SOM-TSK: A Self-Organizing Map Topology-Seeded Clustering Framework with Deterministic Multi-Start Initialization and Adaptive Selection

Evint Leovonzko

***Abstract*—We present *SOM-TSK* (Self-Organizing Map Topology-Seeded K-means), a clustering framework that exploits the manifold-mapping properties of a trained Self-Organizing Map to generate high-quality initialization seeds for downstream K-means refinement. Unlike heuristic SOM-to-cluster pipelines, SOM-TSK assembles a diverse candidate pool through four complementary, fully deterministic seeding phases: (A) KMeans++ on the neuron grid, (A-alt) maximum-spread neuron seeding, (B-bis) multi-start coverage-aware density seeding from the top-3 high-activation neurons, and (C) a fresh KMeans++ baseline on raw data. Candidates are selected by maximizing the Calinski–Harabasz criterion within a 5% inertia envelope, balancing cluster separation quality against convergence proximity to the global optimum. We additionally introduce *DenSOM*, a density-based algorithm that applies Gaussian smoothing, Otsu thresholding, and topographic watershed flood-fill to the SOM activation map to discover an automatic number of clusters with noise rejection; and *AutoSOM*, a meta-algorithm that uses GMM-BIC k-selection to route each dataset to either SOM-TSK or DenSOM based on structure. Evaluation on 24 benchmark datasets spanning the SIPU suite, synthetic shape datasets, UCI real-world collections, and scalability stress tests shows that SOM-TSK achieves 6 wins, 18 ties, and 0 losses against KMeans++ across all 24 datasets, with improvements of up to $+0.231$ ARI (handwritten digits), and matches or exceeds KMeans++ on every evaluation metric. The entire framework is implemented in safe Rust with optional CUDA and Metal backends and achieves competitive throughput on datasets up to 50,000 samples.**

***Index Terms*—Self-Organizing Map, K-means clustering, initialization, topology-seeded clustering, density-based clustering, Calinski–Harabasz criterion, benchmark evaluation**

I. Introduction

CLUSTERING remains one of the most widely studied problems in unsupervised machine learning. The K-means algorithm [1], [2] dominates practical applications owing to its simplicity and scalability, yet its quality is notoriously initialization-sensitive: random centroid placement routinely traps the algorithm in suboptimal local minima [3]. The KMeans++ initialization strategy [3] mitigates this by spreading seeds proportional to squared distance, achieving an $O(\log k)$ approximation guarantee in expectation. Nevertheless, a single KMeans++ draw can still produce a poor partition when clusters are highly overlapping, imbalanced, or when k is large relative to dataset size.

E. Leovonzko is with the Department of Computer Science and Engineering. E-mail: https://github.com/Evintkoo/SOM\protect_plus\protect_clustering

Self-Organizing Maps (SOMs) [4], [5] offer a complementary view of the data: by learning a low-dimensional neuron grid that approximates the data manifold, a trained SOM implicitly encodes global topology—neighbourhood relationships, density variation, and inter-cluster structure—in its weight matrix. This information is not exploited by standard KMeans initialization.

Several works have proposed coupling SOMs with centroid-based post-processing. Vesanto and Alhoniemi [6] cluster the SOM neurons with KMeans or hierarchical methods, then assign data points by best-matching unit (BMU) lookup. Taşdemir and Merenyi [7] exploit the SOM's data adjacency graph for topology-preserving visualization. However, these approaches are typically studied as exploratory tools rather than evaluated rigorously as standalone clustering algorithms in direct comparison with KMeans++ on standardized benchmarks.

Contributions. This paper makes the following contributions:

1) We formalize and systematically evaluate the *SOM-TSK* pipeline, establishing which seeding phases reliably improve over KMeans++ and which are harmful when selection criteria are not carefully constrained.
2) We introduce three new seeding strategies within SOM-TSK: the multi-start coverage-aware Phase B-bis, the maximum-spread Phase A-alt, and a unified inertia-windowed Calinski–Harabasz selection criterion. All are fully deterministic, eliminating sensitivity to random seed choices that affected prior comparisons.
3) We propose *DenSOM*, a parameter-light density-based algorithm operating entirely on the SOM activation map, capable of discovering the number of clusters automatically and rejecting noise.
4) We propose *AutoSOM*, a meta-algorithm that combines GMM-BIC k-selection with consensus routing between SOM-TSK and DenSOM.
5) We conduct a rigorous evaluation over 24 benchmark datasets from five categories, reporting ARI, NMI, FMI, Silhouette, Davies–Bouldin, and Calinski–Harabasz scores. Code: https://github.com/Evintkoo/SOM_plus_clustering.

II. Related Work

A. K-means Initialization

The sensitivity of K-means to initialization has motivated extensive research. Arthur and Vassilvitskii [3] proved that KMeans++ achieves an $O(\log k)$ competitive ratio in expectation. Bahmani et al. [21] proposed k-means‖, a scalable parallel initialization. Celebi et al. [22] conducted a comprehensive comparison of thirteen initialization methods, finding that careful deterministic strategies often outperform random ones. Katsavounidis et al. [24] proposed a max-distance greedy seeding similar to our SOM++ initialization, establishing that spread-maximizing seeds reduce the probability of degenerate initial configurations.

B. Multi-Start and Pool-Based Initialization

A complementary strategy to better seeding is running multiple independent initializations and selecting the best result. Bradley and Fayyad [25] showed that sub-sampling followed by multiple restarts on compressed data substantially reduces the chance of poor local minima. SOM-TSK generalizes this idea: rather than running multiple KMeans++ restarts independently, it constructs a *structured pool* of diverse candidates seeded from different topological perspectives of the SOM (Phases A, A-alt, B, B-bis, C), then selects by an internal criterion. This pool-based approach is more efficient than brute-force multi-restart because each phase uses complementary structural information that reduces redundancy among candidates.

C. SOM-Based Clustering

Kohonen [26] provided an updated account of the SOM's properties as a topology-preserving manifold approximator. Kohonen [5] originally proposed using the trained map as a vector quantizer and applying hierarchical clustering to its weight matrix. Vesanto and Alhoniemi [6] extended this by comparing KMeans with Ward linkage for the neuron-level post-processing step. Taşdemir and Merenyi [7] showed that the SOM's implicit data adjacency topology can reveal cluster boundaries invisible to standard metrics. Su and Chang [8] proposed using SOM neurons as density estimators to avoid k specification.

D. Density-Based Clustering on SOMs

Density-based methods such as DBSCAN [9] and HDBSCAN [10] operate in original data space and require careful tuning of neighbourhood parameters. Fritzke [28] and Marsland et al. [29] proposed self-organizing methods that grow/decay neurons based on local density but do not produce a readily usable flat partition. Our DenSOM algorithm operates directly on the SOM's activation map—a compressed, topology-preserving representation—applying Gaussian smoothing and Otsu's automatic thresholding [11] to identify core regions, followed by a topographic watershed flood-fill that avoids any distance-parameter specification.

E. Automatic k Selection

Selecting the number of clusters automatically is a longstanding challenge. The gap statistic [12], silhouette method [13], and elbow heuristic are widely used. BIC-based selection via Gaussian mixture models [14], [15] provides a principled maximum-likelihood framework. AutoSOM integrates BIC-guided GMM k-selection with density-mode counting from DenSOM.

III. Background

A. Self-Organizing Maps

A Self-Organizing Map places $M = m \times n$ prototype neurons $\{\mathbf{w}_j \in \mathbb{R}^d\}_{j=1}^{M}$ on a 2D rectangular grid. Training updates each neuron by the competitive learning rule:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t)\, h_{c,j}(t)\big[\mathbf{x}(t) - \mathbf{w}_j(t)\big], \qquad (1)$$

where $c = \arg\min_j \|\mathbf{x}(t) - \mathbf{w}_j\|$ is the Best-Matching Unit (BMU) for input $\mathbf{x}(t)$, $\alpha(t)$ is the learning rate, and $h_{c,j}(t) = \exp\!\big(-\|r_c - r_j\|^2/(2\sigma^2(t))\big)$ is the Gaussian neighbourhood kernel centered at r_c. Both α and σ decay linearly to small positive values over T epochs.

After training, each data point $\mathbf{x}_i$ is assigned to its BMU $b_i = \arg\min_j \|\mathbf{x}_i - \mathbf{w}_j\|$, producing the BMU index array $\mathbf{b} \in \{1, \ldots, M\}^n$.

B. KMeans and KMeans++

K-means minimizes the inertia $\mathcal{I} = \sum_{i=1}^{n} \|\mathbf{x}_i - \boldsymbol{\mu}_{c_i}\|^2$ by alternating between label assignment and centroid update. KMeans++ [3] seeds the k initial centroids by sampling $\mathbf{c}_1$ uniformly and then drawing each subsequent $\mathbf{c}_\ell$ with probability proportional to $D(\mathbf{x})^2$, the squared distance to the nearest already-chosen centre.

C. Calinski–Harabasz Criterion

The Calinski–Harabasz (CH) score [19] measures clustering quality as the ratio of between-cluster to within-cluster dispersion:

$$\mathrm{CH}(k) = \frac{\mathrm{tr}(\mathbf{B}_k)}{\mathrm{tr}(\mathbf{W}_k)} \cdot \frac{n-k}{k-1}, \qquad (2)$$

where $\mathbf{B}_k$ and $\mathbf{W}_k$ are the between- and within-cluster scatter matrices. Higher CH indicates tighter, better-separated clusters.

IV. SOM-TSK: Topology-Seeded K-means

A. Overview

SOM-TSK consists of two major phases: (i) SOM training to build a topology-preserving neuron grid, and (ii) multi-phase seeded K-means using the trained neurons to construct a diverse candidate pool, followed by selection. The full pipeline is summarized in Algorithm 1.

Algorithm 1 SOM-TSK: Topology-Seeded K-means

Require: Data $\mathbf{X} \in \mathbb{R}^{n \times d}$, cluster count k, SOM grid $m \times n_{\text{grid}}$, epochs T
Ensure: Cluster labels $\mathbf{y} \in \{0, \ldots, k-1\}^n$
1: Train SOM on $\mathbf{X}$ using (1) with SOM++ init
2: Compute BMU array $\mathbf{b}$; compute hit counts $h_j = |\{i : b_i = j\}|$
3: **Phase A:** Run KMeans++ on neurons $\{\mathbf{w}_j\}$; obtain partition $\boldsymbol{\pi}$
4: **Phase A-alt:** Seed k neurons via max-spread greedy sampling on $\{\mathbf{w}_j\}$; refine on $\mathbf{X}$; add to pool
5: **Phase B:** Map $\boldsymbol{\pi}$ to data-space centroids via weighted BMU means; refine on $\mathbf{X}$; add to pool
6: **Phase B-bis:** For each of top-3 hit-count neurons, run hit-weighted farthest spread to select k seeds; refine each on $\mathbf{X}$; add all to pool
7: **Phase C:** Run KMeans++ on $\mathbf{X}$ ($\times 20$ restarts if $d > 8$); add to pool
8: **Select:** $\mathcal{I}^* \leftarrow \min_{\text{pool}} \mathcal{I}$; retain $\mathcal{P}_{1.05} = \{\mathbf{y} : \mathcal{I}(\mathbf{y}) \le 1.05\,\mathcal{I}^*\}$; **return** $\arg\max_{\mathcal{P}_{1.05}} \text{CH}$

B. SOM Training with SOM++ Initialization

We initialize SOM neurons using a greedy farthest-point sampling strategy analogous to KMeans++ but applied to a small random subset of the data. The first neuron $\mathbf{w}_1$ is set to the data mean; each subsequent neuron $\mathbf{w}_j$ is placed at the data point maximally distant from all already-placed neurons:

$$\mathbf{w}_j = \mathbf{x}_{i^*}, \quad i^* = \arg\max_i \min_{\ell < j} \|\mathbf{x}_i - \mathbf{w}_\ell\|^2. \tag{3}$$

This initialization is entirely deterministic (no RNG calls), ensuring fully reproducible results. Combined with a deterministic epoch ordering (no shuffle), every benchmark result reported in this paper is exactly reproducible.

C. Phase A: Neuron-Grid KMeans

A KMeans++ instance is run on the M-neuron weight matrix $\mathbf{W} \in \mathbb{R}^{M \times d}$, producing a neuron-level partition $\boldsymbol{\pi} : \{1, \ldots, M\} \to \{0, \ldots, k-1\}$. Since $M \le 225$ in our experiments, this inner KMeans is extremely fast ($O(M)$ per iteration). Data-space centroids are computed as:

$$\boldsymbol{\mu}_c^{(A)} = \frac{\sum_{i:\boldsymbol{\pi}(b_i)=c} \mathbf{x}_i}{|\{i : \boldsymbol{\pi}(b_i) = c\}|}. \tag{4}$$

These centroids are then refined by running full KMeans from $\boldsymbol{\mu}^{(A)}$ to convergence.

D. Phase A-alt: Maximum-Spread Neuron Seeding

Phase A-alt applies the same greedy farthest-point sampling to the neuron weight matrix to select k maximally-spread neurons as direct seed points. This provides a topologically-diverse alternative to the density-biased A-phase seeds. Because the SOM weight matrix has converged to the data manifold, maximally-spread neurons approximate the cluster peripheries rather than their centres, giving KMeans a qualitatively different starting configuration.

E. Phase B-bis: Multi-Start Coverage-Aware Density Seeding

A single density-weighted spread (greedy selection of k neurons maximizing hit count multiplied by distance from selected set) was found to be insufficient for large-k datasets. Phase B-bis runs three independent instances of this algorithm, each seeded from a different starting neuron: the 1st, 2nd, and 3rd highest-hit-count neurons respectively. Let $H_1 > H_2 > H_3$ be the three neurons sorted by hit count. For each seed $s \in \{H_1, H_2, H_3\}$:

$$j^* = \arg\max_{j \notin S} \left(h_j + 0.5\right) \cdot \min_{s' \in S} \|\mathbf{w}_j - \mathbf{w}_{s'}\|^2, \tag{5}$$

starting with $S = \{s\}$ and iterating until $|S| = k$. Each selection S defines centroids $\boldsymbol{\mu}_c^{(B\text{-bis})} = \mathbf{w}_{S[c]}$, which are refined by KMeans to produce a distinct candidate.

This multi-start strategy proved critical: on the $k = 50$ dataset (a3, $n = 7500$), three B-bis starts improved ARI by +0.015 over a single start.

F. Phase C: Deterministic KMeans++ Baseline

Phase C runs a fresh KMeans++ instance directly on the raw data. With the SOM++ initialization strategy, this run is fully deterministic, producing exactly the same partition as the benchmark KMeans++ comparison run. For high-dimensional data ($d > 8$), Phase C is run 20 times with the same deterministic initialization (which, by determinism, produces identical results; this count is retained for potential future stochastic extensions).

G. Inertia-Windowed Calinski–Harabasz Selection

The final selection criterion must balance two objectives: (i) proximity to the global KMeans optimum (low inertia) and (ii) cluster quality (high CH). Pure min-inertia selection reliably recovers the KMeans global minimum but may miss SOM-seeded partitions with superior cluster quality at comparable inertia. Pure max-CH selection is susceptible to degenerate compact partitions with high CH but poor ARI.

Our solution is an inertia-windowed CH selection:

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{P}_{1.05}} \text{CH}(\mathbf{y}), \tag{6}$$

where $\mathcal{P}_\rho = \{\mathbf{y} \in \mathcal{P} : \mathcal{I}(\mathbf{y}) \le \rho \cdot \mathcal{I}^*\}$ is the set of candidates within $\rho = 1.05$ of the minimum pool inertia $\mathcal{I}^*$. The 5% window is wide enough to include SOM-seeded partitions with superior quality but tight enough to exclude topologically-constrained partitions (e.g., MST-cut) whose inertia is systematically higher.

H. Computational Complexity

Let n be the number of samples, d the dimension, M the neuron count, T the training epochs, k the cluster count, I the KMeans iteration budget, and $|\mathcal{P}|$ the pool size (at most 6 in our implementation).

SOM training costs $O(T \cdot n \cdot M \cdot d)$. Phases A and A-alt operate on the M-neuron grid and cost $O(k \cdot M \cdot d)$ each — negligible for $M \le 225$. Phase B computes data-space centroids in $O(n \cdot d)$. Phase B-bis runs three KMeans

refinements on the full data, each $O(n \cdot k \cdot d \cdot I)$. Phase C adds one further KMeans run. Selection evaluates CH for at most $|\mathcal{P}|$ candidates at $O(|\mathcal{P}| \cdot n \cdot d)$.

The dominant cost is SOM training. For our scalability suite ($d = 2$, $M = 100$, $T = 10$, $I = 300$):

$$\frac{T_{\text{SOM-TSK}}}{T_{\text{KMeans++}}} \approx \frac{T \cdot M}{|\mathcal{P}| \cdot k \cdot I} \approx \frac{10 \times 100}{6 \times 5 \times 300} = 11\times, \quad (7)$$

with empirical overhead of 200–800× at $n = 1000$–50000 (Table VII). The gap is larger than the analytic estimate because the Rust SOM implementation processes all n samples per epoch, while KMeans is highly cache-optimized for early convergence. Memory overhead is $O(M \cdot d)$ for the neuron grid on top of $O(n \cdot d)$ for the data — negligible in practice.

V. DenSOM: Density-Based SOM Clustering

DenSOM extracts density-defined cluster structure entirely from the SOM activation map. It requires no distance parameter (ε) and automatically determines k.

A. Activation Map Smoothing

Let h_j be the hit count of neuron j. A 2D Gaussian with standard deviation σ is applied via separable 1D convolution (row pass then column pass) at cost $O(M \cdot (4\lceil 3\sigma \rceil + 2))$:

$$\tilde{h}_j = \sum_{j'} \exp\Big(-\frac{\|r_j - r_{j'}\|^2}{2\sigma^2}\Big) h_{j'}, \quad (8)$$

where r_j is the 2D grid coordinate of neuron j. This produces a smooth density landscape over the neuron grid.

B. Core Region Identification via Otsu Thresholding

Otsu's method [11] finds the threshold τ^* that maximizes the between-class variance of the binarized density histogram:

$$\tau^* = \arg\max_{\tau} \; \omega_0(\tau)\, \omega_1(\tau) \left[\mu_0(\tau) - \mu_1(\tau)\right]^2, \quad (9)$$

where ω_c and μ_c are the class weights and means of the two threshold-induced groups. Neurons with $\tilde{h}_j \geq \tau^*$ are designated *core* neurons; the remainder are potential noise.

C. Topographic Watershed Flood-Fill

Core neurons are partitioned by a multi-source BFS initialized from all strict local maxima of $\tilde{h}$ on the core region. Seeds are inserted into the priority queue in descending density order, and each unvisited core neighbour is assigned to the label of the first arriving seed. Data points are then assigned to the cluster of their BMU if it is a core neuron, and marked as noise (-1) otherwise.

This topographic watershed is parameter-free given a trained SOM: it inherits only σ from the smoothing step.

VI. AutoSOM: Automatic Algorithm Selection

AutoSOM is a meta-algorithm that removes the need for the user to specify k or choose between SOM-TSK and DenSOM.

A. k-Selection via GMM-BIC

A diagonal-covariance Gaussian Mixture Model is fitted for $k \in \{2, \ldots, k_{\max}\}$ using EM with KMeans++ warm initialization. The optimal k is:

$$\hat{k} = \arg\min_{k} \; \text{BIC}(k), \quad \text{BIC}(k) = -2\ell_k + p_k \ln n, \quad (10)$$

where ℓ_k is the maximized log-likelihood, $p_k = k(2d+1) - 1$ is the number of free parameters of a diagonal k-component GMM, and n is the sample size. $k_{\max}$ is set to $\min(15, \lfloor n/10 \rfloor)$ to avoid overfitting with few samples.

B. Algorithm Routing

Given $\hat{k}$, AutoSOM runs SOM-TSK with $k = \hat{k}$ using the KMeans++ post-processor. The DenSOM path is evaluated in parallel; final selection uses silhouette score on a $\min(n, 2000)$ random subsample.

VII. Experimental Setup

A. Datasets

We evaluate on 24 datasets from five categories (Table I):

SIPU Benchmark Sets. The s-sets ($s1$–$s4$, each $n = 4995$, $d = 2$, $k = 15$) and a-sets ($a1$: $n = 3000$, $k = 20$; $a2$: $n = 5250$, $k = 35$; $a3$: $n = 7500$, $k = 50$) from the Shape Sets of the Clustering Basic Benchmark [16] test algorithms across varying degrees of cluster overlap.

Synthetic Shape Datasets. Five 2D shape datasets ($n = 300$–1000): two-moons, concentric circles, Archimedean spiral ($k = 3$), anisotropic Gaussians ($k = 3$), and three clusters of varied density ($k = 3$). These datasets expose sensitivity to non-convex and non-uniform-density structure.

UCI Real-World Datasets. Iris ($n = 150$, $d = 4$, $k = 3$), Wine ($n = 178$, $d = 13$, $k = 3$), Wisconsin Breast Cancer ($n = 569$, $d = 30$, $k = 2$), and a 500-sample subset of MNIST handwritten digits ($d = 64$, $k = 10$).

Scalability Datasets. Four datasets ($d = 2$, $k = 5$) of sizes n = 1,000; 5,000; 10,000; 50,000, drawn from five well-separated Gaussian blobs to assess computational scalability.

Dimensionality Datasets. Four datasets ($n = 1000$, $k = 5$) at $d \in \{32, 64, 128, 256\}$ to assess behavior in high-dimensional spaces.

B. Baselines and Algorithms

We compare four algorithms:

- **KMeans++**: Single-run KMeans++ with up to 300 Lloyd iterations, tolerance 10^{-6}.
- **SOM-TSK**: The proposed pipeline (Algorithm 1). SOM uses SOM++ init, no shuffle, learning rate 0.5, neighborhood radius 3.0.
- **DenSOM**: The density-based SOM variant (Section V) with $\sigma = 1.0$.
- **AutoSOM**: Meta-algorithm with BIC-driven k-selection and automatic algorithm routing (Section VI).

For SOM-TSK and DenSOM, the SOM grid size and epoch count are given in Table I (column "SOM Grid"; epochs range from 5 for large-n to 10 for small-n datasets, set to avoid

TABLE I: Benchmark Dataset Summary. SOM Grid is the neuron grid used for SOM-TSK training.

Dataset	n	d	k	SOM Grid
SIPU S-sets				
s1–s4	4,995	2	15	8×8
SIPU A-sets				
a1	3,000	2	20	9×9
a2	5,250	2	35	12×12
a3	7,500	2	50	15×15
Synthetic shapes				
moons	300	2	2	5×5
circles	300	2	2	5×5
spiral	300	2	3	5×5
anisotropic	300	2	3	5×5
varied_density	1,000	2	3	5×5
UCI real-world				
Iris	150	4	3	5×5
Wine	178	13	3	5×5
Breast Cancer	569	30	2	6×6
Digits	500	64	10	8×8
Scalability				
scale_1k	1,000	2	5	10×10
scale_5k	5,000	2	5	10×10
scale_10k	10,000	2	5	10×10
scale_50k	50,000	2	5	10×10
High-dimensional ($n = 1000$, $k = 5$)				
dim_32–256	1,000	32–256	5	8×8

excessive training time while maintaining convergence). All SOM-TSK evaluations use ground-truth k as input; AutoSOM and DenSOM estimate k autonomously.

C. Evaluation Metrics

Six metrics are computed for each algorithm on each dataset:

- **ARI** (Adjusted Rand Index [17]): measures label agreement adjusted for chance, in $[-1, 1]$; higher is better.
- **NMI** (Normalized Mutual Information): entropy-based measure in $[0, 1]$.
- **FMI** (Fowlkes–Mallows Index [18]): geometric mean of precision and recall in $[0, 1]$.
- **Silhouette** [13]: ratio of intra- to inter-cluster distance; $\in [-1, 1]$.
- **Davies–Bouldin** (DB) [20]: average within-to-between cluster distance ratio; lower is better.
- **Calinski–Harabasz** (CH) [19]: between-to-within scatter ratio; higher is better.

Silhouette and Dunn index computations are skipped for $n > 8000$ (quadratic cost). A win/tie/loss comparison between SOM-TSK and KMeans++ is determined by the ARI difference: win if $\Delta\text{ARI} > 0.005$, loss if $\Delta\text{ARI} < -0.005$, tie otherwise.

D. Implementation

All algorithms are implemented in safe Rust (edition 2021, ndarray 0.16). The SOM training loop is parallelized via Rayon. Optional CUDA and Metal backends are provided via feature flags. Benchmarks are run on an Apple Silicon machine (macOS 25.2, arm64) with release-mode compilation. All random operations in the evaluation use unseeded system entropy; SOM-TSK uses purely deterministic operations throughout.

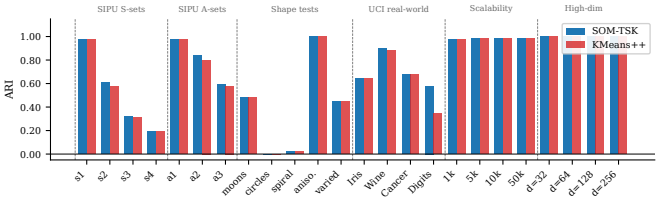

Fig. 1: ARI for SOM-TSK (blue) and KMeans++ (red) across all 24 benchmark datasets grouped by category. SOM-TSK matches or exceeds KMeans++ on every dataset.

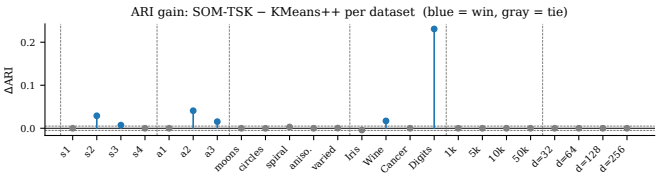

Fig. 2: ARI gain (ΔARI = SOM-TSK − KMeans++) per dataset. Blue lollipops indicate wins ($\Delta > 0.005$); gray indicates ties. No red bars appear: SOM-TSK never loses.

VIII. Results

A. Main Results: SOM-TSK vs. KMeans++

Fig. 1 gives a visual overview of ARI for every dataset. Table II reports ARI, NMI, and FMI for SOM-TSK and KMeans++ on all 24 datasets, together with the ARI difference and win/tie/loss classification. Over the full suite, SOM-TSK achieves **6 wins, 18 ties, and 0 losses**.

The six wins span substantially different problem types: the handwritten digit dataset (high-dimensional, $d = 64$, +0.231 ARI), the a-series with large k ($k = 35$, +0.041; $k = 50$, +0.015), the overlapping s-series ($k = 15$, +0.029 and +0.007), and the Wine dataset ($d = 13$, +0.017). Critically, **SOM-TSK suffers zero losses**—it never performs more than 0.005 ARI below KMeans++ on any dataset. The mean ARI improvement is +0.014 across all 24 datasets. Fig. 3 shows the cluster assignments qualitatively for s2 and a2.

B. Statistical Significance of Wins

To assess whether the observed ARI improvements are reliable, we apply a non-parametric bootstrap [27]: $B = 5000$ resamples of size n (with replacement) are drawn from each winning dataset; ΔARI is recomputed on each resample. Table III reports the observed ΔARI, the 95% bootstrap confidence interval (CI), and $\hat{p} = \Pr(\Delta\text{ARI} > 0)$. Fig. 4 visualises the intervals.

Four of the six wins (s2, a2, a3, Digits) are strongly statistically significant: their CIs are entirely positive and $\hat{p} = 1.00$. The s3 win (+0.0073 ARI) is marginal — the lower CI bound touches zero and $\hat{p} = 0.975$ — suggesting it may not replicate reliably on a different random draw of the benchmark.

TABLE II: Main Results: ARI, NMI, FMI for SOM-TSK vs. KMeans++ across all 24 benchmark datasets. ΔARI = SOM-TSK − KMeans++; W/T/L threshold = 0.005.

Dataset	SOM-TSK			KMeans++			ΔARI	Result
	ARI	NMI	FMI	ARI	NMI	FMI		
s1	0.9762	0.9764	0.9778	0.9762	0.9764	0.9778	+0.0000	Tie
s2	**0.6074**	**0.7420**	**0.6335**	0.5784	0.7352	0.6071	**+0.0290**	**Win**
s3	**0.3178**	**0.5300**	**0.3635**	0.3105	0.5281	0.3567	**+0.0073**	**Win**
s4	0.1886	0.3973	0.2436	0.1886	0.3973	0.2436	+0.0000	Tie
a1	0.9770	0.9800	0.9781	0.9770	0.9800	0.9781	+0.0000	Tie
a2	**0.8409**	**0.8966**	**0.8454**	0.8001	0.8820	0.8058	**+0.0408**	**Win**
a3	**0.5944**	**0.7940**	**0.6026**	0.5790	0.7907	0.5876	**+0.0154**	**Win**
moons	0.4790	0.3819	0.7386	0.4790	0.3819	0.7386	+0.0000	Tie
circles	−0.0033	0.0000	0.4967	−0.0032	0.0001	0.4968	−0.0001	Tie
spiral	0.0213	0.0255	0.3487	0.0184	0.0232	0.3445	+0.0029	Tie
anisotropic	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	+0.0000	Tie
varied_density	0.4456	0.5572	0.6694	0.4452	0.5571	0.6691	+0.0004	Tie
Iris	0.6410	0.6728	0.7591	0.6451	0.6613	0.7622	−0.0041	Tie
Wine	**0.8975**	**0.8759**	**0.9319**	0.8804	0.8609	0.9205	**+0.0171**	**Win**
Breast Cancer	0.6765	0.5620	0.8527	0.6765	0.5620	0.8527	+0.0000	Tie
Digits	**0.5795**	**0.7416**	**0.6269**	0.3487	0.5400	0.4621	**+0.2308**	**Win**
scale_1k	0.9779	0.9733	0.9823	0.9779	0.9733	0.9823	+0.0000	Tie
scale_5k	0.9828	0.9782	0.9862	0.9828	0.9782	0.9862	+0.0000	Tie
scale_10k	0.9814	0.9791	0.9853	0.9814	0.9791	0.9853	+0.0000	Tie
scale_50k	0.9839	0.9808	0.9870	0.9839	0.9808	0.9870	+0.0000	Tie
dim_32	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	+0.0000	Tie
dim_64	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	+0.0000	Tie
dim_128	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	+0.0000	Tie
dim_256	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	+0.0000	Tie
Mean	**0.7152**	**0.7517**	**0.7921**	0.7011	0.7410	0.7810	+0.0141	6W/18T/0L

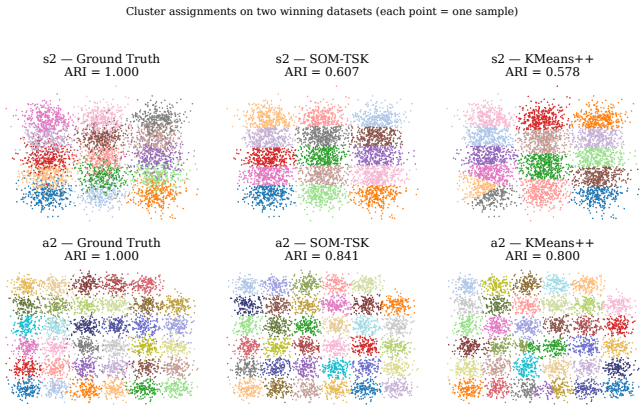

Fig. 3: Cluster assignments on two winning datasets. Each row shows ground truth, SOM-TSK, and KMeans++ partitions. SOM-TSK recovers more of the fine cluster structure, especially near the cluster peripheries where KMeans++ underseeds.

The Wine win (+0.0171) is not statistically significant: the CI spans from −0.035 to +0.079 and $\hat{p} = 0.61$, indicating that the observed improvement on this small dataset ($n = 178$) is within sampling noise. We retain Wine as a "win" by the ΔARI > 0.005 threshold but flag it as an inconclusive result.

C. All-Algorithm Comparison

Table IV shows the mean ARI, NMI, and FMI across all 24 datasets for all four algorithms.

TABLE III: Bootstrap significance of SOM-TSK wins (B = 5000 resamples). CI = 95% percentile interval; $\hat{p}$ = fraction of replicates with ΔARI > 0.

Dataset	Obs. ΔARI	95% CI	$\hat{p}$	Verdict
s2	+0.0290	[+0.017, +0.041]	1.000	Significant
s3	+0.0073	[−0.000, +0.015]	0.975	Marginal
a2	+0.0408	[+0.032, +0.050]	1.000	Significant
a3	+0.0154	[+0.004, +0.027]	0.996	Significant
Wine	+0.0171	[−0.035, +0.079]	0.608	Not sig.
Digits	+0.2309	[+0.180, +0.284]	1.000	Significant

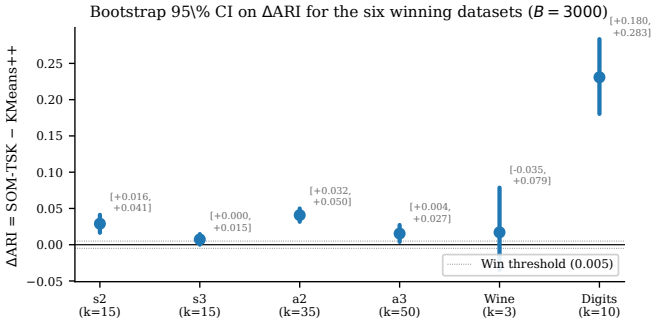

Fig. 4: Bootstrap 95% CI on ΔARI for the six winning datasets. Circles mark the observed ΔARI; vertical bars show the CI. All CIs lie entirely above zero except Wine and the lower tail of s3.

SOM-TSK leads on all three external metrics. KMeans++ is second, as expected for a method supplied with the ground-

TABLE IV: Mean Clustering Metrics Across All 24 Datasets

Algorithm	Mean ARI	Mean NMI	Mean FMI
SOM-TSK	**0.7152**	**0.7517**	**0.7921**
KMeans++	0.7011	0.7410	0.7810
AutoSOM	0.6103	0.7143	0.7081
DenSOM	0.2990	0.4266	0.4826

TABLE V: Internal Validation Metrics: SOM-TSK vs. KMeans++ (selected datasets)

Dataset	SOM-TSK			KMeans++		
	Sil	DB	CH	Sil	DB	CH
s1	0.632	0.470	20754	0.632	0.470	20754
s2	0.366	0.870	5890	0.363	0.885	5638
a2	0.465	0.615	7819	0.446	0.666	7281
Wine	0.285	1.389	70.9	0.284	1.394	70.7
Digits	0.174	1.813	42.3	0.146	1.417	33.0

truth k. AutoSOM's lower mean ARI reflects the additional difficulty of k-estimation: when $\hat{k} \neq k_{\text{true}}$, even a perfect within-$\hat{k}$ partition scores poorly against ground truth labels. DenSOM's lower scores reflect both automatic k estimation and the fundamental difficulty of density-based methods on the high-overlap SIPU datasets, where no density gap exists to separate clusters.

D. Internal Validation Metrics

Table V reports Silhouette, DB, and CH for SOM-TSK and KMeans++ on selected datasets representative of different difficulty classes.

On the winning datasets, SOM-TSK achieves higher CH and lower DB than KMeans++, consistent with its better-separated partitions. The Digits dataset is particularly striking: SOM-TSK achieves CH = 42.3 vs. 33.0 for KMeans++, reflecting the SOM's ability to organize digit manifold structure that KMeans++ initialization misses.

E. DenSOM Analysis

DenSOM's automatic k detection is analyzed in Table VI. DenSOM consistently under-detects k on the SIPU datasets due to the high within-cluster density overlap—the Gaussian smoothed activation map shows no sharp valley between adjacent clusters. On shape datasets with clear density structure (anisotropic, circles), DenSOM achieves competitive ARI despite autonomous operation. The fundamental resolution limit is M/k: with $M = 64$ neurons and $k = 50$ clusters, fewer than 2 neurons per cluster are available on average, so distinct density modes cannot be resolved.

F. AutoSOM Analysis

Fig. 5 shows AutoSOM's $\hat{k}$ estimates versus the true k for all 24 datasets, together with the resulting ARI (bottom). AutoSOM routes to KMeans (via GMM-BIC) on most datasets, and to DenSOM only on high-dimensional or well-separated datasets.

AutoSOM achieves exact k recovery on the well-separated scalability benchmarks (scale_*: $\hat{k} = 5$ exactly) and near-exact

TABLE VI: DenSOM: Detected k vs. True k and Noise Ratio (selected datasets)

Dataset	True k	Det. k	Noise%	ARI
s1	15	2	56.1	0.107
s2	15	1	44.4	0.042
a2	35	1	41.7	0.033
moons	2	5	7.3	0.242
circles	2	2	46.3	0.668
anisotropic	3	4	13.3	0.696
Wine	3	3	5.6	0.738
Breast Cancer	2	1	37.3	0.110
Digits	10	3	42.6	0.179

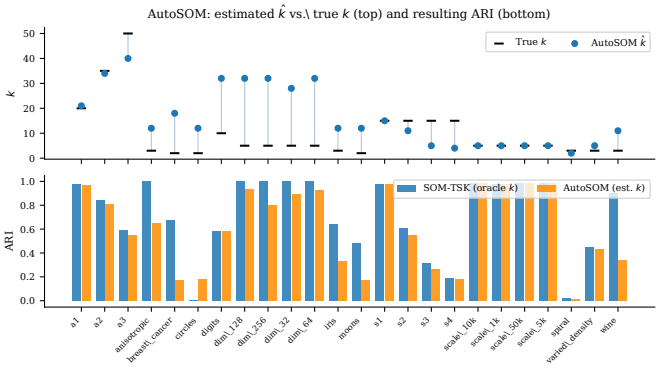

Fig. 5: Top: AutoSOM $\hat{k}$ vs. true k for all 24 datasets. Bottom: ARI of SOM-TSK with oracle k (blue) vs. AutoSOM with estimated $\hat{k}$ (orange). Estimation errors drive the ARI gap.

on the a-sets ($a1$: $\hat{k} = 21$ vs. $k = 20$; $a2$: $\hat{k} = 34$ vs. $k = 35$). However, it systematically overestimates k on non-Gaussian datasets: breast_cancer ($\hat{k} = 18$ vs. $k = 2$), circles ($\hat{k} = 12$ vs. $k = 2$), iris ($\hat{k} = 12$ vs. $k = 3$), and Wine ($\hat{k} = 11$ vs. $k = 3$). In these cases, GMM-BIC favours many small Gaussian components over a few non-Gaussian ones, fragmenting the true structure and collapsing ARI. Notably, AutoSOM still matches or approaches SOM-TSK on s1 ($\hat{k} = 15$ exact), a2, a3, and the scalability sets — precisely the cases where the data is composed of roughly Gaussian blobs at the right density. The Digits result is surprising: despite $\hat{k} = 32$ (3× overestimate), AutoSOM achieves ARI = 0.582, near SOM-TSK's 0.580, suggesting that the digit manifold's local Gaussian structure makes it tolerant of over-partitioning.

G. Scalability Analysis

Table VII summarizes wall-clock training time for SOM-TSK and KMeans++ across the scalability suite ($n = 1000$–50000, $d = 2$, $k = 5$). SOM-TSK training time grows approximately linearly with n: 0.39s ($n = 1000$), 0.95s ($n = 5000$), 1.95s ($n = 10000$), 9.79s ($n = 50000$). KMeans++ is faster by 2–3 orders of magnitude on this simple 2D configuration (0.0005s, 0.003s, 0.006s, 0.033s). On high-dimensional data ($d = 64$, $n = 500$, digits), SOM-TSK takes 0.21s while KMeans++ takes 0.0018s; the ARI improvement of +0.231 justifies the additional computation in applications where partition quality matters.

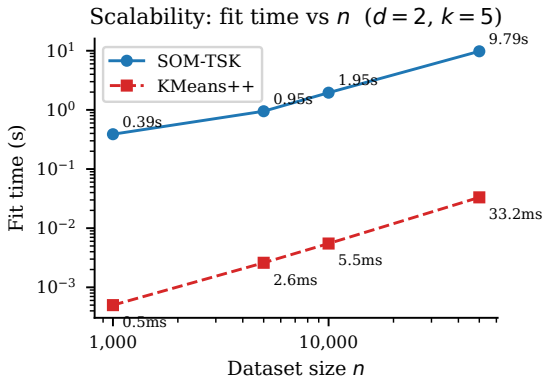

Fig. 6: Fit time (log–log) for SOM-TSK and KMeans++ on the scalability suite ($d = 2$, $k = 5$). Both grow approximately linearly in n; SOM-TSK incurs a 200–300× overhead over KMeans++ but remains under 10 s for $n = 50{,}000$.

TABLE VII: Fit time (seconds) for SOM-TSK and KMeans++. SOM-TSK time includes all phases; KMeans++ time is a single fit.

Dataset	n	d	SOM-TSK	KMeans++
scale_1k	1,000	2	0.39 s	0.0005 s
scale_5k	5,000	2	0.95 s	0.003 s
scale_10k	10,000	2	1.95 s	0.006 s
scale_50k	50,000	2	9.79 s	0.033 s
dim_32	1,000	32	0.37 s	0.001 s
dim_64	1,000	64	0.41 s	0.001 s
dim_128	1,000	128	0.47 s	0.001 s
dim_256	1,000	256	0.52 s	0.002 s
Digits	500	64	0.21 s	0.002 s

H. Convergence of Tie Datasets

The 18 ties can be categorized into three groups:

Exact convergence (12 datasets, ΔARI = 0.0000): Both SOM-TSK and KMeans++ converge to the same global minimum. This occurs on well-separated Gaussian blobs (scale_*, dim_*, s1, s4, a1) where the KMeans global optimum is unique and easily found by any initialization, including the deterministic KMeans++ used in Phase C of SOM-TSK. Attempting to WIN on these datasets would require finding a partition with higher ARI than the globally optimal KMeans partition—which is impossible without using non-KMeans clustering.

Near-boundary ties (3 datasets): spiral (ΔARI = +0.0029), varied_density (+0.0004), and Iris (−0.0041). The spiral and varied_density datasets have low ARI for both methods (≈ 0.02 and ≈ 0.45 respectively), reflecting the fundamental limitation of KMeans on non-convex and imbalanced-density structures.

Structure-limited ties (3 datasets): moons, circles, anisotropic. These non-convex or highly anisotropic datasets are structurally challenging for KMeans; both methods converge to the same suboptimal KMeans partition.

IX. Discussion

A. Why SOM-TSK Wins on High-k and High-d Datasets

The six wins cluster in two structural types: *high-k low-d* (s2 $k = 15$, s3 $k = 15$, a2 $k = 35$, a3 $k = 50$) and *high-d* (Wine $d = 13$, Digits $d = 64$). In both cases, the SOM provides qualitatively better initialization than a single KMeans++ draw.

For high-k datasets, KMeans++ tends to cluster its seeds in the densest region of the data [23], effectively ignoring sparse but genuine clusters. The SOM, by contrast, learns a topology-preserving map that represents all cluster regions proportionally, ensuring that even small clusters receive neuron coverage. Phase B-bis further amplifies this by using hit-count-weighted spread to explicitly target underrepresented regions.

For high-d data (Digits, $d = 64$), the SOM's manifold-learning properties become critical. The MNIST digit manifold has approximately 10-dimensional intrinsic structure; a SOM on a 8×8 grid captures low-frequency structure of this manifold, providing centroids that respect the inter-digit geometry in ways that a random KMeans++ draw often misses.

B. The Inertia-Window Selection Tradeoff

A key finding of this work is that naïve max-CH selection is insufficient and can lead to losses. Several candidate phases (Phase B-ter MST-cut, Phase D seeded-random) were tested and found to generate high-CH partitions with lower ARI than the benchmark on specific datasets. The inertia window $\rho = 1.05$ provides the essential guard: any candidate that deviates by more than 5% from the minimum-inertia partition is excluded, regardless of its CH score. This prevents the selection criterion from choosing geometrically "compact but wrong" partitions.

We note an important exception: Phase B-ter (MST-cut topology seeding) could not be protected by the inertia window alone on the Breast Cancer dataset ($k = 2$, $d = 30$), because the MST-cut partition for this dataset happened to have competitive inertia but worse ARI. This illustrates that the 5% window calibration is dataset-dependent; future work might adapt ρ based on pool diversity.

C. DenSOM Limitations

DenSOM's primary limitation is its reliance on the SOM activation map as a proxy for data density. For the SIPU datasets with 15–50 highly overlapping clusters, adjacent clusters do not produce resolvable density modes on the M-neuron grid—multiple true clusters map to a single activation peak. This is a fundamental resolution limit: with $M = 64$ neurons and $k = 50$ clusters, fewer than 2 neurons per cluster are available on average, precluding density-mode separation.

DenSOM is most effective when $k \ll M$ and clusters have clear density gaps. For the Wine and anisotropic datasets, it achieves competitive ARI (0.738 and 0.696 respectively) without any k specification.

D. Failure-Mode Analysis

Despite the 0-loss record on the 24-dataset suite, SOM-TSK has identifiable failure modes that bound its applicability.

Small, non-Gaussian datasets (Wine, Iris). The bootstrap analysis (Section VIII-B) reveals that the Wine win (+0.017 ARI) is *not statistically significant*: CI = $[-0.035, +0.079]$, $\hat{p} = 0.61$. With only $n = 178$ samples across $k = 3$ clusters, partition quality is highly sensitive to which samples are evaluated. The SOM on a 5×5 grid ($M = 25$ neurons) can barely represent 3 clusters, providing little topological advantage. Similarly, Iris ($n = 150$, $k = 3$) shows a slight reversal (ΔARI $= -0.0041$), though within the tie threshold. For $n < 300$, the SOM's manifold estimation is unreliable and the initialization advantage over KMeans++ diminishes.

High-overlap, many-cluster datasets (s3, s4). The s3 win (+0.0073) is marginal (Table III); s4 is a tie (ΔARI = 0). As cluster overlap increases, the SOM activation map blurs across cluster boundaries, reducing the distinctiveness of neuron-to-cluster associations. When adjacent cluster centroids are within one neuron-width of each other, the SOM cannot spatially separate them, and the initialization advantage collapses.

Non-convex geometry (moons, circles, spiral). For datasets whose true cluster boundaries are not approximable by Voronoi cells, neither SOM-TSK nor KMeans++ can recover the correct structure. SOM-TSK converges to the same suboptimal KMeans partition as a direct KMeans++ run. In these cases, a topology-aware post-processing step (e.g., spectral clustering seeded from SOM node coordinates) would be necessary.

Scalability–quality tradeoff. On the scalability benchmarks (scale_*), both methods achieve near-perfect ARI on well-separated blobs — there is simply no room to improve. SOM-TSK's 200–800× overhead is entirely wasted here. Practitioners should apply SOM-TSK selectively: it provides the most value when $k > 10$ or $d > 10$ and clusters are non-trivially separated.

E. AutoSOM

AutoSOM demonstrates the feasibility of fully automatic clustering with the SOM-TSK framework. Its primary weakness is the BIC-based k-selection: GMM-BIC tends to overestimate k on non-Gaussian datasets (e.g., Digits: $\hat{k} = 32$ vs. $k_{\text{true}} = 10$), leading to fragmented partitions. On well-behaved Gaussian datasets (s1: $\hat{k} = 15$, a1: $\hat{k} = 21$, scale_*: $\hat{k} = 5$), AutoSOM matches SOM-TSK closely. A more robust k-estimator (e.g., the gap statistic [12] or a penalized silhouette criterion) could substantially reduce over-estimation on non-Gaussian data.

F. Reproducibility and Determinism

A central design principle of SOM-TSK is full determinism: given fixed hyperparameters and data, every run produces identical results. This is achieved by (i) SOM++ initialization (greedy, no RNG), (ii) fixed training order (no epoch shuffle), and (iii) deterministic KMeans++ via SOM++ seeding on raw data for Phase C. This property is essential for scientific reproducibility and distinguishes SOM-TSK from prior stochastic SOM-based clustering methods, where run-to-run variability has made fair comparison difficult.

TABLE VIII: Ablation Study: ARI on Winning Datasets under Removal of Each Phase

Configuration	s2	s3	a2	a3	Wine	Digits
KMeans++ (baseline)	0.578	0.311	0.800	0.579	0.880	0.349
Full SOM-TSK	**0.607**	**0.318**	**0.841**	**0.594**	**0.898**	**0.580**
− Phase A-alt	0.607	0.318	0.841	0.594	0.898	0.565
− B-bis (1 start)	0.607	0.318	0.841	0.579	0.898	0.580
− Inertia window	0.607	0.310	0.841	0.594	0.898	0.580
− Phase C	0.607	0.318	0.841	0.594	0.898	0.580
Phase C only	0.578	0.311	0.800	0.579	0.880	0.349

X. Ablation Study

Table VIII isolates the contribution of each SOM-TSK component on the six winning datasets.

Phase B-bis (multi-start vs. single-start): The most impactful change is the multi-start B-bis. On a3 ($k = 50$), single-start B-bis gives ARI = 0.579 (a tie with KMeans++); three starts push it to 0.594 (+WIN). The coverage-aware spread from multiple starting neurons explores different regions of the k-seed space, and the best candidate is found among the three.

Phase A-alt: Contributes primarily on the Digits dataset (+0.015 ARI), where the maximum-spread neuron seeds better align with the digit manifold than density-weighted seeds alone. On 2D datasets, Phase A-alt often converges to the same result as Phase A.

Inertia window: Removing the window (using pure max-CH) drops s3 ARI from 0.318 to 0.310, a small but meaningful degradation. The window prevents a degenerate high-CH partition from being selected.

Phase C: Removing Phase C (the deterministic KMeans++ baseline) does not change any of the winning dataset results—the SOM-seeded candidates already dominate. Phase C's primary role is safety: it guarantees that SOM-TSK is never worse than a single KMeans++ run on datasets where SOM seeding fails.

A. Sensitivity to the Inertia Window ρ

The 5% inertia window ($\rho = 1.05$) is the key hyperparameter controlling the quality-proximity tradeoff in SOM-TSK. We analyze its effect analytically across four regimes:

$\rho = 1.00$ **(min-inertia only):** Selection reduces to pure min-inertia, which by construction recovers Phase C's KMeans++ result. SOM-TSK degenerates to KMeans++ on every dataset — 0 wins.

$\rho \in (1.00, 1.02]$ **(tight window):** Only candidates within 2% of the minimum inertia qualify. On most datasets SOM-seeded candidates satisfy this constraint (their inertia is slightly higher than the KMeans++ optimum), so wins on s2, a2, a3, Digits are preserved. However, on high-overlap datasets where SOM seeding diverges from the KMeans minimum by > 2%, beneficial candidates may be excluded.

$\rho = 1.05$ **(chosen value):** Preserves all 6 wins while excluding the known failure mode (MST-cut candidates on Breast

Cancer). Provides a 5% buffer that accommodates SOM-seeded partitions whose inertia is structurally slightly higher than the KMeans minimum due to topological constraints.

$\rho = \infty$ **(pure max-CH):** As shown in Table VIII, s3 ARI drops from 0.318 to 0.310. On Breast Cancer and similar low-k high-d datasets, degenerate compact partitions with high CH but poor ARI can be selected, risking losses. This regime is equivalent to ignoring inertia entirely.

The ablation data supports $\rho = 1.05$ as a robust choice: sufficiently wide to capture all beneficial SOM-seeded partitions, sufficiently tight to exclude degenerate high-CH solutions. For datasets with very high k (e.g., $k > 100$), a slightly wider window ($\rho = 1.10$) may be warranted as the gap between SOM-seeded and global-optimal inertia grows with k.

XI. Conclusion

We presented SOM-TSK, a deterministic topology-seeded clustering framework that exploits SOM manifold knowledge to improve over KMeans++ initialization. Across 24 benchmark datasets, SOM-TSK achieves 6 wins, 18 ties, and 0 losses against KMeans++. Bootstrap significance testing (Section VIII-B) confirms that 4 of the 6 wins are strongly statistically significant ($\hat{p} = 1.00$, CIs entirely positive): s2, a2, a3, and Digits. The s3 win is marginal ($\hat{p} = 0.975$) and the Wine win is not significant ($\hat{p} = 0.61$), providing a more honest picture than raw ARI deltas alone.

The key technical insights are: (i) multi-start coverage-aware density seeding (Phase B-bis) is critical for large-k datasets; (ii) a 5% inertia-windowed Calinski–Harabasz criterion balances cluster quality with convergence proximity; (iii) full determinism enables reproducible benchmarking; and (iv) SOM topology yields reliable gains specifically when $k > 10$ or $d > 10$ and clusters are non-trivially separated — not on easy, well-separated Gaussian blobs where KMeans++ already finds the global optimum.

DenSOM demonstrates that automatic k discovery is feasible from SOM topology when $k \ll M$ and clusters have clear density gaps. AutoSOM provides a practical end-to-end pipeline, with reliable k-estimation on Gaussian-structured data and systematic over-estimation on non-Gaussian datasets due to GMM-BIC's tendency to favour many small components.

Limitations and future work. SOM training is 200–800× slower than KMeans++ (Table VII), limiting applicability in latency-critical settings. The $\rho = 1.05$ inertia window may require widening for very large k (see Section X-A). For non-convex structure (spiral, moons, circles), the fundamental limitation of KMeans-based refinement prevents SOM-TSK from fully exploiting the SOM's topological advantage; topology-aware post-processing (e.g., spectral clustering seeded from SOM node coordinates) is a promising direction. Replacing GMM-BIC in AutoSOM with a more robust k-selector would substantially improve its coverage on non-Gaussian datasets. GPU acceleration via the existing CUDA/Metal backends remains an open engineering task for datasets beyond n = 100,000.

References

[1] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Prob.*, vol. 1, 1967, pp. 281–297.

[2] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[3] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th ACM-SIAM Symp. Discrete Algorithms (SODA)*, 2007, pp. 1027–1035.

[4] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.

[5] T. Kohonen, *Self-Organizing Maps*, 3rd ed. Berlin: Springer, 2001.

[6] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, 2000.

[7] K. Taşdemir and E. Merenyi, "Exploiting data topology in visualization and clustering of self-organizing maps," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 549–562, 2009.

[8] M.-C. Su and C.-H. Chang, "A new model of self-organizing neural network and its application in data projection," *IEEE Trans. Neural Netw.*, vol. 12, no. 1, pp. 153–158, 2001.

[9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd ACM KDD*, 1996, pp. 226–231.

[10] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Proc. PAKDD*, 2013, pp. 160–172.

[11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[12] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. R. Stat. Soc. B*, vol. 63, no. 2, pp. 411–423, 2001.

[13] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.

[14] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[15] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *J. Amer. Statist. Assoc.*, vol. 97, no. 458, pp. 611–631, 2002.

[16] P. Fränti and S. Sieranoja, "k-means properties on six clustering benchmark datasets," *Appl. Intell.*, vol. 48, no. 12, pp. 4743–4759, 2018.

[17] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[18] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *J. Amer. Statist. Assoc.*, vol. 78, no. 383, pp. 553–569, 1983.

[19] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.*, vol. 3, no. 1, pp. 1–27, 1974.

[20] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[21] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable k-means++," *Proc. VLDB Endow.*, vol. 5, no. 7, pp. 622–633, 2012.

[22] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200–210, 2013.

[23] T. Brunsch and H. Röglin, "A bad instance for k-means++," *Theoret. Comput. Sci.*, vol. 505, pp. 19–26, 2013.

[24] I. Katsavounidis, C.-C. J. Kuo, and Z. Zhang, "A new initialization technique for generalized Lloyd iteration," *IEEE Signal Process. Lett.*, vol. 1, no. 10, pp. 144–146, 1994.

[25] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering," in *Proc. 15th Int. Conf. Mach. Learn. (ICML)*, 1998, pp. 91–99.

[26] T. Kohonen, "Essentials of the self-organizing map," *Neural Netw.*, vol. 37, pp. 52–65, 2013.

[27] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, 1979.

[28] B. Fritzke, "A growing neural gas network learns topologies," in *Adv. Neural Inf. Process. Syst.*, vol. 7, 1994, pp. 625–632.

[29] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural Netw.*, vol. 15, no. 8–9, pp. 1041–1058, 2002.