m+c)\big). \tag{21}$$

Since $m^2 \le T^2 n$, every term in (21) is linear in n, so the fusion costs $O(n)$ per iteration when T and c are far smaller than n.

In contrast, co-association methods need $O(n^2 T)$ time and $O(n^2)$ memory merely to build the matrix, plus $O(n^3)$ or $O(n^2 \log n)$ for the subsequent spectral or hierarchical clustering (Murtagh & Contreras, 2012). In MAWEC all quadratic forms occur on Q and G, and no $O(n^2)$ quantity is ever formed.

3.6 Convergence Analysis

We write $\mathcal{L}^t = \mathcal{L}(Y^t, Z^t, \alpha^t)$ for the objective value at the end of the t-th outer iteration.

Theorem 2. *The sequence $\{\mathcal{L}^t\}$ generated by the alternating optimization of Section 3.4 is monotonically non-increasing and bounded below by $-\lambda n(c-1)/c + \gamma/T$. It therefore converges.*

The proof is given in Appendix E. No block update increases the objective: Y is handled by MM, every row of Z takes its minimum, and PG starts from α^t. The lower bound comes from Theorem 1 and (12), the same two inequalities that rule out the trivial solutions. In practice the objective levels off within about eight iterations.

4 Experiments

4.1 Performance Comparison

We evaluate MAWEC on eight real-world datasets (Asuncion & Newman, 2007): Control, Segment, MnistData_05, Waveform, Texture, Odr, Isolet and Pendigits, with 600 to 10,992 samples, 16 to 784 dimensions and 3 to 26 classes (Appendix F). It is compared with LWEA, LWGP (Huang et al., 2018), ECPCS (Huang et al., 2021), MDEC-BG, MDEC-SC (Huang et al., 2022), SCAC (Yang et al., 2025b), CEHM (Li et al., 2025) and CDEC (Zhang et al., 2025b), which cover the three paradigms of Section 2, under the four metrics ACC (Cai et al., 2005), NMI (Strehl & Ghosh, 2002), Purity (Manning et al., 2008) and ARI (Hubert & Arabie, 1985), where a higher value is better. All methods run in MATLAB R2024a on an AMD Ryzen 5 7535U CPU with 16 GB of memory, and the compared ones use the parameters recommended in their papers. All of them share the same base clusterings with $T = 50$, so the reported time excludes this common stage, and every experiment is repeated 20 times. Table 1 gives the clustering quality and the running time.

In terms of quality, MAWEC attains the highest ACC on all eight datasets and the best or second best NMI, Purity and ARI on most of them, because the two-granularity weighting suppresses unreliable base clusterings and base clusters at once, the three kinds of learning share one soft label

Table 1: Clustering performance (mean ± std %) and running time (mean seconds) on eight real-world datasets. For each metric the best result is in bold and the second best is underlined; for Time the shortest is in bold and the second shortest is underlined.

Dataset	Metric	LWEA	LWGP	ECPCS	MDEC-BG	MDEC-SC	SCAC	CEHM	CDEC	MAWEC
Control	ACC	59.9±4.6	69.8±4.9	67.2±2.7	64.7±7.4	60.9±6.2	72.4±8.0	68.6±5.3	73.1±11.0	**77.9±7.3**
	NMI	78.7±2.5	79.3±2.2	78.8±1.0	81.6±1.3	80.7±0.8	72.9±5.1	78.5±2.3	64.7±11.6	**82.0±2.3**
	Purity	68.5±3.1	75.5±2.9	74.5±2.0	72.2±5.2	69.6±4.6	74.7±5.4	74.5±3.3	75.6±9.7	**80.3±4.6**
	ARI	61.1±1.7	64.0±3.1	63.3±1.1	64.5±2.9	62.9±2.0	59.1±7.0	63.1±2.6	55.2±12.2	**69.0±4.0**
	Time	0.27	0.28	0.18	0.72	0.71	0.47	0.54	0.36	**0.16**
Segment	ACC	64.1±2.9	60.4±6.2	65.4±3.5	60.9±3.8	63.1±4.5	64.1±7.4	61.1±5.4	60.0±6.6	**65.7±3.1**
	NMI	60.9±1.3	62.7±6.2	64.8±1.0	72.0±1.8	**72.5±2.6**	60.8±6.0	62.6±2.1	48.4±6.4	66.5±2.6
	Purity	64.9±2.1	62.1±5.3	66.4±2.3	69.5±1.4	**70.2±2.6**	66.4±6.4	63.8±3.5	63.5±5.3	67.9±2.6
	ARI	48.9±2.6	46.4±8.4	52.9±2.4	56.9±2.4	**57.4±3.5**	45.4±7.3	48.1±5.2	40.9±6.5	54.2±3.3
	Time	1.27	0.62	0.61	5.73	6.05	10.75	1.04	5.16	**0.44**
MnistData_05	ACC	60.9±3.5	57.8±2.4	58.4±1.9	69.0±3.0	68.2±2.8	51.4±4.6	57.7±3.3	63.3±3.0	**69.3±3.6**
	NMI	60.5±2.1	61.3±1.1	58.8±1.5	**72.3±1.1**	72.1±0.9	55.4±2.8	60.4±1.0	58.2±2.6	69.5±1.8
	Purity	65.2±2.0	65.4±1.5	64.9±1.5	**73.4±2.0**	73.3±1.2	59.3±3.3	64.3±2.8	66.8±3.3	69.5±3.1
	ARI	47.9±3.4	43.5±1.7	45.2±1.9	**59.9±2.4**	59.8±2.3	37.4±5.0	43.4±2.6	46.5±3.3	56.6±2.2
	Time	41.96	14.22	14.29	25.03	25.24	40.57	16.84	64.31	**11.63**
Waveform	ACC	64.3±10.1	53.3±1.1	51.7±1.7	50.4±0.4	51.0±0.8	54.8±5.3	53.5±3.2	50.8±1.5	**65.6±6.8**
	NMI	41.5±3.9	37.0±0.1	36.9±0.2	36.7±0.2	36.6±0.3	32.0±6.2	37.8±1.0	33.2±1.9	**42.2±2.5**
	Purity	65.1±9.4	53.5±0.8	52.0±1.6	51.7±0.4	51.9±0.7	56.1±4.4	55.4±2.8	51.9±1.1	**66.2±5.9**
	ARI	**34.8±8.1**	25.5±0.2	25.2±0.4	25.1±0.1	25.1±0.2	24.1±5.5	26.9±1.8	23.5±1.0	34.7±4.6
	Time	5.85	2.17	1.74	69.50	72.87	57.58	2.76	58.68	**1.13**
Texture	ACC	66.4±4.0	66.3±4.2	67.6±3.7	63.2±5.2	63.1±5.1	65.9±4.7	66.4±3.2	66.7±5.4	**74.6±3.1**
	NMI	69.9±2.0	72.5±2.2	71.7±1.7	**77.3±2.0**	75.1±2.5	71.0±2.9	70.2±1.4	66.3±5.9	76.0±1.6
	Purity	68.7±2.7	69.1±3.0	70.3±2.4	71.1±3.9	70.7±3.7	70.4±3.8	68.7±2.0	69.5±4.3	**75.9±2.7**
	ARI	54.1±4.4	54.9±4.4	57.2±3.4	51.6±5.4	51.0±4.9	52.1±5.1	54.7±3.7	53.1±6.7	**64.4±3.0**
	Time	7.07	2.56	2.36	26.93	30.29	77.27	2.88	37.33	**1.88**
Odr	ACC	84.7±2.9	83.0±3.0	87.3±2.3	84.8±3.6	87.6±3.6	82.5±6.5	83.7±4.0	82.9±5.9	**92.2±1.9**
	NMI	83.4±1.3	83.2±1.0	84.1±1.5	87.0±2.1	**88.4±1.4**	84.5±3.8	81.8±1.4	70.7±5.2	87.7±1.4
	Purity	85.5±2.5	84.6±2.3	87.4±2.2	86.3±3.4	89.3±2.4	85.1±5.6	84.8±2.4	84.9±4.6	**92.2±1.9**
	ARI	76.9±2.8	75.8±3.2	79.4±2.6	79.4±4.0	82.7±3.0	72.5±6.5	74.7±3.9	64.7±6.0	**84.7±2.6**
	Time	9.02	3.08	3.02	44.69	47.10	81.58	4.00	39.45	**2.13**
Isolet	ACC	57.2±2.0	52.0±2.8	56.3±1.9	56.2±2.0	54.8±2.0	55.9±2.5	58.8±2.8	57.1±3.8	**61.8±3.0**
	NMI	75.6±0.9	73.6±1.2	75.8±0.6	78.0±0.7	77.0±1.3	72.4±1.1	74.6±0.7	68.4±3.4	**83.0±1.1**
	Purity	61.6±1.5	58.0±2.1	61.7±1.1	62.0±1.4	60.7±2.0	58.8±1.9	61.8±1.9	60.6±3.7	**63.6±2.9**
	ARI	51.8±2.0	44.2±4.2	50.3±2.4	50.1±2.1	49.0±2.0	48.5±2.4	51.7±1.7	47.4±3.5	**62.7±2.1**
	Time	110.18	35.75	36.18	94.14	101.60	215.49	45.43	464.08	**25.10**
Pendigits	ACC	75.0±4.4	79.9±4.6	76.8±4.2	78.1±5.3	77.9±6.3	69.9±7.5	73.8±4.5	70.2±7.6	**84.3±4.0**
	NMI	75.3±2.1	80.0±1.8	78.7±1.9	81.0±2.1	80.0±2.6	74.0±4.8	73.7±1.1	59.6±6.7	**82.3±1.9**
	Purity	76.9±3.1	80.9±3.5	79.5±3.0	80.6±3.7	80.4±4.5	73.3±6.3	76.0±2.8	71.8±5.9	**84.6±3.6**
	ARI	62.0±3.7	67.0±5.0	65.8±4.3	67.5±5.1	67.3±6.1	56.7±7.5	60.0±3.0	50.8±7.1	**73.5±3.9**
	Time	17.61	5.08	3.84	130.75	149.79	302.17	4.71	519.25	**2.57**

matrix, and the structural regularizer keeps the solution away from the trivial ones. On Segment and MnistData_05, MDEC-BG and MDEC-SC are better on NMI, Purity and ARI, since the mapping matrix cannot split a base cluster that straddles several true classes, while these two methods keep the structure at the sample level. MAWEC still gives the highest ACC on both.

In terms of efficiency, MAWEC is the fastest on every dataset, and its margin over MDEC-BG, MDEC-SC, SCAC and CDEC tends to widen as the sample size grows, because its quadratic forms involve only small matrices, whereas these four methods decompose or iterate on structures that grow with n.

4.2 Ablation Study

To verify the contribution of each mechanism, we design five ablation variants, each of which changes one component of (9) and is solved by the same alternating optimization. Expanding the centering matrix in (6) gives

$$\mathrm{Tr}\big(Y^\top HY\big) = \|Y\|_F^2 - \frac{1}{n}\big\|\mathbf{1}^\top Y\big\|_2^2, \tag{22}$$

where maximizing the first term drives row sparsity and subtracting the second drives column balance, so the two components of the regularizer can be ablated separately. The five variants are

Table 2: Ablation results and results under different base clustering generation strategies on the Pendigits dataset (mean ± std %). KMeans and NCut denote a single run of k-means and of normalized cut. Within each block, the best result in each column is in bold and the second best is underlined.

Method	ACC	NMI	Purity	ARI
Ablation variants				
MAWEC-EW	81.52±05.69	78.53±02.57	82.23±04.98	69.85±05.59
MAWEC-w/oR	77.65±08.01	76.26±03.60	79.22±06.85	63.21±08.36
MAWEC-Dis	80.51±08.50	79.15±04.20	83.28±06.51	68.83±09.27
MAWEC-Bal	79.36±02.82	78.58±01.52	79.62±02.67	66.45±02.94
MAWEC-w/oB	76.52±06.09	75.33±02.97	79.74±05.27	61.72±06.14
MAWEC	**84.28±04.02**	**82.28±01.89**	**84.56±03.61**	**73.53±03.85**
Base clustering generation				
KMeans	68.82±05.01	68.01±00.95	71.65±03.03	54.69±03.17
NCut	74.14±03.29	73.46±01.23	74.51±02.97	64.05±03.40
MAWEC-KMeans	84.28±04.02	82.28±01.89	84.56±03.61	73.53±03.85
MAWEC-NCut	85.87±03.49	84.94±01.42	86.47±02.70	76.01±03.26
MAWEC-Mixed	**87.31±02.89**	**85.68±01.51**	**87.53±02.55**	**78.17±02.58**

as follows. MAWEC-EW fixes $\alpha = \frac{1}{T}\mathbf{1}$ and removes the weighting at the base clustering level. MAWEC-w/oR keeps the adaptive weights but sets $\gamma = 0$. MAWEC-Dis keeps only the discretization component, so the memberships concentrate but the cluster sizes are unconstrained. MAWEC-Bal keeps only the balance component, so the clusters are of similar size but the memberships are not pushed to concentrate. MAWEC-w/oB sets $\lambda = 0$ and drops the structural regularizer as a whole. Their objectives are given in Appendix G, and the results on Pendigits are in the upper block of Table 2.

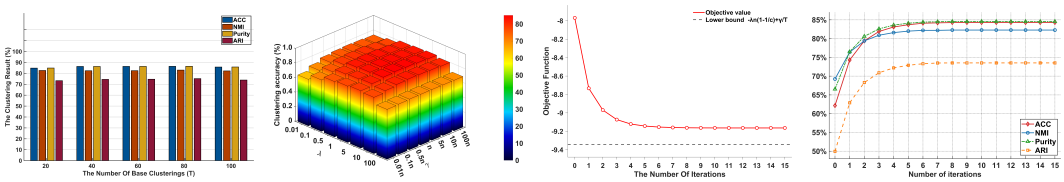
Three conclusions follow. First, the full MAWEC outperforms all the variants on the four metrics, so the adaptive weighting and the two regularizers are all necessary, and the variant without the structural regularizer degrades most on ACC, NMI and ARI, because Y is then driven by the fidelity term alone and slides toward trivial solutions. Second, MAWEC-Dis and MAWEC-Bal beat MAWEC-w/oB on most metrics yet stay below the full model; the former fluctuates more, and the latter is the most stable but less accurate, so the discretization component improves the discriminability of the solution and the balance component improves its stability. Third, the variant without the weight regularizer is even worse than the equally weighted variant, so adaptive weights only work together with the weight regularizer, which agrees with Section 3.3.

4.3 Sensitivity to the Base Clustering Generation

MAWEC generates its base clusterings by k-means, which favors convex clusters, so we check whether the result depends too heavily on this generator. Keeping the other settings unchanged, we compare three strategies: MAWEC-KMeans is the original one, MAWEC-NCut replaces the generator by normalized cut (Shi & Malik, 2000), and MAWEC-Mixed produces half of the base clusterings by each. Two baselines, KMeans and NCut, run the corresponding algorithm once on the original data. The results are in the lower block of Table 2. The three MAWEC variants clearly outperform the two baselines on all four metrics, so the improvement comes from fusing multiple base clusterings rather than from any single generator. A single run of normalized cut is much stronger than a single run of k-means, yet the gap between MAWEC-NCut and MAWEC-KMeans is much smaller, which shows that the two-granularity adaptive weighting suppresses unreliable base clusters and weakens the dependence on the quality of the generator. In addition, MAWEC-Mixed is the best on all four metrics, because the two generators make errors at different places and mixing them brings extra diversity.

4.4 Parameter Sensitivity

MAWEC involves the number of base clusterings T and the two regularization coefficients λ and γ. Figure 2(a) reports the four metrics as T grows from 20 to 100, and Figure 2(b) reports the clustering



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

Figure 2: Parameter sensitivity and convergence of MAWEC on the Pendigits dataset. (a) The four metrics for $T \in \{20, 40, 60, 80, 100\}$. (b) Clustering accuracy over the grid $\lambda \in \{0.01, 0.1, 0.5, 1, 5, 10, 100\}$ and $\gamma \in \{0.01n, 0.1n, 0.5n, n, 5n, 10n, 100n\}$. (c) Objective value against the number of iterations, where the dashed line is the lower bound of Theorem 2. (d) The four metrics against the number of iterations. All results are averaged over 20 independent runs.

accuracy over a grid of λ and γ, where γ is scaled by n so that it stays on the same order as the fidelity term. In Figure 2(a), all four metrics vary only slightly with T. They rise a little as T grows from 20 to 40, then level off, and even drop slightly at the largest T, where redundant partitions are introduced, so a value between 40 and 80 balances performance and cost. In Figure 2(b), the accuracy forms a wide plateau over the intermediate values and drops only at the two ends of the grid, where the fidelity term or a regularizer dominates. We therefore set $T = 50$, $\lambda = 1$ and $\gamma = n$ in all the other experiments.

4.5 Convergence Verification

Section 3.6 proves that the objective of MAWEC is monotonically non-increasing and bounded below. In Figure 2(c), the objective decreases monotonically, stops changing after about eight iterations, and stays above the theoretical lower bound marked by the dashed line, which agrees with Theorem 2. In Figure 2(d), the four metrics rise monotonically and stabilize within the same number of iterations, so the decrease of the objective translates into a real gain in clustering quality. MAWEC needs only a few iterations to converge, which confirms that the stopping rule in Section 3.4 is reasonable.

5 Conclusion

This paper addresses four problems in ensemble clustering: weights that are fixed and act on a single granularity, the information loss caused by two-stage processing, the degeneracy toward trivial solutions, and the limited scalability of the co-association matrix. We propose MAWEC, which replaces the co-association matrix by a base-cluster mapping matrix, learns the weights, the mapping and the labels within one objective, excludes three trivial solutions with two regularizers, outputs the result directly from the alternating optimization, and scales linearly with the number of samples. Experiments on eight real-world datasets show that MAWEC attains the highest ACC on all of them and the shortest running time, and the further studies confirm the necessity of each module and the stability of the method. Some limitations remain: the mapping matrix cannot separate the samples of a base cluster that straddles several true classes, and the two coefficients and the target number of clusters still have to be given in advance. Future work includes designing a parameter-free objective, introducing a sample-level correction, and extending the framework to multi-view clustering.

AI Use Statement

Among the research subtasks for which the ICLR AI policy requires disclosure, we used generative AI tools only to assist with translation, that is, to turn author-written drafts into English. We did not use generative AI tools to develop the theoretical model or conceptual framework, to formulate the mathematical claims, to provide critical ingredients for proving them or to write the proofs, to propose or refine the hypotheses, to design or give feedback on the methodology and the experiments, to implement the method, to clean or reformat the datasets, or to interpret the results; the remaining subtasks with required disclosure, namely generating synthetic datasets and supporting qualitative and thematic data analysis, are not applicable to this work. Among the subtasks with

recommended disclosure, we additionally used generative AI tools to edit the paper for readability. We have reviewed all AI-assisted work: every AI-assisted passage was checked line by line by the authors against the original draft and against the corresponding derivations, tables and figures, and was rewritten where the meaning had shifted. We take responsibility for the final content of this work, including its text, claims and artifacts.

Ethics Statement

This work proposes a general-purpose unsupervised clustering algorithm and evaluates it only on publicly available benchmark datasets that contain no personally identifiable or sensitive information. No human subjects were involved and no new data were collected. We are aware of no direct ethical concern raised by this work; as with any clustering method, care is needed when it is applied to data about people, since a partition learned without supervision may reproduce biases present in the data.

Reproducibility Statement

The generation of the base clusterings is specified in Section 3.1, the model in Section 3.2, and the full optimization procedure in Section 3.4, with the pseudocode in Algorithm 1 of Appendix C and the acceleration strategies of the coordinate descent in Appendix B. Complete proofs of Theorems 1 and 2, including all assumptions, are given in Appendices A and E. The datasets, the evaluation metrics, the hardware and the choice of λ, γ and T are reported in Sections 4.1 and 4.4, the remaining implementation details in Appendix F, and the objectives of all ablation variants in Appendix G.

References

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*, 2017.

Najla Altwaijry, Kais Feki, and Nicuşor Minculete. On some generalizations of Cauchy–Schwarz inequalities and their applications. *Symmetry*, 15(2):304, 2023.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1027–1035, 2007.

Arthur Asuncion and David Newman. UCI machine learning repository. School of Information and Computer Sciences, University of California, Irvine, 2007.

Liang Bai, Jiye Liang, Hangyuan Du, and Yike Guo. An information-theoretical framework for cluster ensemble. *IEEE Transactions on Knowledge and Data Engineering*, 31(8):1464–1477, 2019.

Liang Bai, Jiye Liang, and Fuyuan Cao. A multiple k-means clustering ensemble algorithm to find nonlinearly separable clusters. *Information Fusion*, 61:36–47, 2020.

James C. Bezdek and Richard J. Hathaway. Some notes on alternating optimization. In *AFSS International Conference on Fuzzy Systems*, pp. 288–300. Springer, 2002.

Tossapon Boongoen and Natthakan Iam-On. Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science Review*, 28:1–25, 2018.

Deng Cai, Xiaofei He, and Jiawei Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, 2005.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pp. 272–279, 2008.

Xiaoli Zhang Fern and Carla E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pp. 186–193, 2003.

Xiaoli Zhang Fern and Carla E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pp. 36, 2004.

Ana L. N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005.

Keyvan Golalipour, Ebrahim Akbari, Seyed Saeed Hamidi, Malrey Lee, and Rasul Enayatifar. From clustering to clustering ensemble selection: A review. *Engineering Applications of Artificial Intelligence*, 104:104388, 2021.

Derek Greene, Alexey Tsymbal, Nadia Bolshakova, and Pádraig Cunningham. Ensemble clustering in medical diagnostics. In *Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems*, pp. 576–581. IEEE, 2004.

Zhezheng Hao, Zhoumin Lu, Guoxu Li, Feiping Nie, Rong Wang, and Xuelong Li. Ensemble clustering with attentional representation. *IEEE Transactions on Knowledge and Data Engineering*, 36(2):581–593, 2024.

Dong Huang, Jian-Huang Lai, and Chang-Dong Wang. Robust ensemble clustering using probability trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 28(5):1312–1326, 2016.

Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. LWMC: A locally weighted meta-clustering algorithm for ensemble clustering. In *International Conference on Neural Information Processing (ICONIP)*, pp. 167–176. Springer, 2017.

Dong Huang, Chang-Dong Wang, and Jian-Huang Lai. Locally weighted ensemble clustering. *IEEE Transactions on Cybernetics*, 48(5):1460–1473, 2018.

Dong Huang, Chang-Dong Wang, Jian-Sheng Wu, Jian-Huang Lai, and Chee-Keong Kwoh. Ultra-scalable spectral clustering and ensemble clustering. *IEEE Transactions on Knowledge and Data Engineering*, 32(6):1212–1226, 2019.

Dong Huang, Chang-Dong Wang, Hongxing Peng, Jianhuang Lai, and Chee-Keong Kwoh. Enhanced ensemble clustering via fast propagation of cluster-wise similarities. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(1):508–520, 2021.

Dong Huang, Chang-Dong Wang, Jian-Huang Lai, and Chee-Keong Kwoh. Toward multidiversified ensemble clustering of high-dimensional data: From subspaces to metrics and beyond. *IEEE Transactions on Cybernetics*, 52(11):12231–12244, 2022.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

Yuheng Jia, Sirui Tao, Ran Wang, and Yongheng Wang. Ensemble clustering via co-association matrix self-enhancement. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8): 11168–11179, 2024.

Feijiang Li, Jieting Wang, Liuya Zhang, Yuhua Qian, Shuai Jin, Tao Yan, and Liang Du. k-HyperEdge medoids for clustering ensemble. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18271–18278, 2025.

Zhenguo Li, Xiao-Ming Wu, and Shih-Fu Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 789–796, 2012.

Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779, 2007.

Stuart Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2): 129–137, 1982.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3523–3542, 2022.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pp. 849–856, 2001.

Gbeminiyi John Oyewole and George Alex Thopil. Data clustering: application and trends. *Artificial Intelligence Review*, 56(7):6439–6475, 2023.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

Ying Sun, Prabhu Babu, and Daniel P. Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65 (3):794–816, 2016.

Alexander Topchy, Anil K. Jain, and William Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12): 1866–1881, 2005.

Xiaoqian Wang, Feiping Nie, and Heng Huang. Structured doubly stochastic matrix for graph based clustering. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1245–1254, 2016.

Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

Fangyuan Xie, Feiping Nie, Weizhong Yu, and Xuelong Li. Parameter-free ensemble clustering with dynamic weighting mechanism. *Pattern Recognition*, 151:110389, 2024.

Jing Xu, Xiaoyu Wang, Hao Zhang, and Panpan Lv. Heterogeneous and clustering-enhanced personalized preference transfer for cross-domain recommendation. *Information Fusion*, 99:101892, 2023.

Lijun Xu and Shifei Ding. Dual-granularity weighted ensemble clustering. *Knowledge-Based Systems*, 225:107124, 2021.

Xiaojun Yang, Weihao Zhao, Jing Wang, Siyuan Peng, and Feiping Nie. Auto-weighted graph reconstruction for efficient ensemble clustering. *Information Sciences*, 689:121486, 2025a.

Xiaojun Yang, Zhenwei Zheng, Jiahui Xie, Weihao Zhao, Jingjing Xue, and Feiping Nie. Spectral ensemble clustering from graph reconstruction with auto-weighted cluster. *Pattern Recognition Letters*, 196:243–249, 2025b.

Rui Zhang, Shuo Hang, Zhenyu Sun, Feiping Nie, Rong Wang, and Xuelong Li. Anchor-based fast spectral ensemble clustering. *Information Fusion*, 113:102587, 2025a.

Rui Zhang, Xuelong Wu, Hongyuan Chen, Guanghui He, Zhen Wang, Rong Wang, and Feiping Nie. Toward balance adaptive weighted ensemble clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(6):5279–5292, 2025b.

Yuxia Zhang, Jinsong Liu, and Wei Shen. A review of ensemble learning algorithms used in remote sensing applications. *Applied Sciences*, 12(17):8654, 2022.

Zhi-Hua Zhou and Wei Tang. Clusterer ensemble. *Knowledge-Based Systems*, 19(1):77–83, 2006.

Weiwei Zhuang, Yanfang Ye, Yong Chen, and Tao Li. Ensemble clustering for internet security applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1784–1796, 2012.

Quan Zou, Gang Lin, Xingpeng Jiang, Xiangrong Liu, and Xiangxiang Zeng. Sequence clustering in bioinformatics: an empirical study. *Briefings in Bioinformatics*, 21(1):1–10, 2020.

A Proof of Theorem 1

Proof. Let $y_j \in \mathbb{R}^{n\times 1}$ be the j-th column of Y, i.e. the membership vector of the j-th cluster over all samples, and let $n_j = \mathbf{1}^\top y_j$ be the size of the j-th cluster. From $Y\mathbf{1} = \mathbf{1}$ we get $\sum_{j=1}^{c} n_j = \mathbf{1}^\top Y \mathbf{1} = n$. Expanding H in (6) gives

$$\operatorname{Tr}\big(Y^\top H Y\big) = \sum_{j=1}^{c} \Big[\|y_j\|_2^2 - \tfrac{1}{n}\big(\mathbf{1}^\top y_j\big)^2\Big] = \sum_{j=1}^{c} \|y_j\|_2^2 - \frac{1}{n}\sum_{j=1}^{c} n_j^2. \tag{23}$$

From $0 \le Y_{ij} \le 1$ we have $Y_{ij}^2 \le Y_{ij}$, hence

$$\sum_{j=1}^{c} \|y_j\|_2^2 = \sum_{i,j} Y_{ij}^2 \le \sum_{i,j} Y_{ij} = n, \tag{24}$$

with equality if and only if all entries of Y are 0 or 1, that is, the clustering is hard. On the other hand, the Cauchy–Schwarz inequality (Altwaijry et al., 2023) $\big(\sum_j n_j\big)^2 \le c\sum_j n_j^2$ together with $\sum_j n_j = n$ yields

$$\sum_{j=1}^{c} n_j^2 \ge \frac{n^2}{c}, \tag{25}$$

with equality if and only if all n_j are equal, i.e. $n_j = n/c$ and the clusters are balanced. Substituting (24) and (25) into (23), we obtain

$$\operatorname{Tr}\big(Y^\top H Y\big) \le n - \frac{1}{n}\cdot\frac{n^2}{c} = \frac{n(c-1)}{c}, \tag{26}$$

and the two equality conditions must hold simultaneously, which requires c to divide n. □

B Acceleration Strategies for the Coordinate Descent

When row g of Z is updated, it has c candidate vectors $[1,0,\ldots,0],\ldots,[0,0,\ldots,1]$, which correspond to c candidate matrices $Z^{(1)},\ldots,Z^{(c)}$ that differ only in this row. Computing the objective value of each candidate one by one would be expensive, so we use the following three strategies.

Acceleration Strategy 1: Incremental Computation. We introduce an auxiliary matrix $Z^{(0)} \in \mathbb{R}^{m\times c}$ whose row g is all zero and which otherwise agrees with every $Z^{(l)}$. Let z_l be the l-th column of Z, q_g the g-th column of Q, q_{gg} the (g,g) entry of Q, R_{gl} the (g,l) entry of R, and e_g the standard basis vector whose g-th entry is 1 and all the others are 0. Each $Z^{(l)}$ sets the l-th position of row g from 0 to 1, i.e. $z_l^{(l)} = z_l^{(0)} + e_g$, while the other columns are the same as in $Z^{(0)}$. Expanding gives

$$F\big(Z^{(l)}\big) - F\big(Z^{(0)}\big) = \big(z_l^{(0)} + e_g\big)^\top Q\big(z_l^{(0)} + e_g\big) - \big(z_l^{(0)}\big)^\top Q z_l^{(0)} - 2R_{gl} = 2q_g^\top z_l^{(0)} + q_{gg} - 2R_{gl}. \tag{27}$$

Since both $F(Z^{(0)})$ and q_{gg} are independent of l, solving (17) for row g reduces to

$$\beta = \operatorname*{arg\,min}_{l\in\{1,\ldots,c\}} \phi(l), \qquad \phi(l) = q_g^\top z_l^{(0)} - R_{gl}, \tag{28}$$

where β is the newly selected position. To avoid storing the $c+1$ candidate matrices, we work on the current Z. Let $p \in \{1,\ldots,c\}$ be the position of the entry 1 in row g of Z. When $l = p$ we have $z_p^{(0)} = z_p - e_g$, hence

$$\phi(p) = q_g^\top z_p - q_{gg} - R_{gp}, \tag{29}$$

and when $l \neq p$ we have $z_l^{(0)} = z_l$, hence

$$\phi(l) = q_g^\top z_l - R_{gl}. \tag{30}$$

Row g of Z is then updated as

$$Z_{gl} = \begin{cases} 1, & l = \beta, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

If p is among the minimizers, we keep $\beta = p$ and leave the current assignment unchanged, so as to avoid pointless oscillation. Comparing the objective values of c candidate matrices is thus replaced by comparing c scalars.

Acceleration Strategy 2: Storage Optimization. We precompute and store the matrix $U = QZ \in \mathbb{R}^{m\times c}$, whose (i, l) entry is $U_{il} = q_i^\top z_l$ with q_i the i-th column of Q, together with the diagonal entries of Q and the matrix R. Equations (29) and (30) can then be written uniformly as

$$\phi(l) = \begin{cases} U_{gl} - R_{gl} - q_{gg}, & l = p, \\ U_{gl} - R_{gl}, & l \neq p. \end{cases} \tag{32}$$

Evaluating all c candidates then requires only $O(c)$ additions and subtractions, with no matrix–vector multiplication.

Acceleration Strategy 3: Update Optimization. After row g is updated, the cached matrix U must be refreshed. When $\beta = p$, row g is unchanged and all cached quantities stay the same. When $\beta \neq p$, only the p-th and the β-th entries of row g change, i.e. $z_p \leftarrow z_p - e_g$ and $z_\beta \leftarrow z_\beta + e_g$, so correspondingly

$$U_{ip} \leftarrow U_{ip} - Q_{ig}, \qquad U_{i\beta} \leftarrow U_{i\beta} + Q_{ig}, \qquad \forall\, i = 1, \ldots, m, \tag{33}$$

where Q_{ig} is the (i, g) entry of Q. The matrix R and the diagonal of Q do not depend on Z and need no update.

Selecting the candidate by (32) takes $O(c)$ operations and updating the cache by (33) takes $O(m)$ operations, so a single row update costs $O(m + c)$ and one epoch that sweeps all m rows costs $O\big(m(m+c)\big)$. Since each row update reassigns one base cluster, the three strategies reduce the cost of solving for the fine-grained weights at the base cluster level to $O(m + c)$.

C Algorithm Procedure

Algorithm 1 Mapping-based Adaptively Weighted Ensemble Clustering (MAWEC)

Require: Data matrix X, number of clusters c, number of base clusterings T, coefficients λ and γ, maximum number of iterations $t_{\max}$, threshold ε

Ensure: Class label ℓ_i of each sample

1: Generate T base clusterings as in Section 3.1 to obtain $\{Y_k\}_{k=1}^T$, and let $m = \sum_k m_k$
2: Initialize $\alpha^0 = \frac{1}{T}\mathbf{1}$, obtain Z^0 by k-means on the base cluster centers, and set $Y^0 = A^0 Z^0$
3: **for** $t = 0, 1, \ldots, t_{\max} - 1$ **do**
4: Form A^t by (8), compute $A^t Z^t + \lambda H Y^t$, and project each of its rows onto the simplex by (16) to obtain Y^{t+1}
5: Compute $Q = (A^t)^\top A^t$ and $R = (A^t)^\top Y^{t+1}$, cache $U = QZ^t$ and the diagonal of Q, and run CD from Z^t until convergence to obtain Z^{t+1}
6: Compute G and h by (18), and solve (19) by (20) to obtain α^{t+1}
7: **break** if $\left|\mathcal{L}^{t+1} - \mathcal{L}^t\right| < \varepsilon$
8: **end for**
9: Obtain the class label ℓ_i of each sample by (7)

D Breakdown of the Per-iteration Cost

For updating Y, forming $A^t Z^t$ by row lookup costs $O(Tnc)$, the centering costs $O(nc)$, and the row-wise simplex projection in (16) costs $O(nc \log c)$. For updating Z, computing Q costs $O(nT^2)$, since each row of A^t has only T nonzero entries, computing R costs $O(Tnc)$, and each CD epoch costs $O\big(m(m+c)\big)$. For updating α, computing G and h by (18) costs $O(T^2 n)$ and $O(Tn)$ respectively, and each projected gradient step in (20) costs $O(T^2)$. Summing these terms gives (21).

E Proof of Theorem 2

Proof. We first prove the monotonicity. One outer iteration updates Y, Z and α in turn, and the three steps are examined separately.

When Y is updated, Z^t and α^t are fixed. By the MM construction, $g(\cdot, Y^t)$ in (14) is a surrogate upper bound of $\mathcal{L}(\cdot, Z^t, \alpha^t)$ that touches it at Y^t, where the terms independent of Y are absorbed into the constant of g. By (15), Y^{t+1} is the minimizer of $g(\cdot, Y^t)$ over the feasible set. Hence

$$\mathcal{L}(Y^{t+1}, Z^t, \alpha^t) \le g(Y^{t+1}, Y^t) \le g(Y^t, Y^t) = \mathcal{L}(Y^t, Z^t, \alpha^t), \tag{34}$$

where the first inequality follows from the upper-bound condition, the second from the optimality of Y^{t+1}, and the last equality from the touching condition.

When Z is updated, Y^{t+1} and α^t are fixed. CD starts from Z^t and takes the minimum among the c candidates in every row. The current row is itself one of the candidates, so no row update increases the objective, and therefore

$$\mathcal{L}(Y^{t+1}, Z^{t+1}, \alpha^t) \le \mathcal{L}(Y^{t+1}, Z^t, \alpha^t). \tag{35}$$

When α is updated, Y^{t+1} and Z^{t+1} are fixed. Subproblem (19) is a strictly convex quadratic program on the simplex, and PG in (20) starts from the feasible α^t with the step size $1/L$, so no PG step increases its objective. Hence

$$\mathcal{L}(Y^{t+1}, Z^{t+1}, \alpha^{t+1}) \le \mathcal{L}(Y^{t+1}, Z^{t+1}, \alpha^t). \tag{36}$$

Chaining (34), (35) and (36) gives

$$\mathcal{L}^{t+1} \le \mathcal{L}^t, \tag{37}$$

so the objective is monotonically non-increasing.

We next prove the lower bound. In (9), the first term satisfies $\|Y - AZ\|_F^2 \ge 0$. The row-wise simplex projection (16) keeps $Y\mathbf{1} = \mathbf{1}$ and $Y \ge 0$ at every iteration, so Theorem 1 applies and (10) gives $\mathrm{Tr}(Y^\top HY) \le n(c-1)/c$, which bounds the second term. The update (20) keeps α on the simplex (4), so (12) gives $\|\alpha\|_2^2 \ge 1/T$, which bounds the third term. Combining the three terms yields

$$\mathcal{L}(Y, Z, \alpha) \ge -\lambda \frac{n(c-1)}{c} + \frac{\gamma}{T} \tag{38}$$

for every feasible (Y, Z, α). A monotonically non-increasing sequence that is bounded below converges, which completes the proof. □

F Experimental Settings

Table 3 lists the eight datasets. In all the experiments we set the outer stopping threshold to $\varepsilon = 10^{-6}$ and the maximum number of outer iterations to $t_{\max} = 100$, the inner CD loop stops when $F(Z)$ changes by less than 10^{-6}, and the PG loop performs at most 200 steps.

Table 3: Statistics of the eight datasets.

Dataset	**#Samples**	**#Features**	**#Classes**
Control	600	60	6
Segment	2310	18	7
MnistData_05	3495	784	10
Waveform	5000	21	3
Texture	5500	100	11
Odr	5620	64	10
Isolet	7797	617	26
Pendigits	10992	16	10

G Objectives of the Ablation Variants

All the objectives below inherit the constraints of (9).

MAWEC-EW. The weights are fixed at $\alpha = \frac{1}{T}\mathbf{1}$, so the weight regularizer becomes a constant and A reduces to the equally weighted association matrix $A_0 = \frac{1}{T}[Y_1, \ldots, Y_T]$:

$$\min \ \big\|Y - A_0 Z\big\|_F^2 - \lambda \operatorname{Tr}\big(Y^\top H Y\big). \tag{39}$$

The update of α is skipped, and only Y and Z are updated alternately.

MAWEC-w/oR. The adaptive weights are kept and only γ is set to zero:

$$\min \ \big\|Y - AZ\big\|_F^2 - \lambda \operatorname{Tr}\big(Y^\top H Y\big). \tag{40}$$

The coefficient matrix of the subproblem in α is then the Gram matrix G alone, which is only positive semidefinite, so the optimum need not be unique and can collapse onto a few base clusterings.

MAWEC-Dis. Following (22), only the discretization component is kept:

$$\min \ \big\|Y - AZ\big\|_F^2 - \lambda \|Y\|_F^2 + \gamma \|\alpha\|_2^2. \tag{41}$$

This variant isolates the effect of removing column balance.

MAWEC-Bal. In the opposite direction, only the balance component is kept:

$$\min \ \big\|Y - AZ\big\|_F^2 + \frac{\lambda}{n}\big\|\mathbf{1}^\top Y\big\|_2^2 + \gamma \|\alpha\|_2^2. \tag{42}$$

This variant isolates the effect of removing row sparsity.

MAWEC-w/oB. Setting $\lambda = 0$ removes the structural regularizer as a whole:

$$\min \ \big\|Y - AZ\big\|_F^2 + \gamma \|\alpha\|_2^2, \tag{43}$$

so that Y is constrained by the fidelity term only.