Self-Organizing Maps with Optimized Latent Positions

1st Seiki Ubukata
Graduate School of Informatics
Osaka Metropolitan University
Osaka 599-8531, Japan
ubukata@omu.ac.jp

2nd Akira Notsu
Graduate School of Sustainable System Sciences
Osaka Metropolitan University
Osaka 599-8531, Japan
notsu@omu.ac.jp

3rd Katsuhiro Honda
Graduate School of Informatics
Osaka Metropolitan University
Osaka 599-8531, Japan
khonda@omu.ac.jp

***Abstract*—Self-Organizing Maps (SOM) are a classical method for unsupervised learning, vector quantization, and topographic mapping of high-dimensional data. However, existing SOM formulations often involve a trade-off between computational efficiency and a clearly defined optimization objective. Objective-based variants such as Soft Topographic Vector Quantization (STVQ) provide a principled formulation, but their neighborhood-coupled computations become expensive as the number of latent nodes increases. In this paper, we propose Self-Organizing Maps with Optimized Latent Positions (SOM-OLP), an objective-based topographic mapping method that introduces a continuous latent position for each data point. Starting from the neighborhood distortion of STVQ, we construct a separable surrogate local cost based on its local quadratic structure and formulate an entropy-regularized objective based on it. This yields a simple block coordinate descent scheme with closed-form updates for assignment probabilities, latent positions, and reference vectors, while guaranteeing monotonic non-increase of the objective and retaining linear per-iteration complexity in the numbers of data points and latent nodes. Experiments on a synthetic saddle manifold, scalability studies on the Digits and MNIST datasets, and 16 benchmark datasets show that SOM-OLP achieves competitive neighborhood preservation and quantization performance, favorable scalability for large numbers of latent nodes and large datasets, and the best average rank among the compared methods on the benchmark datasets.**

***Index Terms*—self-organizing maps, topographic mapping, vector quantization, entropy regularization, block coordinate descent.**

I. Introduction

Self-Organizing Maps (SOM), introduced by Kohonen [1], are a classical method for unsupervised learning, vector quantization, and topographic mapping of high-dimensional data. Their main appeal lies in organizing reference vectors on a predefined latent topology while preserving neighborhood structure to a useful extent. Owing to this property, SOM and its variants have been widely used in data analysis and visualization. Despite their usefulness, SOM-based methods involve a trade-off between computational efficiency and an explicit objective-based formulation. The original SOM and Batch SOM (BSOM) [2] are efficient and widely used, but their updates are not generally derived from a single explicit smooth objective [3]. In contrast, objective-based methods such as convolved-distortion models [4], [5] and Soft Topographic Vector Quantization (STVQ) [6] offer a clearer optimization perspective, but their explicit neighborhood interactions incur an $O(NM^2)$ per-iteration cost, where N and M are the numbers of data points and map nodes, respectively. Although decomposition techniques can reduce the computational cost [7], neighborhood coupling among latent nodes remains, and each data point is still represented only through assignment weights over predefined discrete nodes.

In this paper, we propose Self-Organizing Maps with Optimized Latent Positions (SOM-OLP), an objective-based topographic mapping method that introduces a continuous latent position for each data point. Starting from the neighborhood distortion of STVQ, we construct a separable surrogate local cost motivated by its local quadratic structure and formulate an entropy-regularized objective based on it. This yields a simple cyclic block coordinate descent (BCD) scheme with closed-form updates for assignment probabilities, latent positions, and reference vectors. As a result, SOM-OLP retains $O(NM)$ per-iteration complexity while providing a continuous latent representation.

The proposed formulation has two main advantages. First, it provides an explicit objective with monotonic non-increase under cyclic BCD. Second, it introduces continuous latent positions while retaining closed-form updates and linear per-iteration complexity without explicit node–node coupling. Experiments on a synthetic saddle manifold, scalability studies on the Digits and MNIST datasets, and 16 benchmark datasets show that SOM-OLP achieves strong neighborhood preservation and quantization performance, favorable scalability for large numbers of latent nodes and large datasets, and the best average rank among the compared methods on the benchmark datasets. A minimal implementation of SOM-OLP is available at https://github.com/subukata/som-olp and on Zenodo (https://doi.org/10.5281/zenodo.19547951).

II. Related Work

Research on SOM has evolved from heuristic update rules to optimization-based and entropy-regularized formulations. In this paper, we focus on optimization-based SOM-type methods and review their representative formulations and limitations after introducing the notation.

This work was supported by JSPS KAKENHI Grant Number JP24K15110.

A. Notation and Problem Setting

Let $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N \subset \mathbb{R}^D$ be a dataset of N data points in the data space $\mathbb{R}^D$, and let $\mathbf{X} = [\boldsymbol{x}_1^\top, \dots, \boldsymbol{x}_N^\top]^\top \in \mathbb{R}^{N\times D}$ be the corresponding data matrix. SOM-type methods represent these data points by M reference vectors arranged on a predefined topology in an L-dimensional latent space $\mathbb{R}^L$. Each node $j \in \{1, \dots, M\}$ has a fixed node coordinate $\boldsymbol{r}_j \in \mathbb{R}^L$ and a reference vector $\boldsymbol{w}_j \in \mathbb{R}^D$. We denote the sets of node coordinates and reference vectors by $\mathcal{R} = \{\boldsymbol{r}_j\}_{j=1}^M$ and $\mathcal{W} = \{\boldsymbol{w}_j\}_{j=1}^M$, and the corresponding matrices by $\mathbf{R} = [\boldsymbol{r}_1^\top, \dots, \boldsymbol{r}_M^\top]^\top \in \mathbb{R}^{M\times L}$ and $\mathbf{W} = [\boldsymbol{w}_1^\top, \dots, \boldsymbol{w}_M^\top]^\top \in \mathbb{R}^{M\times D}$. When soft assignments are used, the correspondence between data points and nodes is represented by $\mathbf{P} = [p_{ij}] \in [0,1]^{N\times M}$ with $\sum_{j=1}^M p_{ij} = 1$ for each data point i, where hard assignments are included as a special case.

B. Classical SOM-Type Methods Without a Single Objective Function

The standard sequential SOM, introduced by Kohonen [1], assigns each input $\boldsymbol{x}_i$ to its best matching unit (BMU),

$$k_i^* = \underset{k\in\{1,\dots,M\}}{\arg\min} \|\boldsymbol{x}_i - \boldsymbol{w}_k\|^2. \tag{1}$$

With the neighborhood kernel

$$\mathcal{K}_{jk} = \exp\left(-\frac{\|\boldsymbol{r}_j - \boldsymbol{r}_k\|^2}{2\sigma^2}\right), \tag{2}$$

where $\sigma > 0$ controls the neighborhood width, the reference vectors are updated as

$$\boldsymbol{w}_j \leftarrow \boldsymbol{w}_j + \eta\mathcal{K}_{jk_i^*}(\boldsymbol{x}_i - \boldsymbol{w}_j), \tag{3}$$

where $\eta \in (0,1)$ is the learning rate.

Batch SOM (BSOM) [2] replaces this sequential update with

$$\boldsymbol{w}_j = \frac{\sum_{i=1}^N \mathcal{K}_{jk_i^*}\boldsymbol{x}_i}{\sum_{i=1}^N \mathcal{K}_{jk_i^*}}, \tag{4}$$

which updates all reference vectors with per-iteration computational complexity $O(NM + M^2)$. Although this is often summarized as $O(NM)$ when $M \ll N$, we retain the M^2 term explicitly because we also consider large-M settings.

Both the sequential and batch forms are widely used for topographic mapping and often preserve neighborhood relations in practice, but they are defined algorithmically rather than as the exact minimization of a single explicit smooth optimization objective. In particular, because the BMU index changes discontinuously across Voronoi boundaries, the induced update field is not generally the gradient of a scalar potential [3]. This limitation motivated subsequent optimization-based formulations of SOM-type learning.

C. Objective-Based SOM Formulations and Their Limitations

Following earlier Bayesian and distortion-based views of SOM-type learning [4], Heskes [5] formulated topographic mapping by the convolved distortion

$$J_{\mathrm{CD}} = \sum_{i=1}^N \sum_{j=1}^M p_{ij}\left(\sum_{k=1}^M h_{jk}\|\boldsymbol{x}_i - \boldsymbol{w}_k\|^2\right), \tag{5}$$

where $p_{ij} \geq 0$, $\sum_{j=1}^M p_{ij} = 1$, and h_{jk} is typically given by the normalized Gaussian kernel

$$h_{jk} = \frac{\exp\left(-\frac{\|\boldsymbol{r}_j - \boldsymbol{r}_k\|^2}{2\sigma^2}\right)}{\sum_{l=1}^M \exp\left(-\frac{\|\boldsymbol{r}_j - \boldsymbol{r}_l\|^2}{2\sigma^2}\right)}. \tag{6}$$

This formulation incorporates neighborhood interaction directly into the objective, but it is linear in p_{ij} and thus degenerates into hard assignments. In addition, the direct evaluation of (5) incurs a per-iteration computational cost of $O(NM^2)$.

To obtain soft assignments, Graepel et al. [6] introduced Soft Topographic Vector Quantization (STVQ), which is based on the entropy-regularized objective

$$J_{\mathrm{STVQ}} = J_{\mathrm{CD}} + \lambda \sum_{i=1}^N \sum_{j=1}^M p_{ij} \ln p_{ij}, \tag{7}$$

where $\lambda > 0$. This yields soft assignments of softmax form, but the neighborhood-coupled structure remains. To reduce the cost, Heskes [7] applied a bias-variance decomposition:

$$\begin{aligned} J_{\mathrm{BVD}} = &\sum_{i=1}^N \sum_{j=1}^M p_{ij}\left(\|\boldsymbol{x}_i - \tilde{\boldsymbol{w}}_j\|^2 + V_j(\mathcal{W})\right) \\ &+ \lambda \sum_{i=1}^N \sum_{j=1}^M p_{ij} \ln p_{ij}, \end{aligned} \tag{8}$$

where

$$\tilde{\boldsymbol{w}}_j = \sum_{k=1}^M h_{jk}\boldsymbol{w}_k, \quad V_j(\mathcal{W}) = \sum_{k=1}^M h_{jk}\|\boldsymbol{w}_k - \tilde{\boldsymbol{w}}_j\|^2. \tag{9}$$

This reduces the per-iteration cost to $O(NM + M^2)$ under dense neighborhood interactions. Although this is more efficient than the original $O(NM^2)$ formulation, the M^2 term due to neighborhood coupling still remains, which can become a practical limitation when the number of nodes is large.

III. Proposed Method: SOM-OLP

In this section, we propose Self-Organizing Maps with Optimized Latent Positions (SOM-OLP), an optimization-based topographic mapping method that introduces a continuous latent position for each data point. We derive a separable surrogate local cost from a continuous extension of the STVQ neighborhood distortion and formulate the SOM-OLP objective based on it.

A. Continuous Latent Extension and Surrogate Local Cost

For a data point $\boldsymbol{x}_i \in \mathbb{R}^D$ and a latent node located at $\boldsymbol{r}_j \in \mathbb{R}^L$, the STVQ neighborhood distortion is given by

$$E_i(\boldsymbol{r}_j) = \sum_{k=1}^M h_{jk}\|\boldsymbol{x}_i - \boldsymbol{w}_k\|^2, \tag{10}$$

where $\boldsymbol{w}_k \in \mathbb{R}^D$ is the reference vector of node k, and h_{jk} is the normalized neighborhood function defined in (6). To extend this formulation to a continuous latent space, let

$\Omega \subset \mathbb{R}^L$ be a compact set containing $\mathcal{R}$, and define the continuous neighborhood function by

$$h(\boldsymbol{r}, \boldsymbol{r}_k) = \frac{\exp\left(-\frac{\|\boldsymbol{r}-\boldsymbol{r}_k\|^2}{2\sigma^2}\right)}{\sum_{l=1}^{M} \exp\left(-\frac{\|\boldsymbol{r}-\boldsymbol{r}_l\|^2}{2\sigma^2}\right)}. \tag{11}$$

The resulting continuous neighborhood distortion is

$$E_i(\boldsymbol{r}) \coloneqq \sum_{k=1}^{M} h(\boldsymbol{r}, \boldsymbol{r}_k) \|\boldsymbol{x}_i - \boldsymbol{w}_k\|^2. \tag{12}$$

Since E_i is continuous on the compact set Ω, it attains a minimum on Ω. Intuitively, smaller values of $E_i(\boldsymbol{r})$ indicate latent positions that are more compatible with $\boldsymbol{x}_i$. We therefore define the latent position of $\boldsymbol{x}_i$ by

$$\boldsymbol{v}_i \in \arg\min_{\boldsymbol{r} \in \Omega} E_i(\boldsymbol{r}). \tag{13}$$

We then approximate E_i locally around $\boldsymbol{v}_i$. Assuming that $\boldsymbol{v}_i$ is a local minimizer in the interior of Ω, the first-order term vanishes at $\boldsymbol{v}_i$, and a local quadratic approximation gives

$$E_i(\boldsymbol{r}_j) \approx E_i(\boldsymbol{v}_i) + \frac{1}{2}(\boldsymbol{v}_i - \boldsymbol{r}_j)^\top \mathbf{H}_i (\boldsymbol{v}_i - \boldsymbol{r}_j), \tag{14}$$

where $\mathbf{H}_i = \nabla^2 E_i(\boldsymbol{v}_i)$. Under a locally isotropic approximation $\mathbf{H}_i \approx 2\gamma \mathbf{I}$, with $\mathbf{I} \in \mathbb{R}^{L \times L}$ denoting the identity matrix and $\gamma \geq 0$, this becomes

$$E_i(\boldsymbol{r}_j) \approx E_i(\boldsymbol{v}_i) + \gamma \|\boldsymbol{v}_i - \boldsymbol{r}_j\|^2. \tag{15}$$

Equation (15) separates the local behavior of $E_i(\boldsymbol{r}_j)$ into a j-independent offset and a quadratic deviation term. In the no-neighborhood limit $h_{jk} \to \delta_{jk}$, (10) reduces to

$$E_i(\boldsymbol{r}_j) \to \|\boldsymbol{x}_i - \boldsymbol{w}_j\|^2, \tag{16}$$

which is the direct quantization error. Motivated by this limiting form together with the local quadratic approximation, we introduce the separable surrogate local cost

$$\ell_{ij} \coloneqq \|\boldsymbol{x}_i - \boldsymbol{w}_j\|^2 + \gamma \|\boldsymbol{v}_i - \boldsymbol{r}_j\|^2. \tag{17}$$

This surrogate preserves the direct data-space distortion term and introduces a latent-space proximity term.

B. Optimization Problem

Using (17), we formulate SOM-OLP as the following constrained optimization problem:

$$\begin{aligned} \underset{\mathbf{P}, \mathcal{V}, \mathcal{W}}{\text{minimize}} \quad & J_{\text{SOM-OLP}} \\ \text{subject to} \quad & p_{ij} \geq 0 \quad \forall i, \forall j, \\ & \sum_{j=1}^{M} p_{ij} = 1 \quad \forall i, \end{aligned} \tag{18}$$

where

$$\begin{aligned} J_{\text{SOM-OLP}} = & \sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij} \left(\|\boldsymbol{x}_i - \boldsymbol{w}_j\|^2 + \gamma \|\boldsymbol{v}_i - \boldsymbol{r}_j\|^2 \right) \\ & + \lambda \sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij} \ln p_{ij}. \end{aligned} \tag{19}$$

Here, $\mathcal{V} = \{\boldsymbol{v}_i\}_{i=1}^{N}$ denotes the set of latent positions. Equation (19) is an entropy-regularized objective based on the separable surrogate local cost in (17), and it can be evaluated in $O(NM)$ time per iteration.

The parameter $\lambda > 0$ controls the strength of entropy regularization and hence the softness of the assignment weights, while $\gamma \geq 0$ controls the balance between the data-space distortion and the latent-space proximity term. When $\gamma = 0$, SOM-OLP reduces to entropy-regularized fuzzy c-means [8]; when $\gamma = 0$ and $\lambda \to 0$, it further reduces to standard k-means [9].

C. Update Rules

We optimize the problem in (18) by block coordinate descent (BCD). Since $J_{\text{SOM-OLP}}$ is convex with respect to each block of variables, $\mathbf{P}$, $\mathcal{V}$, and $\mathcal{W}$, when the others are fixed, each update is obtained by minimizing $J_{\text{SOM-OLP}}$ with respect to the corresponding block.

1) Update of $\mathbf{P}$*:* With $\mathcal{V}$ and $\mathcal{W}$ fixed, minimizing $J_{\text{SOM-OLP}}$ by the method of Lagrange multipliers yields

$$p_{ij} = \frac{\exp\left(-\frac{1}{\lambda}\left(\|\boldsymbol{x}_i - \boldsymbol{w}_j\|^2 + \gamma \|\boldsymbol{v}_i - \boldsymbol{r}_j\|^2\right)\right)}{\sum_{k=1}^{M} \exp\left(-\frac{1}{\lambda}\left(\|\boldsymbol{x}_i - \boldsymbol{w}_k\|^2 + \gamma \|\boldsymbol{v}_i - \boldsymbol{r}_k\|^2\right)\right)}. \tag{20}$$

Thus, each data point is softly assigned by a softmax over a cost that combines data-space distortion and latent-space proximity.

2) Update of $\mathcal{V}$*:* With $\mathbf{P}$ and $\mathcal{W}$ fixed, the stationary condition $\nabla_{\boldsymbol{v}_i} J_{\text{SOM-OLP}} = \mathbf{0}$ yields

$$\boldsymbol{v}_i = \sum_{j=1}^{M} p_{ij} \boldsymbol{r}_j. \tag{21}$$

Hence, $\boldsymbol{v}_i$ is the weighted average of the node coordinates and provides a latent position for $\boldsymbol{x}_i$.

3) Update of $\mathcal{W}$*:* With $\mathbf{P}$ and $\mathcal{V}$ fixed, the stationary condition $\nabla_{\boldsymbol{w}_j} J_{\text{SOM-OLP}} = \mathbf{0}$ yields

$$\boldsymbol{w}_j = \frac{\sum_{i=1}^{N} p_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{N} p_{ij}}. \tag{22}$$

Thus, each reference vector is updated as the weighted centroid of the data points assigned to it.

D. Algorithm and Computational Complexity

Algorithm 1 summarizes the cyclic BCD procedure of SOM-OLP. We first initialize the reference vectors $\mathcal{W}^{(0)}$ by a principal component analysis (PCA)-based scheme [10], and then initialize the assignment probabilities $\mathbf{P}^{(0)}$ from $\mathcal{W}^{(0)}$.

Let $\tilde{\mathbf{X}}$ denote the centered version of $\mathbf{X}$, and let $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_K$ be the leading $K = \min\{L, D\}$ principal directions of $\tilde{\mathbf{X}}$. The node coordinates $\boldsymbol{r}_j$ are normalized along each axis to obtain $\tilde{\boldsymbol{r}}_j \in [-1, 1]^K$, and let $\boldsymbol{s} = (s_1, \ldots, s_K)$ denote the standard deviations of the projected data along these principal directions. The initial reference vector of node j is then set as

$$\boldsymbol{w}_j^{(0)} = \boldsymbol{\mu} + 2 \sum_{l=1}^{K} \tilde{r}_{jl} s_l \boldsymbol{u}_l, \tag{23}$$

Algorithm 1 SOM-OLP

Require: Data points $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, node coordinates $\mathcal{R} = \{\boldsymbol{r}_j\}_{j=1}^M$, hyperparameters γ, λ, and tolerance ε

Ensure: $\mathcal{W}, \mathcal{V}, \mathbf{P}$

Initialize $\mathcal{W}^{(0)}$ by PCA using (23)
Initialize $\mathbf{P}^{(0)}$ using (24)
repeat
  Update $\mathcal{V}$ using (21)
  Update $\mathcal{W}$ using (22)
  Update $\mathbf{P}$ using (20)
until the relative change in $J_{\text{SOM-OLP}}$ is at most ε
return $\mathcal{W}, \mathcal{V}, \mathbf{P}$

where $\boldsymbol{\mu}$ is the data mean. Since the latent positions $\mathcal{V}$ are not available at initialization, we initialize the assignment probabilities using only the data-space distortion term[1]:

$$p_{ij}^{(0)} = \frac{\exp\left(-\frac{1}{\lambda}\|\boldsymbol{x}_i - \boldsymbol{w}_j^{(0)}\|^2\right)}{\sum_{k=1}^M \exp\left(-\frac{1}{\lambda}\|\boldsymbol{x}_i - \boldsymbol{w}_k^{(0)}\|^2\right)}. \tag{24}$$

After initialization, the variables are updated cyclically in the order $\mathcal{V} \to \mathcal{W} \to \mathbf{P}$. Under this cyclic BCD scheme, the objective value is monotonically non-increasing. Convergence can be determined by the relative change in the objective value. Specifically, letting $J^{(t)}$ denote the objective value at iteration t, the algorithm is regarded as converged when

$$\frac{\left|J^{(t)} - J^{(t-1)}\right|}{\max\left\{1, \left|J^{(t-1)}\right|\right\}} \le \varepsilon, \tag{25}$$

where ε is a small positive constant, and the denominator prevents numerical instability when $\left|J^{(t-1)}\right|$ is close to zero.

Each update in Algorithm 1 has closed form, yielding $O(NM)$ per-iteration complexity for SOM-OLP, compared with $O(NM^2)$ for STVQ and $O(NM + M^2)$ for the BVD-based formulation under dense neighborhood interactions.

E. Interpretation of Topographic Consistency

Although $J_{\text{SOM-OLP}}$ contains no explicit node–node interaction term, substituting (21) and (22) into (19) yields

$$\begin{aligned}\tilde{J} &= \frac{1}{2}\sum_{j=1}^M\sum_{i=1}^N\sum_{l=1}^N \frac{p_{ij}p_{lj}}{n_j}\|\boldsymbol{x}_i - \boldsymbol{x}_l\|^2 \\ &\quad + \frac{\gamma}{2}\sum_{i=1}^N\sum_{j=1}^M\sum_{k=1}^M p_{ij}p_{ik}\|\boldsymbol{r}_j - \boldsymbol{r}_k\|^2 + \lambda\sum_{i=1}^N\sum_{j=1}^M p_{ij}\ln p_{ij} \\ &= \operatorname{tr}(\mathbf{X}^\top\mathbf{L}_{\text{data}}\mathbf{X}) + \gamma\operatorname{tr}(\mathbf{R}^\top\mathbf{L}_{\text{lat}}\mathbf{R}) + \lambda\sum_{i=1}^N\sum_{j=1}^M p_{ij}\ln p_{ij},\end{aligned} \tag{26}$$

[1]Random initialization of $\mathbf{P}^{(0)}$ also led to stable optimization in our preliminary experiments.

TABLE I
HYPERPARAMETER SEARCH SPACE FOR EACH METHOD.

Method	Hyperparameter	Search range	Search scale
BSOM	σ	$[10^{-3}, 10^3]$	log
STVQf	σ	$[10^{-3}, 10^3]$	log
	λ	$[10^{-3}, 10^3]$	log
GTM	m	$\{2, 3, \ldots, 15\}$	linear
	s	$[10^{-3}, 10^3]$	log
	λ_{reg}	$[10^{-3}, 10^3]$	log
SOM-OLP	γ	$[10^{-3}, 10^3]$	log
	λ	$[10^{-3}, 10^3]$	log

where

$$\mathbf{L}_{\text{data}} = \mathbf{I}_N - \mathbf{P}\mathbf{D}_n^{-1}\mathbf{P}^\top, \tag{27}$$

$$\mathbf{L}_{\text{lat}} = \mathbf{D}_n - \mathbf{P}^\top\mathbf{P}, \tag{28}$$

$$\mathbf{D}_n = \operatorname{diag}(n_1, \ldots, n_M), \tag{29}$$

$$n_j = \sum_{i=1}^N p_{ij}. \tag{30}$$

The first and second terms can be viewed as graph-Laplacian-type regularizers in the data space and latent space, respectively. For each node, the former favors assigning larger common weights to nearby data points than to distant ones. For each data point, the latter favors assigning larger weights to nearby latent nodes than to distant ones. Thus, the proposed objective encourages assignment structures consistent with neighborhood relations in both the data space and the latent space.

IV. NUMERICAL EXPERIMENTS

We compared Batch SOM (BSOM), Soft Topographic Vector Quantization (STVQ), its fast implementation based on bias–variance decomposition (denoted here by STVQf), Generative Topographic Mapping (GTM) [11], and the proposed SOM-OLP. GTM is a probabilistic generative model for topographic mapping; for this method, we used the `eGTM` class from the Python package `ugtm` (v2.0) [12]. All methods were initialized using PCA to eliminate variability due to random initialization. For all methods, the maximum number of iterations was set to 1000 and the convergence tolerance to 10^{-4}. Unless otherwise noted, we used a two-dimensional square latent grid with $M = 16^2$ nodes on $[-1, 1] \times [-1, 1]$.

As evaluation metrics, we used Trustworthiness (TW), Continuity (CN), and Quantization Error (QE). TW and CN assess two complementary aspects of neighborhood preservation between the data space and the latent space: TW measures the extent to which neighbors in the latent space are also neighbors in the original data space, whereas CN measures the extent to which neighbors in the original data space remain neighbors in the latent space [13], [14]. In contrast, QE measures how accurately the data points are represented by the learned reference vectors in the data space [15]. For TW and CN, the number of neighbors was set to 5.

For hyperparameter tuning, we used Optuna [16] with the multivariate Tree-structured Parzen Estimator (TPE) sampler.

TABLE II
QUANTITATIVE COMPARISON ON THE SADDLE DATASET. **BOLD**: BEST; UNDERLINED: SECOND BEST.

Method	TW ↑	CN ↑	QE ↓	Iters ↓	Time ↓ [ms/iteration]
BSOM	0.9945	0.9952	0.0089	**17**	**0.40**
STVQf	0.9945	0.9952	**0.0057**	34	4.09
GTM	0.9979	0.9986	0.0087	105	16.94
SOM-OLP	**0.9986**	**0.9992**	0.0102	110	3.86

TABLE III
QUANTITATIVE COMPARISON ON THE DIGITS DATASET. **BOLD**: BEST; UNDERLINED: SECOND BEST.

Method	TW ↑	CN ↑	QE ↓	Iters ↓	Time ↓ [ms/iteration]
BSOM	0.9871	0.9657	1.2217	**21**	**11.45**
STVQf	0.9872	0.9685	0.9315	33	25.96
GTM	**0.9910**	0.9746	1.1518	89	58.85
SOM-OLP	0.9864	**0.9761**	**0.8895**	37	29.77

For each method and dataset, one Optuna study consisted of 100 trials maximizing $(\mathrm{TW}+\mathrm{CN})/2$. For the saddle-manifold and Digits case studies, we ran five Optuna studies with different sampler seeds for each method and selected the final model with the smallest QE among the five resulting models. The hyperparameter search ranges are listed in Table I. All experiments were conducted on a Windows machine with an Intel Core i7-14700 CPU and 31.7 GB RAM.

A. Saddle Manifold

The saddle-manifold data were generated by independently sampling x_1 and x_2 from the uniform distribution on $[-1, 1]$, and then defining $x_3 = x_1^2 - x_2^2 + \xi$, where ξ is Gaussian noise drawn from the normal distribution with mean 0 and variance 0.1^2. The number of data points was set to $N = 500$. In this experiment, the generated three-dimensional data were used without additional normalization. For visualization in both the data space and the latent space, the data points were colored consistently according to their x_3 values so that the correspondence between the two spaces could be examined.

For the saddle-manifold dataset, the selected hyperparameters were $\sigma = 0.1184$ for BSOM, $\sigma = 0.0637$ and $\lambda = 8.70 \times 10^{-3}$ for STVQf, $m = 14$, $s = 271.4$, and $\lambda_{\text{reg}} = 1.35 \times 10^{-3}$ for GTM, and $\gamma = 73.79$ and $\lambda = 1.696$ for SOM-OLP.

The quantitative results are reported in Table II. SOM-OLP achieved the best TW and CN values, whereas STVQf attained the smallest QE, and GTM showed the second-best performance in TW, CN, and QE. In terms of computational cost, BSOM required the fewest iterations and the smallest time per iteration. SOM-OLP required more iterations to converge, but its per-iteration time was smaller than those of STVQf and GTM under this experimental setting. These results suggest that SOM-OLP provides competitive topology-preserving performance on the saddle dataset while maintaining moderate computational cost among the compared objective-based methods.

Fig. 1 shows that all four methods capture the overall structure of the saddle manifold in the data space. Since the selected models were obtained under a tuning criterion based on $(\mathrm{TW} + \mathrm{CN})/2$, and qualitative properties may vary with the hyperparameter setting, we do not overinterpret the visual differences. By construction, BSOM and STVQf provide discrete node-based latent representations, whereas GTM and SOM-OLP provide latent positions. Under the present setting, GTM appears to reflect the latent-node arrangement more strongly, whereas SOM-OLP yields a more flexible continuous embedding. Thus, the main qualitative observation is that SOM-OLP combines latent positions with competitive topology-preserving performance.

B. Digits Dataset

We evaluated the performance of each method using the Digits dataset from scikit-learn [17]. This dataset comprises $N = 1,797$ grayscale images of handwritten digits (8×8 pixels), resulting in a $D = 64$ dimensional data space. Each pixel value is an integer in the range $[0, 16]$. We normalized the input as $\mathbf{X} = \mathbf{X}_{\text{raw}}/16$ to scale the components within $[0, 1]$. For visualization, data points are colored according to their ground-truth labels $(0, \ldots, 9)$.

For the Digits dataset, the selected hyperparameters were $\sigma = 0.0812$ for BSOM, $\sigma = 0.0556$ and $\lambda = 0.154$ for STVQf, $m = 15$, $s = 2.385$, and $\lambda_{\text{reg}} = 0.558$ for GTM, and $\gamma = 0.815$ and $\lambda = 0.331$ for SOM-OLP.

Table III summarizes the quantitative metrics. GTM achieved the highest TW, whereas SOM-OLP outperformed the other methods in CN and QE, with STVQf also showing competitive performance. In terms of computational cost, BSOM was the most efficient in both iterations and time per iteration. Although SOM-OLP required more computation than BSOM and STVQf, it achieved the best CN and QE values while requiring approximately half the time per iteration of GTM. These results suggest that SOM-OLP provides competitive performance on the Digits dataset, with a favorable balance between mapping quality and computational cost.

Fig. 2 highlights structural differences in the learned mappings. BSOM and STVQf yield discrete representations in which data points are assigned to grid nodes, whereas GTM and SOM-OLP provide continuous latent positions. Under the present experimental conditions, GTM appears to produce a relatively grid-constrained distribution, although this tendency may depend on the hyperparameter setting, whereas SOM-OLP yields a more flexible representation that adapts to the local density of the digit clusters. Since the selected models were obtained under a tuning criterion based on $(\mathrm{TW}+\mathrm{CN})/2$, we do not overinterpret these visual differences. Rather, the main observation is that SOM-OLP provides flexible latent positions while maintaining competitive neighborhood-preserving performance, consistent with its favorable QE result.

1) Scalability with Respect to the Number of Nodes: To evaluate scalability with respect to the number of nodes, we measured the per-iteration wall-clock time on the Digits

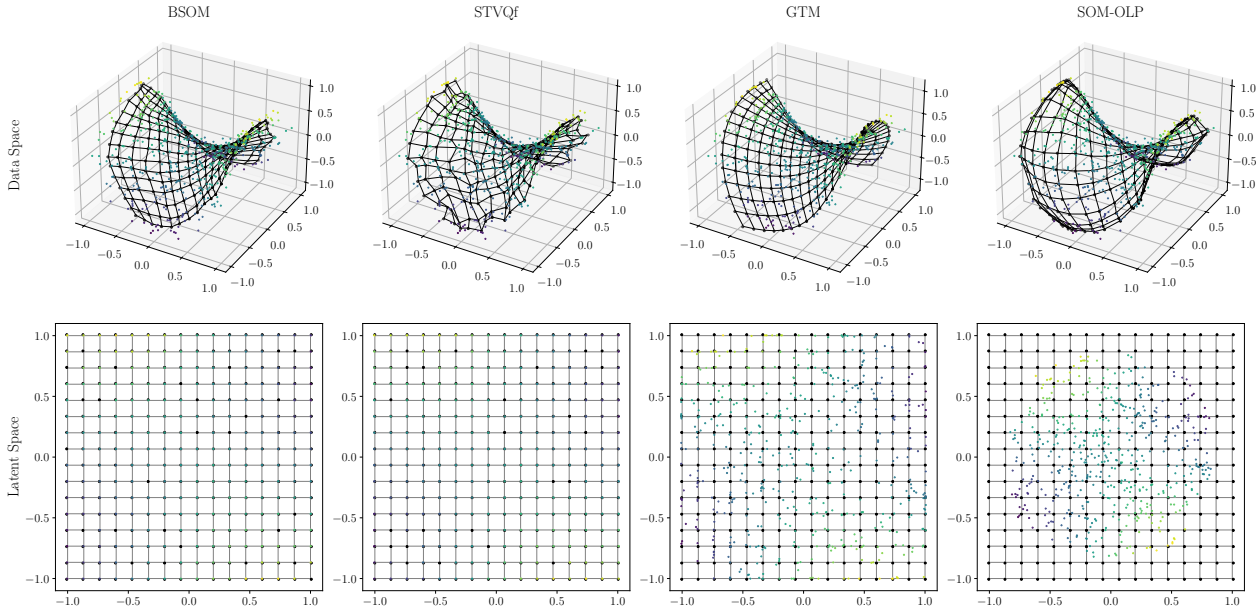

Fig. 1. Comparison of BSOM, STVQf, GTM, and SOM-OLP on the saddle dataset. For each method, the data-space view shows the learned reference vectors, while the latent-space view shows the corresponding representation of the data points. The same color coding based on the original x_3 values is used in both spaces to facilitate visual comparison of neighborhood relationships across the two spaces.

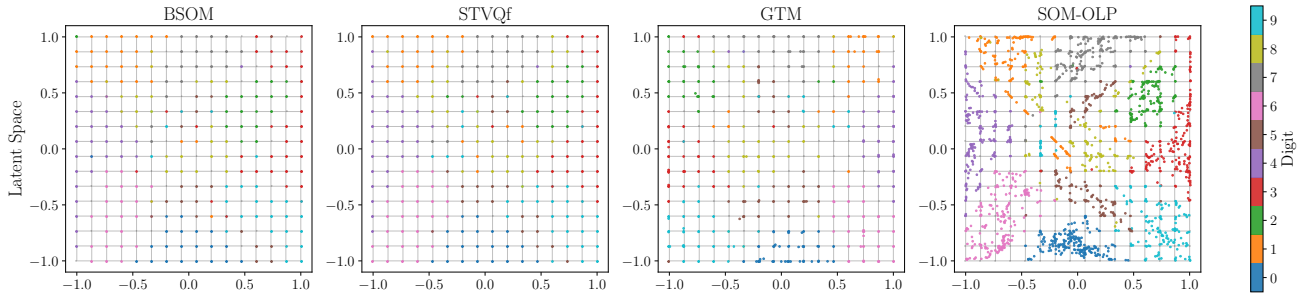

Fig. 2. Latent representations of the Digits dataset. While BSOM and STVQf are constrained to discrete nodes, GTM and SOM-OLP provide continuous latent positions. Notably, SOM-OLP exhibits a more adaptive distribution that reflects the local density of the data manifold.

TABLE IV
PER-ITERATION WALL-CLOCK TIME (MS) VERSUS GRID SIZE ON THE DIGITS DATASET ($T = 20$ ITERATIONS). OOM: OUT-OF-MEMORY ERROR.

M	BSOM	STVQf	GTM	SOM-OLP
$2\,500 = 50^2$	**81**	263	288	265
$10\,000 = 100^2$	**359**	1 173	1 143	1 064
$22\,500 = 150^2$	**975**	3 247	3 324	2 430
$40\,000 = 200^2$	**4 018**	10 356	8 268	4 641
$62\,500 = 250^2$	22 030	157 937	55 339	**8 617**
$90\,000 = 300^2$	OOM	OOM	OOM	**12 988**
$122\,500 = 350^2$	OOM	OOM	OOM	**17 509**
$160\,000 = 400^2$	OOM	OOM	OOM	**23 259**
$202\,500 = 450^2$	OOM	OOM	OOM	**29 897**
$250\,000 = 500^2$	OOM	OOM	OOM	**36 768**

dataset while increasing the latent grid size from $M = 50^2$ to 500^2. For each method, the hyperparameters were fixed to those selected in the preceding Digits experiment. For each grid size, the runtime was averaged over $T = 20$ iterations. The results are summarized in Table IV.

For relatively small to moderate grids ($M \leq 40{,}000$), BSOM is the fastest method, and SOM-OLP is consistently the second fastest. As the grid size increases further, however, the relative advantage of SOM-OLP becomes more apparent. At $M = 62{,}500$, SOM-OLP achieves the smallest per-iteration time among all compared methods. Moreover, for $M \geq 90{,}000$, BSOM, STVQf, and GTM all run out of memory in our experimental environment, whereas SOM-OLP remains executable up to $M = 250{,}000$.

These empirical results are consistent with the computational structure of the methods. BSOM and STVQf involve neighborhood-dependent computations, and GTM also becomes increasingly costly as the number of latent nodes grows in the present experiment. In contrast, SOM-OLP is based on a separable local cost and avoids explicit node–node convolutions. As a result, SOM-OLP scales more favorably as M increases, although BSOM remains more efficient at smaller node counts.

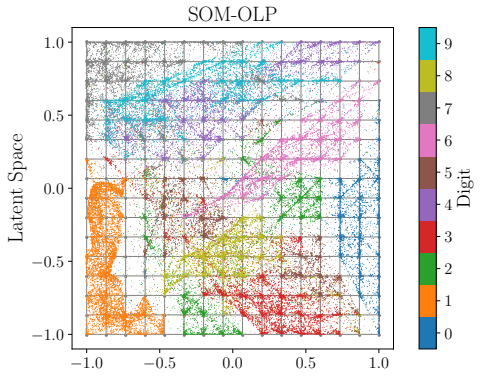

Fig. 3. Latent representation of the MNIST dataset obtained by SOM-OLP with a 16×16 grid. Data points are colored by their ground-truth digit labels.

C. MNIST Dataset

To further examine the scalability of SOM-OLP, we applied it to the MNIST handwritten digit dataset [18], which consists of $N = 70{,}000$ grayscale images (28×28 pixels) in a $D = 784$ dimensional space with 10 classes. Pixel values were normalized to $[0, 1]$ by dividing by 255. The hyperparameters were fixed to those selected in the $N = 2{,}000$ subsampled experiment ($\gamma = 2.767$, $\lambda = 2.439$), and SOM-OLP was trained on the full dataset and a 16×16 grid ($M = 256$). Training converged in 36 iterations, with approximately 5.9 s per iteration. Fig. 3 shows the resulting latent representation. Despite the large scale and high dimensionality of the dataset, SOM-OLP organizes the digit classes into broadly separated regions in the latent space. At the same time, noticeable overlap remains among some classes, suggesting that a fixed two-dimensional latent grid cannot fully disentangle all class relationships in MNIST. These results indicate that SOM-OLP scales to large high-dimensional datasets while still providing a meaningful topographic organization.

D. Real-world Datasets

To further evaluate generality, we compared the methods on 16 benchmark datasets, summarized in Table V. As a simple linear baseline, we also included PCA with a two-dimensional projection. Although PCA does not explicitly preserve topology, it provides a standard reference for assessing whether topology-aware nonlinear methods offer practical advantages over a widely used linear dimensionality-reduction method. The datasets were taken from scikit-learn [17] and the UCI Machine Learning Repository [19]. Missing values, when present, were imputed by the feature-wise median, and all features were standardized to zero mean and unit variance.

Table V summarizes the results on the 16 datasets. SOM-OLP achieved the best score on 9 datasets and the second-best score on 5 datasets, indicating the most consistently strong performance overall. GTM attained the best score on 6 datasets, whereas STVQf was best on Ecoli, and BSOM did not attain the best score on any dataset. PCA was consistently inferior to the nonlinear methods, which confirms the benefit of topology-aware nonlinear mapping.

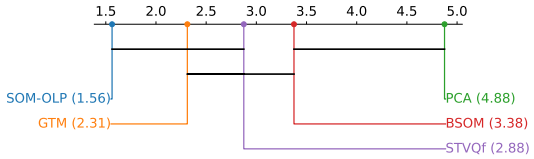

Fig. 4. Critical-difference diagram based on the average ranks of (TW + CN)/2 over the 16 benchmark datasets.

The average-rank analysis led to the same overall conclusion. SOM-OLP achieved the best average rank (1.56), followed by GTM (2.31), STVQf (2.88), BSOM (3.38), and PCA (4.88). The Friedman test rejected the null hypothesis that all methods perform equivalently ($\chi^2 = 39.75$, $p = 4.88 \times 10^{-8}$). In the critical-difference diagram shown in Fig. 4, methods connected by a horizontal bar do not differ significantly at the 0.05 level. The Nemenyi post-hoc analysis showed that SOM-OLP significantly outperformed BSOM, whereas its differences from STVQf and GTM were not significant. PCA performed significantly worse than STVQf, GTM, and SOM-OLP.

V. Discussion

Among related methods, GTM [11] and Parametrized SOM (PSOM) [20] are closely related in that they also relax the strictly discrete node-based representation of classical SOM. However, their formulations differ from that of SOM-OLP. GTM provides continuous latent representations through a probabilistic generative model, whereas PSOM achieves continuity by interpolating a node-supported manifold. In contrast, SOM-OLP introduces a latent position for each data point directly into the objective function and optimizes it jointly with the assignment weights and reference vectors. This yields a simpler reference-vector-based formulation and clarifies its relation to entropy-regularized fuzzy c-means and k-means.

Other recent directions, such as SOM-VAE [21], SatSOM [22], and Topological Autoencoders [23], address different goals. SOM-VAE focuses on deep discrete representation learning, with a particular emphasis on time series; SatSOM addresses plasticity control in continual learning; and Topological Autoencoders preserve topological structures of the input space through persistent-homology-based regularization. By contrast, SOM-OLP targets batch topographic mapping in a shallow reference-vector-based framework, with local topographic consistency induced by the objective itself.

Overall, SOM-OLP can be positioned as an objective-based topographic mapping method with continuous latent positions, closed-form updates, monotonic non-increase under cyclic BCD, and $O(NM)$ per-iteration complexity without explicit $O(M^2)$ node–node interactions. Moreover, it includes entropy-regularized fuzzy c-means as a special case and further reduces to k-means.

VI. Conclusion

This paper presented SOM-OLP, an objective-based topographic mapping method that introduces a continuous latent

TABLE V
$(\mathrm{TW}+\mathrm{CN})/2$ ON BENCHMARK DATASETS ($M = 16 \times 16$, STANDARDIZED). **BOLD**: BEST; UNDERLINED: SECOND BEST.

Dataset	N	D	PCA	BSOM	STVQf	GTM	SOM-OLP
Iris	150	4	0.9820	0.9799	0.9838	0.9787	**0.9869**
Wine	178	13	0.9041	0.9503	0.9531	**0.9595**	0.9593
Sonar	208	60	0.8754	0.9277	0.9391	**0.9395**	0.9344
Seeds	210	7	0.9615	0.9730	0.9751	0.9700	**0.9806**
Glass	214	9	0.8884	0.9527	0.9490	**0.9574**	0.9497
Heart Disease	303	13	0.8348	0.9222	0.9265	0.9317	**0.9350**
Ecoli	336	7	0.9284	0.9663	**0.9694**	0.9621	0.9671
Ionosphere	351	34	0.8866	0.9143	0.9078	0.9143	**0.9151**
Dermatology	366	34	0.8982	0.9452	0.9556	**0.9616**	0.9607
Breast Cancer (WDBC)	569	30	0.9137	0.9434	0.9451	0.9428	**0.9517**
Banknote Authentication	1372	4	0.9835	0.9939	0.9928	0.9948	**0.9963**
Yeast	1484	8	0.8791	0.9560	0.9560	0.9436	**0.9600**
Digits	1797	64	0.8846	0.9717	0.9747	**0.9768**	0.9766
Image Segmentation	2310	19	0.9167	0.9867	0.9852	**0.9877**	0.9870
Rice (Cammeo and Osmancik)	3810	7	0.9610	0.9876	0.9883	0.9891	**0.9911**
Dry Bean	13611	16	0.9637	0.9830	0.9824	0.9831	**0.9865**

position for each data point. Starting from the neighborhood distortion of STVQ, we constructed a separable surrogate local cost and derived an entropy-regularized objective with closed-form cyclic updates for assignment probabilities, latent positions, and reference vectors. As a result, SOM-OLP retains linear per-iteration complexity while avoiding the explicit node–node coupling that appears in previous objective-based SOM formulations. Experiments on a synthetic saddle manifold, scalability studies on the Digits and MNIST datasets, and 16 benchmark datasets showed that SOM-OLP achieves strong neighborhood preservation and quantization performance, favorable scalability for large numbers of latent nodes and large datasets, and the best average rank among the compared methods on the benchmark datasets. Future work includes broader comparisons with manifold learning and clustering methods, further analysis of the surrogate formulation and its theoretical properties, and more systematic strategies for hyperparameter selection.

ACKNOWLEDGMENT

The authors used ChatGPT (OpenAI) to improve the wording and readability of portions of this manuscript.

REFERENCES

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

[2] T. Kohonen, "The self-organizing map," Proceedings of the IEEE, vol. 78, no. 9, pp. 1464–1480, 1990.

[3] E. Erwin, K. Obermayer, and K. Schulten, "Self-organizing maps: ordering, convergence properties and energy functions," Biological Cybernetics, vol. 67, no. 1, pp. 47–55, 1992.

[4] S. P. Luttrell, "A Bayesian analysis of self-organizing maps," Neural Computation, vol. 6, no. 5, pp. 767–794, 1994.

[5] T. Heskes, "Energy functions for self-organizing maps," in Kohonen Maps, Elsevier, 1999, pp. 303–315.

[6] T. Graepel, M. Burger, and K. Obermayer, "Phase transitions in stochastic self-organizing maps," Physical Review E, vol. 56, pp. 3876–3890, 1997.

[7] T. Heskes, "Self-organizing maps, vector quantization, and mixture modeling," IEEE Transactions on Neural Networks, vol. 12, no. 6, pp. 1299–1305, 2001.

[8] S. Miyamoto and M. Mukaidono, "Fuzzy c-means as a regularization and maximum entropy approach," in Proc. 7th Int. Fuzzy Systems Association World Congress (IFSA'97), vol. 2, Prague, Czech Republic, June 1997, pp. 86–92.

[9] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in Proc. 5th Berkeley Symp. Math. Statist. Probab., vol. 1, L. M. Le Cam and J. Neyman, Eds. Berkeley, CA: University of California Press, 1967, pp. 281–297.

[10] I. T. Jolliffe, Principal Component Analysis, 2nd ed. New York: Springer, 2002.

[11] C. M. Bishop, M. Svensén, and C. K. I. Williams, "GTM: The generative topographic mapping," Neural Computation, vol. 10, no. 1, pp. 215–234, 1998.

[12] H. A. Gaspar, "ugtm: A Python package for data modeling and visualization using generative topographic mapping," Journal of Open Research Software, vol. 6, no. 1, p. 26, 2018.

[13] J. Venna and S. Kaski, "Neighborhood preservation in nonlinear projection methods: An experimental study," in Artificial Neural Networks (ICANN 2001), LNCS 2130. Berlin, Heidelberg: Springer, 2001, pp. 485–491.

[14] J. A. Lee and M. Verleysen, "Quality assessment of dimensionality reduction: Rank-based criteria," Neurocomputing, vol. 72, no. 7–9, pp. 1431–1443, 2009.

[15] G. Pölzlbauer, "Survey and comparison of quality measures for self-organizing maps," in Proc. 5th Workshop on Data Analysis (WDA'04), 2004, pp. 67–82.

[16] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining (KDD), 2019, pp. 2623–2631.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] M. Kelly, R. Longjohn, and K. Nottingham, "The UCI Machine Learning Repository," [Online]. Available: https://archive.ics.uci.edu.

[20] J. Walter and H. Ritter, "Rapid learning with parametrized self-organizing maps," Neurocomputing, vol. 12, pp. 131–153, 1996.

[21] V. Fortuin, M. Hüser, F. Locatello, H. Strathmann, and G. Rätsch, "SOM-VAE: Interpretable discrete representation learning on time series," in Proc. 7th Int. Conf. on Learning Representations (ICLR), 2019.

[22] I. Urbanik and P. Gajewski, "SatSOM: Saturation self-organizing maps for continual learning," 2025, arXiv:2506.10680v5. [Online]. Available: https://arxiv.org/abs/2506.10680.

[23] M. Moor, M. Horn, B. Rieck, and K. Borgwardt, "Topological autoencoders," in Proc. 37th Int. Conf. on Machine Learning (ICML), vol. 119. PMLR, 2020, pp. 7045–7054.