ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

UDC 004.93 DOI: https://doi.org/10.30837/2522-9818.2026.2.070

Mykyta Lapin, Yurii Parzhyn, Kostiantyn Bokhan, Kyrylo Perevoznyk, Tetiana Aleksandrova

INVARIANT STRUCTURAL LEARNING: CONCEPT FORMATION AS HYPERGRAPH ATTRACTOR DYNAMICS

Subject of study. The subject of study is the formation of object-class concepts in machine-learning systems as a process of structural and parametric reduction of hypergraph representations, rather than as the optimization of a loss function. ***Objective.*** To construct an alternative learning framework for artificial neural networks – without an error functional (back-propagation) – that realizes invariant structural learning, where a class concept is defined as a structural attractor (the fixed point of a monotone reduction operator on a partially ordered space of hypergraphs), and to validate this theory on the recognition of handwritten digits. **Objectives.** 1) Formalize the framework with axioms of segmentation stability, strict reductivity, positive-only training, and locality of attention; 2) prove convergence and uniqueness of the attractor; 3) establish order-invariance of learning; 4) decompose the attractor into structural and parametric levels; 5) evaluate the transfer on a complete-contour subset of MNIST. **Methods.** Hypergraphs are obtained by skeletonizing binary contours (Growing Neural Gas + Ramer–Douglas–Peucker); concept attractors form via graph reduction over critical-point anchors. Classification uses a graph-edit-distance comparator with property costs and a complexity-adjusted log prior. The training set consists of 76 hand-picked MNIST originals, augmented (rotation ±10°, shift ±10%) to 805 instances. **Results.** The transfer learns 13 concept-attractors with node counts ranging from 3 to 15. Of 8,707 admissible complete-contour MNIST images, 8,685 yielded valid skeletal graphs; on this valid set, the transfer achieves an accuracy of 85.80%, weighted precision of 89.31%, recall of 85.80%, and an F1-score of 86.66%. The error structure is interpretable per concept – every misclassification traces back to a named attractor and feature range. **Conclusions.** The reported performance, achieved without a loss function and using three to nine originals per concept-attractor, supports the claim that learning consists of constructing a structural attractor rather than minimizing an error functional. The framework offers a principled approach to interpretable few-shot classification and a constructive alternative to gradient-based learning.

Keywords: invariant structural learning; structural attractor; concept formation; graph edit distance; few-shot recognition; explainable AI; MNIST; hypergraph reduction.

1. Introduction

Modern machine learning methods, and neural network models in particular, rely heavily on the optimization of loss functions and statistical generalization. These approaches involve an external quality criterion, and the learning process is formalized as the minimization of the discrepancy between the prediction and a given target value. Despite the empirical success of statistical learning, this paradigm has fundamental limitations: dependence on the choice of loss function, sensitivity to data distribution, and the need for large training samples.

The relevance of this issue is heightened by three concurrent factors.

First, the EU Regulation on Artificial Intelligence (Regulation 2024/1689) [1], together with Article 22 of the GDPR [2], establishes a legally binding requirement: automated decisions in regulated areas must be accompanied by an explanation that corresponds to the model's internal logic – a standard that post-hoc surrogates fail to meet in practice.

Second, the energy and computational costs of training large statistical models are growing faster than the volume of available labeled datasets, creating economic pressure in favor of methods capable of learning from a handful of examples.

Third, industrial areas with a chronic shortage of labeled data – medical OCR, defect detection in contour structures, and low-volume document processing – remain underserved by gradient-based classifiers. These three factors motivate the search for a non-statistical alternative, developed below.

An alternative approach involves viewing learning as a process of endogenous structural self-organization, in which the system does not minimize external error but instead converges to an internally consistent state. This paper proposes a formalization of this view, which extends the "information architecture" approach [3] based on hypergraph representations and dynamic structural reduction; this approach is called Invariant-Structural Learning.

The central idea is that each observed object is represented by a hypergraph, and learning involves

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

the sequential accumulation and alignment of such hypergraphs, followed by reduction. The alignment procedure maps elements from different hypergraphs into a common structural space, forming a cumulative hypergraph in which one can analyze the frequencies of element occurrence and their stability under the effect of reduction.

The key concept of the proposed theory is the structural attractor: an invariant structure toward which the reduction process converges. Unlike classical attractors of dynamical systems, this object is defined not through trajectories in continuous space, but through the intersection of coordinated structural invariants. It is shown below that the attractor admits an equivalent characterization as the set of elements whose frequency in the cumulative carrier is equal to one – that is, those that are always present – and this combines a dynamic interpretation with a combinatorial one.

The proposed approach differs from traditional learning methods in four fundamental ways:

(i) there is no explicit error function and no procedure for minimizing it;

(ii) learning is interpreted as the reduction and alignment of structures;

(iii) invariants are formed endogenously, without external templates;

(iv) invariance with respect to the order of example presentation is achieved.

Mathematically, the model relies on discrete structures (hypergraphs), partially ordered sets, and monotonic reduction operators that converge to fixed points.

The main contributions of this work are as follows.

1. A formal hypergraph model of learning via structural reduction is proposed.

2. The concept of a structural attractor is introduced, and it is shown that it coincides with the intersection of class invariants and the single frequency condition.

3. Theorems regarding the convergence of the reduction process and the uniqueness of the attractor are proven.

4. The invariance of the learning result with respect to the order of example presentation is established.

5. It is shown that the attractor naturally decomposes into topological (structural) and parametric levels.

Overall, the work proposes an alternative theoretical framework for machine learning, in which structural invariants and attractor dynamics play a central role, rather than error optimization. This opens the way to constructing models focused on stability, interpretability, and learning with a limited number of examples.

Empirical validation of the proposed framework on a subset of the MNIST dataset containing closed contours is presented in Chapter 5. The concept-attractor classifier based on graph editing distance trains 13 attractors from 76 selected originals – ranging from three to nine per concept – expanded by parametric augmentation (random rotation within ±10° and translation within ±10% of the canvas) to 805 training instances. Of the 8,707 valid images, 8,685 formed valid skeleton graphs; on this valid set, the system achieves 85.80% accuracy, 89.31% weighted precision, 85.80% recall, and 86.66% F1-score, demonstrating the low-variance behavior predicted by the "structure–parameter" decomposition theorems in Section 4.1 and the order-invariance consequence from Theorem 2.

2. Literature Review and Problem Statement

Traditional approaches to handwritten digit recognition on MNIST [4] establish a clear benchmark against which any new method should be evaluated. Convolutional neural networks – from the canonical LeNet-5 architecture [4] to modern residual variants – consistently exceed 99% test accuracy on the full training set of 60,000 images. The Gaussian kernel support vector machine method achieves approximately 98.6% on the standardized MNIST; on raw pixels without augmentation, the result is more modest. Multilayer perceptrons with moderately wide hidden layers achieve 97–98% when trained with deslanting and affine augmentation [4]. A common feature of these methods is the need for tens of thousands of labeled examples and the absence of any built-in explanation mechanism: the decision of a convolutional network is based on linear combinations of feature maps, the interpretation of which by a human is possible only through post-hoc visualization surrogates.

A separate line of research focuses on the few-shot setting, where only a small number of labeled examples per class are available. Prototypical Networks [5] construct a class prototype as the average embedding of support examples in a latent space, trained via episodic optimization. Matching Networks [6] use an attention mechanism to weight the contribution of each support example, while Model-Agnostic Meta-Learning [7] seeks a parameter initialization from which the model adapts to a new task in a few gradient steps.

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

These methods are typically evaluated on the episodic protocol on Omniglot and miniImageNet, rather than on the open MNIST protocol; therefore, a direct numerical comparison with the 85.80% accuracy presented here is best interpreted as a conceptual positioning within the niche of explainable few-shot classifiers, rather than as a direct benchmark. A common limitation of this line of research is that, despite reducing the need for labeled data during adaptation, these methods do not eliminate the dependence on a large base corpus during pre-training, and the prototype in Prototypical Networks remains an opaque vector in a multidimensional space rather than an interpretable object suitable for expert verification.

The dominant approach to achieving interpretability for pre-trained black-box models has been the construction of post-hoc surrogates. Minh et al. [8], Hooshyar and Yang [9], Rajabi and Etminani [10], and Nawaz et al. [11] survey this landscape and classify methods by locality, model specificity, and type of approximation. The most common examples are LIME [12], which locally approximates a nonlinear model with a linear surrogate, and SHAP [13], based on Shapley value attribution. Slack et al. [14] experimentally demonstrated that LIME and SHAP are vulnerable to targeted attacks: a classifier with distinctly discriminatory behavior can be made to appear neutral under both methods. Post-hoc explanations therefore offer no guarantee of correspondence to the model's actual logic, which motivates interest in models that are explainable by design – that is, models in which the internal representation itself is an interpretable description.

Graphs as image representations have a long history in computer vision: skeleton graphs preserve the topology and basic geometry of an object while discarding redundant pixels, and demonstrate the competitiveness of graph representations in pattern recognition tasks. The current trend in graph neural networks, particularly Graph Prototypical Networks [15], extends few-shot learning to attributed graphs, but reverts to latent-vector representations and does not preserve a directly inspectable representation of the concept. In a previous study by the same team [16], the line of interpretable graph attractors was tested on a restricted six-class MNIST dataset (digits 1, 2, 3, 6, 7, 9; 5,467 test images, 8 attractors) and achieved 82.35% accuracy with exact calculation of graph editing distance under a 60-second comparison budget. This work extends the methodology to the full ten-class MNIST alphabet with contour integrity filtering, expands the alphabet to 13 attractors, introduces diagnostic feature weighting based on the width of the reduced range, and transitions to an iterative upper bound for graph editing with progressive refinement under a significantly smaller time budget – a combination that yields the reported 85.80% accuracy from Section 5.

The matching of attributed graphs formally boils down to finding the minimum sequence of editing operations (insertion, deletion, and replacement of vertices and edges) that transforms one graph into another. Sanfeliu and Fu [17] introduced the concept of edit distance for attributed relational graphs and established it as a similarity metric for pattern recognition. Conte et al. [18] summarized three decades of graph matching methods and highlighted the trade-off between exact and approximate computation; the exact graph edit distance remains NP-hard. Riesen and Bunke [19] proposed an efficient binary approximation with assignment, yielding a suboptimal upper bound in polynomial time, enabling the use of the metric on graphs of practically useful sizes at the cost of optimality. More recent neural variants replace explicit edit costs with learned embeddings: Piao et al. [20] compute graph edit distance via neural graph alignment, predicting the approximate distance from training pairs; Moscatelli et al. [21] formulate the basic node matching problem with an explicit emphasis on metric properties and assignment interpretability; Xie et al. [22] extend the paradigm to federated few-shot graph classification. An additional direction within the same class of problems is the reconstruction of incomplete spatial graphs using graph neural networks; an example is the recent work [23] on completing 3D point cloud models in the same journal. These methods significantly improve computational efficiency and scalability; however, most replace interpretable elementary editing costs with latent embeddings, reducing the readability of decisions at the level of "which vertex was matched with which and with what penalty". This leaves room for architectures in which the operations themselves remain interpretable by design – this is the motivation behind the proposed approach.

The proposed framework is based on four well-established lines of evidence from cognitive science. The binding problem – how the brain combines distributed representations of attributes into coherent perceptual wholes – was formalized by von der Malsburg [24] and remains a central problem in neural coding, with extensions in the form of neural

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

syntax [25]; the structural attractor model solves it directly by encoding attributive relations as anchor links (hyperedges) within a single representation. The literature on active perception, originating with Yarbus [25] and developed by Bajcsy et al. [26] and Rucci and Victor [27], shows that the trajectory of a saccade is determined by the cognitive task no less than by the image itself, which motivates the selection of anchors based on attention rather than exhaustive global alignment. Single-unit recordings [28, 29] have revealed concept-cell responses invariant to surface transformations of the stimulus, empirically supporting the existence of class-level invariant representations postulated in Definition 3 (concept). The dendritic computation hypothesis [30, 31, 32, 33, 34] asserts that a neuron's dendritic tree performs substantive logical computation rather than acting as a passive integrator, providing a biological substrate in which the algebra of structural reduction can be physically realized. These four lines of reasoning together justify the non-statistical orientation of the proposed framework and motivate the architectural decisions in Section 4.

To summarize this review, three desirable properties stand out, none of which currently have a satisfactory joint solution:

a) training without backpropagation and without a large labeled corpus;

b) operation with a small number of examples per class;

c) local explainability of each decision at the model level.

Classical convolutional and support vector models provide only accuracy; few-shot meta-learning methods provide adaptation with a small training set but remain opaque; post-hoc methods of explainable AI provide an approximation of behavior rather than the logic itself. The graph attractor architecture proposed in this work, trained without backpropagation, formally satisfies all three properties simultaneously – and this defines the unsolved part of the problem and justifies the relevance of this research.

3. Purpose and Objectives of the Study

The aim of this study is to empirically validate the foundation of Invariant-Structural Learning, formalized in Section 4.1, and in particular Theorems 1–3, by constructing an end-to-end classifier of concept attractors and evaluating it on a subset of MNIST with intact digit contours to demonstrate that few-shot learning of structural attractors without backpropagation can simultaneously achieve competitive accuracy and local interpretability at the decision level. The empirical task is intentionally limited to images with intact digit contours, which corresponds to a psychophysiologically grounded baseline recognition mode based on a complete structural description; images with broken or fragmented contours are excluded from this study, as their recognition requires an additional associative completion mechanism of a different mathematical nature, which warrants separate consideration.

To achieve this goal, this paper addresses the following five tasks.

1. Formalize the "bitmap → attributed graph" transformation transfer as a constructive implementation of the basis representation: transform each MNIST raster into an attributed graph that preserves the object's skeleton along with the coordinates of critical (anchor) points and the geometric parameters of contour segments.

2. Implement the concept-attractor formation operator as an iterative reduction of a set of graph examples to a stable generalized structure that realizes $R = C \circ S \circ R\_p$ (Section 4.1.3 of the theoretical framework), and verify the parametric stratification predicted by augmentation theory through controlled augmentation by random rotation within ±10° and translation within ±10% of the canvas.

3. Implement the classification step as a comparison based on graph editing distance between the invisible image and each concept-attractor, with diagnostically weighted costs for node replacement across fourteen structural and geometric features and a graded cost scale for edge editing operations, returning the winning attractor along with a similarity score and the logarithmic prior complexity used for its selection.

4. Conduct an experimental evaluation of the proposed classifier on the MNIST dataset with intact contours (8,707 valid images, 13 attractor concepts, 10 classes) and report accuracy, weighted precision, recall, and F1-score along with per-concept metrics, top mismatch pairs, and the error structure by attractor name and feature range.

5. Compare the obtained metrics with published results from major classes of competing methods – classical statistical baselines, few-shot meta-learning methods, and the team's previous six-class study – and discuss the position of this work within the niche of explainable few-shot classifiers.

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

4. Materials and Methods

4.1. Theoretical Framework: Invariant-Structural Learning

Section 4 is divided into two parts. Subsection 4.1 presents the theoretical foundation of invariant-structural learning over hypergraph representations, defining the base spaces, the reduction operator, the learning dynamics, and the central convergence and uniqueness theorems. Subsection 4.2 describes an experimental implementation that validates the framework on a subset of MNIST with connected contours, including a "bitmap → graph" transformation transfer, a concept-attractor formation operator, a graph editing distance comparator, diagnostic feature weighting, and an example of attractor formation for the digit 7.

4.1.1. Basis Spaces and Structures

The input data space X consists of original objects (e.g., MNIST images, geometric shapes, contours, 2D and 3D models). The space P of atomic structural primitives contains the basic building blocks of each representation; for MNIST, these are line segments approximating the contour and contour branching points: endpoints, corner points, intersection points, and convergence points of line segments. The hypergraph space Z formalizes the representation of a single detector neuron. An element of Z is a hypergraph H defined by relation (1), where V is a finite set of vertices – structural primitives, E_s are spatiotemporal edges encoding basic connectivity, and E_p are parametric hyperedges encoding the relationships between sets of parameters of different vertices. The hypergraph is constructed as a layer over the underlying spatial graph; each vertex carries its own set of parameters, so the parametric space is stratified over the structure.

$$H = \left(V, E_s, E_p\right) \tag{1}$$

4.1.2. Extraction and measurement

The primitive detector D simulates the operation of a sensory system: it takes the original input and generates structural primitives. Its implementation is not the subject of this study – D is considered to be given [34]. Each measurement function m_i returns the i-th parameter of a structural element, and m collects all the parameters of the graph. Measurement induces parametric modes for each vertex; each vertex is associated with its modal group – a set of parameters measured for it. The resulting multidimensional parametric space is defined exogenously via coordinate systems and measurement scales; the segmentation of these parametric spaces during training induces a metric. Examples for MNIST skeletonized contours include segment length, segment orientation, the angle between segments, and point coordinates.

4.1.3. Reduction

R_p is an ordering/compression operator on metrics. It does not alter the spatial structure of the hypergraph but acts on the space of parametric relations: continuous values are replaced by discrete classes (segments). The clustering/quantization operator Q with stability parameter ε partitions the parametric space into stable segments σ_i. Parameters may be reduced to qualitative categories or disappear entirely from the hypergraph. The metric d_i in the parametric space determines the segmentation structure. Axiom 1 (segmentation stability) states that the segmentation is idempotent: $Q \circ Q = Q$ on the parametric space for a chosen ε. The structural reduction R_s is performed in three stages.

First, the frequency measure f_x of an element x is defined as the proportion of class examples in which x is present after alignment, where $f_x \in [0,1]$ and N are the number of accumulated class examples.

Second, the structural selection operator S retains only those elements that are reduction-invariant – those that appear (almost) in all class examples. The set A of such elements (anchors) is dynamically updated with each new instance. If the training set contains instances with an incomplete attractor structure, the strict equality $f_x = 1$ is relaxed to $f_x \geq \theta$ with θ – a hyperparameter that typically belongs to $(0.7,1]$; for clarity of presentation, the theory below assumes $\theta = 1$.

Third, the closure/connectivity restoration operator C reconstructs a single hypergraph from the surviving elements (anchors), preserving spatial connectivity.

$$f_x = \frac{1}{N}\sum_{i=1}^{N} 1\left[x \in A_i\right] \tag{2}$$

Definition 1 (anchor). An element $x \in V$ is called an anchor if its frequency measure satisfies $f_x = 1$ (equivalent to $f_x \geq \theta$ for the weakened regime). An anchor is a structural element (or parameter value) that is consistently present in (almost) every instance of the

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

class. It serves as a fixed point around which hypergraphs are aligned; anchors are not removed by reduction and are restored by the operator C to preserve the global structure after removing non-critical vertices ("attractor noise"). For MNIST, anchors are critical branching points of the contour: endpoints and corner points.

The complete reduction operator R, defined by relation (3), is applied first parametrically, then structurally. Axiom 2 (strict reductivity and minimality) states that under the action of R, the size of the hypergraph (in terms of the number of elements and parameters) does not increase, and the resulting attractor is minimal: it contains no elements from $f_x < 1$.

$$R = C^{\circ}S^{\circ}R_p \tag{3}$$

4.1.4. Learning dynamics – cumulative, invariant with respect to order

For a new positive example x', the system constructs the following objects. The anchor alignment μ is a partial mapping $\mu : A' \to A_t$ between the anchors of the new example and the accumulated anchors. The family μ is consistent if the sequential alignment of anchors between examples is order-independent, i.e., there are no conflicting identifications of the same anchor. For cumulative aggregation, given a preserved hypergraph H_t and a new instance H', vertex aggregation identifies vertices via μ; spatial and parametric edges are transferred by the operator μ. The merged hypergraph $H_t \oplus_\mu H'$ merges the identified anchors and adds the remaining elements. The training iteration is defined by relation (4): the operator $\oplus$ expands the structure, while R reduces it. Given a consistent μ and a fixed selection rule μ, the result of the cumulative process is associative and commutative up to structural equivalence E: two hypergraphs are equivalent if their segmented parameters coincide, and the minimal structures are locally isomorphic around common anchors. Class-detector neurons are not required to construct hypersurfaces separating different classes; instead, an invariant hypergraph is constructed independently for each class. The algebraic reduction of hypergraphs allows for an interpretation as the structural plasticity of active dendrites, where patterns of synaptic inputs on dendritic branches form complex logical elements encoding structural invariants [35, 36]

$$H_{t+1} = R\left(H_t \oplus_\mu H'\right). \tag{4}$$

Axiom 3 (learning from positive examples). A class detector neuron is trained only on positive examples of its class. When an input arrives, the system does not "compute" a response but evolves until it reaches a stable state; its dynamics depend on the measurement of its own structural components, modeling internal consistency without an external criterion of optimality. All subsequent statements are derived from the axioms above.

4.1.5. The Principle of Invariant Reduction

The system evolves toward reducing structural redundancy while preserving invariants until it reaches a minimal fixed structure. This principle does not rely on gradient optimization or classical variational principles. Instead, the learning is guided by a monotonic reduction operator on a partially ordered space of structures, which converges to minimal invariant fixed points. A unified hierarchy of coordinate system and measurement scale segmentations is introduced, $\sigma^{(k)}$, which is specified exogenously (modeling the self-organization of segments in biological neural structures); a higher k denotes coarser ("more general") segments. The transition operators between levels are quantization operators at each level. A partial order $|$ is defined on this space (Davey, Priestley, 2002): it is reflexive and transitive (transitivity follows from the nesting and the order on the segmentation hierarchy). The invariance functional *Inv* induces a closure analogous to that in Formal Concept Analysis [37]. Reduction operators are expressed in terms of segments: parametric reduction moves to a coarser level; structural reduction is the composition $C^{\circ}S^{\circ}R_p$. Classical neural networks lose the topological structure of the image, whereas the proposed framework solves the classical neurobiological problem of binding by preserving anchor connections (hyperedges) within the attractor [24]. The discrete invariance gradient – geometrically a vector in the segmentation hierarchy – specifies the direction of generalization growth. Invariants can be added or removed (false), but true invariants are preserved: the dynamics are not monotonic, but the sequence of invariant sets converges to a stable structural core E^*. Extremality is induced by the operator and the order themselves, rather than by an external optimality functional – unlike loss functions in statistical neural networks (Buzsáki, 2010).

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

4.1.6. Structural Attractor and Concept

The central object is the structural attractor: an invariant structure that is a fixed point of the reduction operator and defines the internal model of the class.

Definition 2 (structural attractor). Let K be a class of objects, Z be a hypergraph space, and R be a reduction operator.

A structural attractor of the class K is a hypergraph C_K that satisfies:

(i) invariance – $R(C_K) \cong C_K$;

(ii) structural minimality – for every $C \in Z$ in $R(C) \cong C$, $C_K \mid C$ holds;

(iii) attraction – for every representation $H \in Z$ of the class K, there exists a finite t such that $R^t(H) \cong C_K$.

$$R(C_K) \cong C_K, C_K \preceq C \text{ whenever } R(C) \cong C,\ \exists t < \infty : R^t(H) \cong C_K \quad (6)$$

The attractor C_K is a fixed point R, the minimal carrier of the invariant structure, the boundary of the training process, and the canonical form of the class's objects. Therefore, training is the reduction of various representations of an object to a single structural normal form.

Definition 3 (concept). A concept is a structural attractor of a perceived holistic image, that is, its internal model. Therefore, a concept is an endogenous ontological object for the system. Models with memory of structural attractors form a separate class of self-organized dynamic systems [38, 39]; biological neurons and neural structures are one example. The proposed model suggests a neurobiological interpretation in which

a) the attractor corresponds to a stable activation pattern;

b) anchor elements are realized as stable synaptic configurations;

c) reduction reflects the selection of stable structures in dendritic trees.

This interpretation is illustrative, not a direct biological statement. Self-organization here refers to a modeled process, not a physical one; in particular, feature detectors, coordinate systems, and scales are specified exogenously, whereas reduction operators model the energy dynamics of natural self-organization.

Theorem 1 (Convergence of R)

Conditions:

1) Z is a partially ordered hypergraph space;

2) $R: Z \to Z$ satisfies Axioms 1–2 (segmentation stability, strict reductibility, and minimality);

3) the complexity functional Φ, defined by relation (5), takes integer values and is bounded from below by 0.

Proof outline.

Step 1 (decreasing complexity): at every non-stationary iteration, $\Phi(R(H)) < \Phi(H)$ by Axiom 2.

Step 2 (finiteness): a strictly decreasing sequence of natural numbers is finite.

Step 3 (stabilization): the sequence $(R^t H)_{t \geq 0}$ reaches a fixed point in a finite number of steps.

Conclusion: there exists an C_∞ such that $C_\infty = R(C_\infty)$, where $t^* \leq \Phi(H_0)$ limits the number of steps to convergence by the initial complexity. Corollary 1 (convergence to a class attractor). If a unique structural attractor C_K is induced by the class K, then for every $H \in Z$ of the class K there exists a finite t such that $R^t(H) \cong C_K$.

$$\Phi(H) = |V| + \lambda |E| \quad (5)$$

Theorem 2 (Uniqueness of the C_K).

Conditions:

1) **all examples of the class are anchor-connected, i.e., for any pair** (H_i, H_j) of examples of the class, there exists a chain of common anchors connecting them μ-consistent mappings;

2) R is monotonic and minimal (Axioms 1–2);

3) the dynamics of cumulative composition is order-independent.

Proof outline.

Step 1 (convergence): by Theorem 1, $R^t(H)$ converges to a fixed point.

Step 2 (order independence): the frequency function f_x depends only on the set of accumulated examples, not on the order of their presentation; therefore, the set $x: f_x = 1$ is order-independent.

Step 3 (formalization of the fixed point): by the selection axiom, the set of vertices of the fixed point coincides with the set of elements from $f_x = 1$; the complete attractor is the closure of this set of vertices.

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

Step 4 (uniqueness): anchor connectivity guarantees the consistency of the μ mapping across the entire class (analogous to local-global consistency in bundle theory [40]), so the resulting structure C_K is unique up to E.

Conclusion: there exists a unique C_K such that $C_K = R(C_K)$ and C_K are independent of the order of presentation.

$$\exists! C_K : C_K = R(C_K) \quad (7)$$

4.1.7. Few-shot learning

Consider the subset $T \subset K$. The coverage condition requires that every element of the attractor be present in at least one instance of T. Definition 4 (sufficient covering set): T is a sufficient covering set if (1) the coverage condition holds and (2) anchor connectivity is preserved on T – the hypergraph of the intersection of anchors T is connected, so all examples from T can be matched via common anchors. Corollary 2: If T is a sufficient covering set, then all elements of the attractor survive reduction to T, and the attractor obtained from T coincides with C_K. From this follows the property of embeddability: for every $x \in K$ there exists a mapping $\iota : C_K \to H(x)$ that preserves structural relations – the attractor is embedded in every instance. Thus, an attractor can be defined not by the entire class, but by a small subset of its carriers, which corresponds to the psychology of human perception and opens the way to unsupervised learning. A sufficient covering set ensures the reconstructability of the attractor's structure, but, in general, does not guarantee its uniqueness. Therefore, the stronger concept of an identifying set is introduced. Definition 5 (identifying set): T is an identifying set if (i) it covers all elements of the attractor; (ii) it covers all anchor relations (edges); (iii) anchor connectivity holds. T is then a minimal cover: it induces a unique fixed point of the operator R. Every identifying set is a sufficient cover; the converse generally does not hold. Corollary 3 (Reconstruction): If an attractor is embedded in every instance of the class and there exists an identifying set T^*, then the attractor obtained from T^* is equal to C_K up to E. It is not the cardinality T that matters, but its covering power. Two or three examples may be sufficient, whereas a hundred may not. Given a finite number of anchors and finite connectivity, there exists a finite minimal subset of examples that reconstructs the attractor of the class.

4.1.8. The Role of Augmentation

Augmentation is interpreted as an operator for parametric analysis of structure. Given the example $H_x = (V, E_s, E_p)$, augmentation generates the family $H_{x\ i}^{(i)}$ with structures that are equivalent up to isomorphism and with continuously varying parameters. Augmentation constructs the segmentation σ_i from observed values: it is assumed that parametric variations do not depend on the structure – variations obtained from a single example are representative of the entire class in terms of parameters. Therefore, training is divided into two levels.

1. Structural level: from a small subset T, the structural skeleton is reconstructed – anchors and their relationships.

2. Parametric level: augmentation generates parametric stratification – for each anchor, an interval/segment of valid values. The full attractor combines structural coverage (the small T) and parametric coverage (augmentation). Unlike classical models, where augmentation increases the sample size to reduce overfitting, here augmentation constructs the parametric structure (and thus the metric) through segmentation. Structural examples define the topology; augmentation defines the geometry (metric). Thus, augmentation acts as a metrization operator. The induced metric on the attractor is defined locally for each parameter via its segmentation σ_i, and globally via aggregation by anchors.

Properties of this metric:

1) it is induced by segmentation;

2) it respects invariance – invariants contribute zero to the variation;

3) it is consistent – defined only on anchors and consistent via μ.

Therefore, this metric is not external (exogenous) but arises endogenously: structure → segmentation → metric. This is the opposite of metric learning, contrastive learning, and kernel methods.

4.1.9. The Attention Mechanism and Anchors

The most challenging practical problem in structural learning is the alignment of hypergraphs. Here, it is solved using an attention operator over anchor structural points.

Definition 6 (attention operator). The **attention operator** F is a local selector that chooses a subset $F(H) \subset V$ to construct a matching based on local

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

information – for example, the "capture" points of the MNIST contour. A typical example is given by equation (8), where I is a measure of local information, and τ is a threshold. The operator selects elements with extreme parameter values for a given structure or with rarity/contrast (points of maximum informativeness). Instead of solving the global hypergraph isomorphism problem (which is based on graph editing distance [20]), the attention operator restricts the alignment to the set $F(H)$, transforming the problem from a global combinatorial optimization problem into a local alignment problem and circumventing the NP-hardness of the exact graph edit distance [18]. The alignment μ is therefore defined only on $F(H)$.

$$F(H) = \{x \in V : I(x) \geq \tau\} \quad (8)$$

Axiom 4 (locality of attention and consistency anchors)

1. **The attention operator** F is local – it depends only on H, restricted to $F(H)$.

2. Anchors are invariant under reduction: $F(R(H)) = R(F(H))$ up to structural equivalence.

The attention operator depends on task-induced top-down modulation; it models the brain's active perception process [26, 27, 28]. Yarbus experimentally demonstrated that the trajectory of saccades is determined not only by image properties but also by the cognitive task; in our context, this confirms that the selection of anchor points is a goal-directed action F that minimizes perceptual redundancy.

4.1.10. Neural Map

The class "K" is represented by a self-organizing attractor map (9) – a finite set of structural attractors (detector neurons) – corresponding to subclasses or stable variations within the class. Each attractor of a subclass $C_K^{(j)}$ satisfies $C_K \mid C_K^{(j)}$: the class attractor is a minimal invariant substructure embedded in every subclass attractor. The pseudo-distance d on the structural space is bounded from below between different attractors – see (10) – ensuring structural distinguishability. The class map has a strict hierarchy: the central neuron is the conceptual class detector; d measures the edit distance between this central hypergraph and the hypergraphs of subclass detectors or memorized individual examples. Subclass attractors are formed under specific novelty conditions (their formulation is beyond the scope of this work [41]).

$$M_K = C_{K\ j=1,\ldots,n}^{(j)}, n < \infty \quad (9)$$

$$d\left(C_K^{(i)}, C_K^{(j)}\right) \geq \delta > 0 (i \neq j) \quad (10)$$

Theorem 3 (Self-organization of the attractor map)
Conditions:

1) **The reduction operator** R satisfies the axioms (monotonicity, invariant selection, and connectedness closure);

2) the learning dynamics is given by $H_{t+1} = R\left(H_t \oplus_\mu x_{t+1}\right)$ and for each class K, according to Theorem 2, there exists a structural attractor C_K;

3) a pseudometric d is defined on the structural space, distinguishing structures with accuracy up to E;

4) a separability threshold $\delta > 0$ is fixed.

Conclusions: During the learning process, a finite map $M_K = C_{K\ j=1,\ldots,n}^{(j)}$ is endogenously formed with the following properties:

(i) structural embeddability ($C_K \mid C_K^{(j)}$ for every j);

(ii) separability ($d\left(C_K^{(i)}, C_K^{(j)}\right) \geq \delta$ for $i \neq j$);

(iii) attraction (every example of the class converges under the action of R to some $C_K^{(j)}$); (iv) self-organization (M_K is not specified a priori, but arises as a set of stable fixed points); (v) finiteness $(n < \infty)$.

Proof outline.

Step 1 (convergence to fixed points): By Theorem 1, every learning trajectory stabilizes at a fixed point R. Step 2 (nature of fixed points): by the selection axiom, every limit structure is equal to a closure $x: f_x = 1$ on its trajectory; different subsets of examples can induce different attractors.

Step 3 (emergence of new attractors): a new example x' with an anchor system A', compatible with the current attractor $C^{(j)}$, yields the same attractor; otherwise R generates a new fixed poin t $C^{(j+1)} \neq C^{(j)}$.

Step 4 (endogenous map formation): iterating over the sequence of examples yields a set of fixed points; no attractor is specified exogenously.

Step 5 (structural nesting): by Theorem 2, $C_K \mid C_K^{(j)}$ for every j (an attractor of the class is a common substructure).

Step 6 (separability): by the threshold condition $d \geq \delta$, attractors do not merge under the action of R.

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

Step 7 (finiteness): the structural space is finite due to the finiteness of the number of vertices and parameter segmentations; therefore, $n < \infty$.

Conclusion: the attractor map $M_K = C_K^{(j)}$ arises as a set of stable fixed points $(R, \oplus_\mu, d)$, and its structure is determined endogenously by these three operators, rather than by an external function. Therefore, the map is the result of a dynamic partition of the example space into basins of attraction.

4.1.11. Competition between attractors

The distance between attractors enables "winner-take-all" competition both within a single-class map and between maps of different classes. To achieve this, the response of each detector neuron must aggregate information about the structural and parametric composition of its attractor. This view, in our opinion, sheds light on the question of the neural code – in particular, on the problem of binding [24] and the formation of invariant conceptual representations [29, 30] – assuming that the basic element of the neural code is not the temporal frequency of spikes, but the topology of connections in the dendritic tree [33]. For a finite set of attractors, the distance d is a mixed pseudometric (11) that combines discrete structures and parameters, where d_S is the structural editing distance, and d_P is the parametric distance after alignment. The weights α, β are determined empirically and play a decisive role. According to Riesen [42], the structural distance is defined via partial alignment $\mu \in \Pi(C, C')$ as the sum of three terms – unaligned elements C, unaligned elements C', and the size of the aligned substructure – which yields the number of elements that could not be matched. The parametric distance is computed after matching – either directly or via segments σ_i. The function d satisfies non-negativity, symmetry, and identity $(d(C,C) = 0)$; $d(C_1, C_2) = 0$ implies $C_1 \cong C_2$ (a pseudometric, since the identification is up to E). Together, these properties formally define a metric between attractors that supports competition on a "winner-takes-all" basis.

$$d = \alpha d_S + \beta d_P \tag{11}$$

4.2. Experimental Implementation

Section 4.2 describes how each abstract construct from Section 4.1 – the dictionary of structural primitives, the reduction operator "R", parametric augmentation stratification, and attention-guided anchor selection – is implemented in code and empirically validated. The theoretical foundation of Section 4.1 is formulated over hypergraphs $H = (V, E_s, E_p)$ with parametric hyperedges E_p that connect parametric sets of different vertices. In this implementation, the parametric layer is reduced to a set of attributes per vertex rather than to an explicit hyperedge: each vertex $v \in V$ carries a vector of fourteen continuous and categorical attributes (Table 1) instead of participating in a separately stored relation E_p. The base graph (V, E_s) is therefore a directed attribute graph in NetworkX, and the hyperedge layer emerges implicitly through the diagnostically weighted cost of node substitution, described later in this subsection, which links attributes between matched pairs of vertices at the time of comparison. This is a representative simplification of Section 4.1, not a theoretical deviation: hypergraph terminology is retained in the general formalism of Section 4.1, and thereafter, starting with Section 4.2, the term "graph" is used where this simplification is made.

The conversion of a raster image into an attributed graph is implemented as a three-stage transfer.

First, the image is binarized using a fixed intensity threshold that separates foreground pixels from the background.

Second, the Growing Neural Gas algorithm [43] adaptively places vertices along the object's median line, forming a topologically correct skeleton that respects the local distribution of foreground pixels.

Third, the Ramer–Douglas–Peucker algorithm [44] removes intermediate vertices whose deviation from the chord between neighbors does not exceed a specified tolerance, leaving only structurally significant points: endpoints, corner points, and convergence (branching) points.

The resulting representation is a simple directed attributed graph. Vertices of the first type – Point nodes – represent critical points of the skeleton and implement the V_{anchor} from Definition 1. Vertices of the second kind – Vector nodes – represent oriented segments between adjacent Point nodes and implement structural edges E_s, elevated to first-class objects (so that each segment can carry its own set of parameters, as required by the stratification of the parametric space over the structure from Section 4.1). The spatial coordinates of all vertices are normalized relative to the center of mass of

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

the foreground mask within a symmetric range, ensuring invariance under parallel translation in the image plane.

Each vertex carries a 14-element feature vector. The features are divided into four conceptual categories. (i) Spatial-geometric: $normalized\_x$, $normalized\_y$ – horizontal and vertical coordinates relative to the centroid; distance_to_centroid – radial distance. (ii) Orientational, defined only for Vector nodes: horizontal_direction, vertical_direction – categorical orientation along the two coordinate axes; angle_with_ox – orientation angle in degrees (also defined on Point nodes as the angle of convergence of incident segments). (iii) Structural-functional, defined only for Point nodes: is_endpoint, is_corner – Boolean flags indicating whether the node terminates a single segment or marks a change in direction exceeding a fixed threshold, respectively; junction_angle_min – the smallest internal angle at a branching point. (iv) Topological (per neighborhood): length_ratio_to_max – the length of a segment normalized by the longest segment in the graph; eccentricity – graph-theoretical eccentricity of a node (maximum distance in edges to any other node); avg_neighbor_vector_length – average normalized length of incident edges; neighbor_endpoint_count, neighbor_junction_count – number of neighboring Point nodes by role. The complete set is summarized in Table 1. Attributes defined only on one node type – for example, the orientation triplet on Vector or the structural-functional triplet on Point – are implicitly omitted during comparison when matching with a node of a different type, so the decomposition of values presented later in this subsection always operates on the intersection of features actually present on both sides of the candidate substitution.

Table 1. *Feature vector of a node in the graph representation of an image, grouped by conceptual categories*

Name	Description
Normalized x-coordinate	The horizontal coordinate of the node, normalized relative to the center of mass of the foreground mask; localizes the critical point of the skeleton along the horizontal axis of the image plane.
Normalized y-coordinate	The vertical coordinate of the node, normalized relative to the center of mass; localizes the critical point along the vertical axis.
Distance to the centroid	The normalized radial distance of the node from the center of mass; distinguishes between central and peripheral skeleton points.
Angle with the OX axis	The orientation angle of the segment in degrees for Vector nodes; the angle of convergence of incident segments for Point nodes; an invariant shape descriptor of the segment.
Endpoint flag	Boolean attribute; set to true if the node has degree 1 and terminates a single skeleton segment.
Corner point flag	Boolean attribute; positive when the node marks a change in direction exceeding a fixed angular threshold.
Minimum branching angle	The smallest internal angle at a branching point between converging segments; distinguishes between acute and obtuse convergence points in the skeleton.
Number of neighboring terminal points	The number of neighboring Point nodes marked as endpoints; characterizes the local density of dead-end branches near the node.
Number of neighboring branching points	The number of neighboring Point nodes marked as convergence points; indicates proximity to complex structural connections.
Average length of neighboring vectors	The average normalized length of segments incident to a Point node; characterizes the local geometric scale around the point.
Horizontal direction	A categorical attribute for Vector nodes; the orientation of the segment along the OX axis (left or right).
Vertical direction	A categorical attribute for Vector nodes; the orientation of the segment along the OY axis (up or down).
Length-to-maximum ratio	The ratio of a Vector node's length to the longest segment in the graph; distinguishes between main segments and short connecting segments.
Eccentricity in a graph	The maximum distance in the edges from a vertex to any other vertex in the graph; endpoints have high eccentricity, while central vertices have low eccentricity.

A concept attractor is formed by inductively and iteratively applying the reduction operator "$R = C^{\circ}S^{\circ}R_p$" from Section 4.1 to a small set of skeleton example graphs of the same class. At each step, the operator takes the current state of the concept and a fresh example graph and returns an updated concept that generalizes both inputs. The schematic operation of the operator is given

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

by the equation below: the new state of the concept is the result of the reduction of the pair (current state of the concept, new example).

$$C_{i+1} = R\left(C_i, G_{i+1}\right) \quad (12)$$

Generalization is performed on a per-component basis. For continuous attributes, a valid interval is constructed, bounded by the minimum and maximum values observed among the absorbed instances, and the typical value is stored at the center – this implements the parametric segmentation σ_i from Section 4.1.3. For categorical attributes, the intersection of valid values is stored. For list attributes (where present), set intersection is applied. Structural alignment of nodes between the current concept and the new example uses the same graph edit distance comparator as during classification, ensuring a common core of similarity for the training and classification procedures. After several examples, the iteration stops adding new vertices and only expands the intervals of existing ones, signaling the attainment of a fixed point of the operator R – a structural attractor C_K of the class K (Definition 2).

The role of augmentation in this work is to provide a practical implementation of Section 4.1.8 of the theoretical framework, rather than to address the classic issue of overfitting. Each selected original example H_x is expanded into a family $H_x^{(i)}{}_i$ of variants that are structurally equivalent up to anchor alignment and with continuously varying parameters. Variants are generated by an internal helper that rotates each original by an angle uniformly distributed over $\left[-10^\circ, +10^\circ\right]$ and shifts it by an offset uniformly distributed over $\left[-10\%, +10\%\right]$ of the canvas dimensions, with black fill outside the rotated mask. The number of variants per original ranges from five to ten depending on the concept; the total number of augmentations per concept is listed in Table 2.

This gives rise to three theoretical effects described in Section 4.1.8. First, the augmented family tightens the bounds of the admissible coordinate intervals at each anchor and narrows the estimates of the diagnostic weights (see below); without augmentation, the interval estimated from three to nine selected originals reduces to an almost degenerate width at most coordinates, and the diagnostic weight is uniformly saturated across all features. Second, the segmentation σ_i of each parameter is forced to refer to a stable interval rather than a single point, which directly implements the metric-from-segmentation principle (structure → segmentation → metric): the width of the post-reduction interval governs each feature's contribution to the substitution cost, and a narrower width receives a higher weight during comparison. Third, the augmented family empirically ensures order invariance in training (Theorem 2): after merging the originals in any order with their variants, the resulting attractor is invariant under permutation of the merge order with accuracy up to E, since the set of elements with frequency one on the cumulative carrier does not depend on the order. Therefore, the augmentation contract is not to "increase the training set", but to "make the metric measurable from the data".

Classifying an unseen image boils down to calculating the upper bound of the graph edit distance between the skeleton graph of the test image and each concept-attractor in the alphabet. The implementation uses the iterative upper bound generator "optimize_graph_edit_distance" from NetworkX [45], which is initialized with a polynomial approximation via the Biedel assignment [19] and monotonically refines the upper bound by iterating over alternative vertex mappings. After a fixed time budget – 15 seconds per concept comparison in our campaign – the procedure stops and returns the best upper bound obtained. Therefore, the returned value is a time-limited estimate: it does not guarantee the global minimum of the edit distance, but on graphs with the size and structure of the trained alphabet (3 to 15 anchor nodes), it stabilizes near the optimum within the time budget for over 99% of comparisons. Edge processing is deliberately simplified: any pair of edges is considered compatible, edge removal costs MINOR, edge insertion is free, and explicit attribute substitution at the edge level is not computed. Therefore, the entire structural penalty is concentrated at the vertex level via the cost scale below and the diagnostic weighting below.

Editing operations are evaluated on a six-level monotonically increasing scale that assigns a single cost class to each quality category of correspondence: NO_COST = 0.00 – exact attribute match, MINOR = 0.65 – missing vertex or edge (deletion/insertion on one side), GENERAL = 0.75 – moderate attribute deviation, SEVERE = 1.00 – severe attribute deviation, NO_MATCH = 1.50 – when the attribute is present on only one node, IMPOSSIBLE = 10.00 – label mismatch or prohibited insertion. The scale values are round numbers selected by the engineer within the range

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

MINOR < GENERAL < SEVERE < NO_MATCH < IMPOSSIBLE; only MINOR = 0.65 is empirically established as a key finding of the preliminary tuning study. The ratio NO_MATCH: MINOR ≈ 2.3 makes the presence of a feature on one side and its absence on the other cost more than two structural removals, motivating the comparator to prefer like-to-like matches over forced cross-type substitutions. The IMPOSSIBLE guard condition acts as a strict barrier, completely eliminating Point ↔ Vector substitutions from the optimization.

Within each candidate vertex substitution, the comparator must aggregate the feature-wise contributions into a single substitution score. Uniform aggregation ($1/N$ s per feature), as previous ablations have shown, dilutes the contribution of highly diagnostic features as soon as the list of features exceeds five elements. A mitigation is diagnostic feature weighting based on interval width – calculated concept-by-feature from the post-reduction interval width (defined below). Here, "$SCALE\_STRENGTH(f)$" is a fixed per-feature multiplier (default 1.0; reduced to 0.3 for redundant features such as "cycle_count"), "\[range_width\left({{f}_{c}} \right)\]" is the post-reduction interval width of the feature "f" on the concept "c", and "ε" is a stabilization parameter ("1.0" in the current series) that prevents division by zero on features with degenerate widths. Categorical features are assigned a constant weight, independent of width. The final value of a node's substitution is the weighted sum of the feature-specific attribute values, as defined by the value scale. A narrow interval – that is, a feature whose value is stable across all absorbed originals and their augmented variants – yields a high weight, causing the comparator to focus on those features that the reduction operator has already identified as diagnostic.

$$w(f,c)=\frac{SCALE\_STRENGTH(f)}{range\_width(f,c)+\varepsilon},\ \sum_{f} w(f,c)=1 \quad (13)$$

The final similarity metric is constructed by bounded normalization of the upper bound of the edit distance relative to the size of the larger of the two compared graphs (the formula is given below), where n is equal to the sum of the number of vertices and edges of the corresponding graph. The denominator $\mathrm{GED}_{\mathrm{upper}}+\max(n_{\mathrm{test}},n_C)$ ensures that the mapping is bounded in "[0,1]" $for any non-negative cost, with$ \mathrm{sim}=1" for exact matches, and $\mathrm{sim}\to 0$ for graphs with incompatible sizes or label structures. This mapping partially compensates for the asymmetry in graph size, but does not eliminate the bias in favor of structurally minimal concepts – a phenomenon that was empirically observed in the excessive activation of the compact attractor of the alphabet (Section 5). The predicted class is the concept with the maximum of the scoring function (formula at the end), where "$\Phi(C)$" is the structural complexity of the concept (the sum of the number of vertices and edges), and "λ" is a small positive constant. This is a Bayesian prior in the spirit of the principle of minimal description length: a more complex concept is treated as a priori more specific, partially compensating for the bias of the raw similarity metric in favor of small attractors. Along with the predicted class, the classifier returns an explanatory artifact with four fields: the winning concept's identifier, the similarity score, and two complexities: $\Phi(C)$ and $\Phi(H_{\mathrm{test}})$. These four fields constitute a complete local explanation of the decision: any subsequent audit of the prediction boils down to determining why the winning concept outperformed its closest competitor, without the need for a post-hoc surrogate model.

$$\mathrm{sim}(H_{\mathrm{test}},C)=1-\frac{\mathrm{GED}_{\mathrm{upper}}(H_{\mathrm{test}},C)}{\mathrm{GED}_{\mathrm{upper}}(H_{\mathrm{test}},C)+\max(n_{\mathrm{test}},n_C)} \quad (14)$$

$$\mathrm{score}(C)=\mathrm{sim}(H_{\mathrm{test}},C)+\lambda\cdot\log\Phi(C) \quad (15)$$

To illustrate the operation of the reduction operator, the formation of attractor 7_1 (the only class-7 attractor in the alphabet) is traced step by step. The initial state consists of nine selected originals of the digit 7 from the MNIST training set following Growing Neural Gas skeletonization and Ramer–Douglas–Peucker simplification. Due to handwriting variability, the graph examples differ in the presence of intermediate corner points along the horizontal segment, slight bends in the diagonal segment, and variations in the upper corner angle. The same graph editing distance kernel used during classification identifies an invariant kernel of three critical points in each example: the starting endpoint of the horizontal segment in the upper-left part of the field, the upper corner point in the upper-right part where the horizontal segment transitions into the diagonal segment, and the endpoint of the diagonal segment in the lower part of the field. These three Point vertices are connected by two Vector vertices: one for the horizontal segment between the starting point and the upper corner point, and one for the diagonal segment between the upper corner point and the end point. All other vertices found only in

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

the subset of originals – additional intermediate corners on the horizontal segment, micro-bends along the diagonal – are removed in the next step as elements with frequency " $f_x < 1$", in accordance with Definition 1.

For each remaining vertex of the invariant core, the reduction operator determines the range of valid values for each coordinate feature among the absorbed originals. The intervals reflect the expected topology of the digit: the initial point is located in the upper-left part, the upper corner point in the upper-right, and the final point in the lower half with a relatively wide horizontal spread. Vector nodes inherit similar intervals: the horizontal segment is consistently located in the upper band with a small positional spread, while the diagonal segment crosses the field from the upper-right section to the lower band. The width of the resulting interval directly determines the diagnostic weight: coordinates whose values consistently repeat between originals carry greater weight during classification. After several iterations, the structure stabilizes into a linear bidirectional chain alternating between Point and Vector vertices: initial Point – horizontal Vector – upper corner Point – diagonal Vector – final Point. Subsequent iterations expand the intervals but do not introduce new vertices – this is a sign of reaching an attractor. The resulting compact attractor 7_1 has 5 Point + Vector vertices and a complexity of $\Phi(C) = 5 + 4 = 9$, making it one of the smallest non-trivial attractors in the alphabet. The compact structure encodes the topological invariant of the digit 7, yet that very compactness is the source of the overactivation pattern described in Chapter 5: a small, highly segmented attractor wins many ambiguous matches against larger concepts.

5. Research results

The experiment was conducted on a subset of MNIST images with intact contours. The original images, which were 28×28 pixels in size, were scaled to 100×100 using the Lanczos method so that the skeleton representation would have sufficient resolution to detect endpoints and convergence points. From the initial test set of 10,000 images, the manifest contour integrity filter (structure = complete) retained 8,707 images; the class-wise counts of valid images are shown in Table 3 (the "Volume" column). Images with discontinuous or fragmented contours were deliberately excluded from the scope of this study, as their recognition requires an additional associative completion mechanism of a different mathematical nature, as already noted in Section 3 (Objective). The training set was formed from 76 selected originals, distributed across 13 concepts; each original was augmented using parametric augmentation (random rotation within $[-10^\circ, +10^\circ]$ and translation by $[-10\%, +10\%]$ of the canvas dimensions) with a per-concept multiplier ranging from five to ten (the full per-concept count is given in Table 2), resulting in 729 augmented instances and 805 training instances in total. No test images were used during concept formation.

The classification parameters are specified in the experiment configuration: threshold for skeletonization 110, simplification $\varepsilon = 4.55$; a 14-dimensional feature vector, as shown in Table 1; a graph edit distance comparator with iterative refinement of the upper bound, a time limit of 15 seconds per concept comparison; value scale NO_COST / MINOR / GENERAL / SEVERE / NO_MATCH / IMPOSSIBLE = $0 / 0.65 / 0.75 / 1.0 / 1.5 / 10$; diagnostic weighting with stabilization parameter $\varepsilon = 1.0$; small positive logarithmic a priori coefficient λ on the logarithm of complexity. The full transfer was run end-to-end in batch mode on a multi-instance Nuclio deployment (8 classification consumers, 2 skeletonization consumers, 2 contour analysis consumers) with shared Kafka producers and concept-based graph caching (concept graphs are loaded once during function initialization and reused across messages). The campaign yielded 8,685 successfully classified images and 22 entries in the queue of unresolved messages $(\approx 0.25\%)$ from images where the skeletonization stage was unable to construct a connected graph at the selected threshold. Failed images were excluded from metric calculations.

The trained alphabet contains 13 attractor concepts covering ten classes of digits; classes 1, 2, and 4 each have two attractors that capture different stylistic variants, while the remaining seven classes have one each. The sizes of the concepts (the number of preserved structural primitives) range from three nodes for the simplest variant of the digit 1 to fifteen nodes for the digit 8: the latter is the only digit in the alphabet with a closed-loop topology and a double connection, which is reflected in the largest number of preserved primitives. The compact attractor 7_1 (5 nodes, complexity $\Phi(C) = 9$), one of the smallest non-trivial attractors, is structurally a linear chain of two segments meeting at a single corner point. The largest – 8_1 (15 nodes,

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

complexity ≈ 29) – encodes the topology of a double loop of the number 8. The discrepancy in sizes is a property of the number shapes themselves; it is not a tuning artifact, and it forms the structural background of the overactivation pattern described below. The total number of originals, augmented variants, and training sets is shown in Table 2.

Of the 8,707 images fed into the transfer, 8,685 formed a valid skeleton graph and were included in the metric calculations; the remaining 22 images $(\approx 0.25\%)$ did not form a connected graph at the binarization stage and were excluded from the per-class average metrics without a significant impact on the overall accuracy estimate. Accuracy on the valid set is 85.80%, weighted precision is 89.31%, recall is 85.80%, and F1 score is 86.66%. The 3.51 percentage point gap between weighted precision and recall is informative: it indicates uneven coverage of individual classes – primarily class 2 with a coverage of 60.0% and class 8 with a coverage of 74.7%; both classes are characterized by high precision and low coverage, which is a typical marker of an under-covering attractor alphabet. Class-specific precision, completeness, F1-score, and volume are presented in Table 3.

The highest F1 scores belong to classes 0, 9, and 1 – the digits with the most stable skeletal topology in handwritten form. Moderate F1 scores are observed for classes 3, 4, 5, 6, and 8. Two classes clearly deviate from the trend and account for the residual error budget of the alphabet: class 2 with an F1 of 71.1% (due to a completeness of 60.0% with high precision) and class 7 with an F1 of 78.8% (due to a precision of 66.2% with very high completeness). The excessive activation in class 7 as a complementary counterpart to the coverage deficit in class 2 is not a coincidence – it has a single common structural cause, which is explained below.

Residual errors are dominated by two mismatch patterns, which together account for the majority of the 14.20% class error budget. The complete mismatch matrix is shown in Figure 4; Table 4 lists the ten largest mismatch pairs by absolute count. The dominant pattern – mismatch 2 → 7 (252 instances, 66.0% of class 2 errors) – and the secondary patterns 3 → 7, 4 → 7, and 1 → 7 share a single structural cause. The compact attractor 7_1 with five Point + Vector vertices and a complexity of 9 encodes a linear topology consisting of two segments and a single angular bend. This topology is locally consistent with the stylistic variant of the digit 2 with an open tail (a corner in the upper-left part, followed by an endpoint with an open loop and a diagonal segment to the lower endpoint), with elongated forms of the digit 3 without a closing arc, with segments of the digit 4 converging at a single branching point, and with thin variants of the digit 1 with a slanted upper stroke. None of these classes has a competing attractor in the alphabet that would favor an angle in the upper-left zone with a narrower coordinate interval, so the comparator selects 7_1 as the highest-similarity match for these images. The advantage of a small attractor in mapping is a direct empirical consequence of the "structure–parameter" decomposition from Section 4.1.8: when only the structural skeleton matches and the parametric mapping is broad, a small and widely segmented attractor wins many ambiguous matches against larger concepts.

The second pattern – Class 2 under-coverage – is the dual aspect of the same problem from the perspective of Class 2. The two attractors, 2_1 and 2_2, cover the closed-loop variant and the variant with a wide base of the digit 2, respectively; however, the stylistic variant with an open tail – common among left-handed people and in fast writing – falls outside the convex envelope of the intervals of both attractors but remains within the interval envelope of 7_1. This can be corrected by extending the alphabet with a third stylistic attractor, 2_3 (Section 6, correction (i); Section 7, prospective point (i)). The proportion of unclassified images in class 8 (127 out of 580 class 8 images, $\approx 21.9\%$ of class 8 and 86.4% of class 8 errors according to Table 4) – a separate phenomenon: 8_1 has the largest concept size in the alphabet and requires significant structural matching before evaluation becomes possible. When the binarization threshold breaks the closed loops of the number 8, the resulting graph either fails the connectivity check during the skeletonization stage (and is placed in the queue of unsolved messages, calculated as 22 entries for the entire corpus), or it passes this stage as a fragmented graph that no residual concept can align above the comparator threshold – in this case, the image is rejected before a class label is assigned. This is a pre-classification rejection mode, not a comparator rejection, and it can be mitigated either by adjusting the threshold for class 8 candidates (correction at the extraction stage) or by introducing a fragment completion stage at the graph level (a promising direction beyond the scope of the current study).

Direct numerical comparisons between architectures should be made with caution, as the leading candidate baseline models were evaluated under conditions

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

different from the current ones. Convolutional neural networks from the LeNet-5 family [4] and modern residual variants achieve over 99% accuracy on the standard MNIST test split, but they require thousands of labeled examples per class, a full training cycle with backpropagation, and do not offer built-in explanations – the network's decision is based on linear combinations of feature maps, which can only be interpreted by humans through post-hoc visualization surrogates, the fragility of which is documented by Slack et al. [14]. Support vector machines with Gaussian kernels and invariance-aware preprocessing achieve approximately 98.6% accuracy on the standard MNIST dataset; on raw pixels without invariance augmentation, the result is more modest. Multilayer perceptrons with skewing and affine augmentation achieve 97–98% [4]. The prototype networks by Snell et al. [5], following the canonical 5-way 5-shot protocol on Omniglot or miniImageNet, achieve approximately 95–97% within the protocol benchmark; this figure is not directly comparable to the current 85.80% under the open MNIST test protocol. A previous study by the graph attractors team on six classes [16] on the restricted MNIST alphabet (digits 1, 2, 3, 6, 7, 9; 5,467 test images, 8 attractors, 60-second comparison budget, exact graph editing distance) achieved 85.42% accuracy. The current study extends the methodology to the full ten-class alphabet using contour integrity filtering, expands the alphabet to 13 attractors, introduces diagnostic feature weighting based on post-reduction interval width, and transitions to an iterative upper bound of the graph editing distance with progressive refinement within a significantly smaller time budget – a combination of which yields the 85.80% accuracy reported here.

Therefore, the proposed architecture does not compete with deep convolutional networks in terms of absolute accuracy on the standard MNIST benchmark. Its niche lies in the simultaneous fulfillment of three properties, which in Chapter 2 are described as having been solved separately but not together: training without backpropagation, operation with a small number of examples per class, and local interpretability of each decision at the model level. The result of 85.80%, achieved by thirteen attractor concepts trained on 76 selected originals (from three to nine per concept), demonstrates that this trio is achievable in practice on a non-trivial recognition task – this is empirical confirmation of Tasks (4) and (5) of Section 3 and closes the gap stated in Section 2.

Figures 1–4 provide visual support. Figure 1 shows the complete transfer for a single image: the raw raster, the binary mask, the skeleton after algorithmic approximation of the medial line, the simplified graph, and the selected concept attractor. Figure 2 shows a graph representation of a typical handwritten digit 7 with annotations of critical points. Figure 3 shows the resulting attractor 7_1 after reduction with superimposed valid coordinate intervals. Figure 4 shows the mismatch matrix; the visual distinctiveness of column 7 quantitatively corresponds to the excessive activation of attractor 7_1, documented in Table 4.

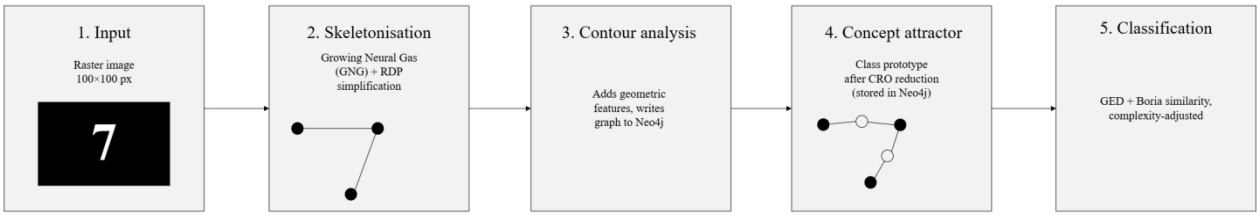

Fig 1. End-to-end transfer from a raster image to classification by a concept-attractor based on graph editing distance

ISSN 2522-9818 (print)
ISSN 2524-2296 (online)
Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

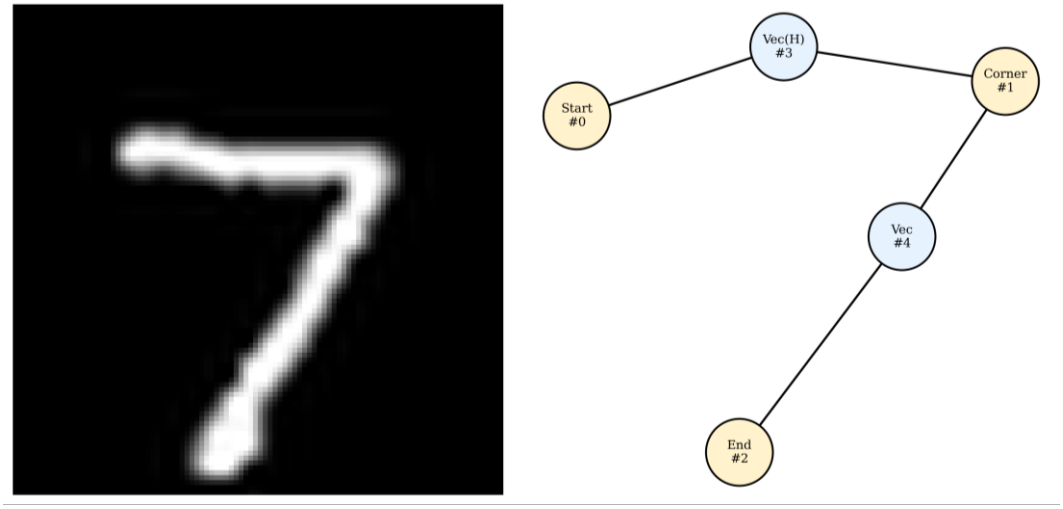

Fig 2. Graph representation of the digit 7: on the left – the original MNIST raster; on the right – the corresponding skeleton graph with Point and Vector nodes

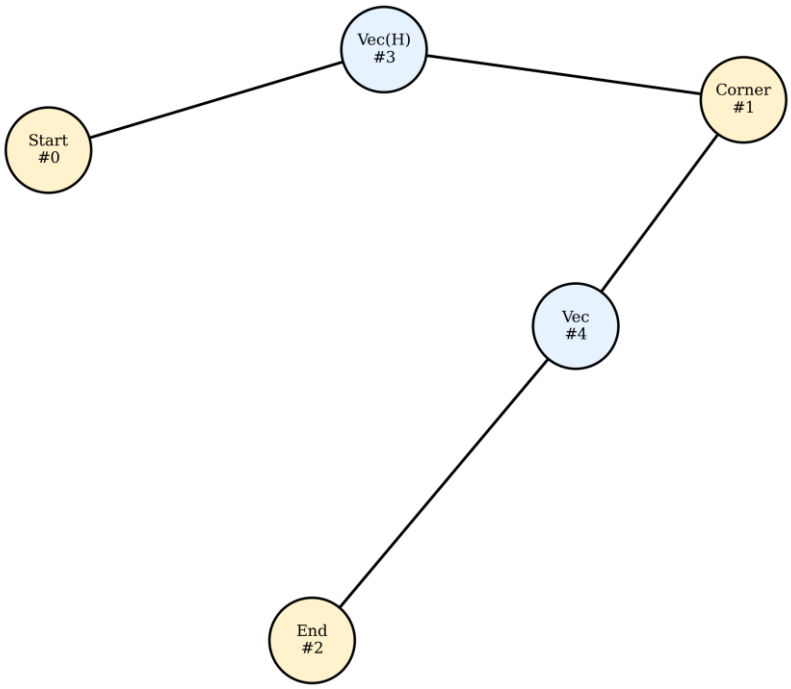

Fig 3. Concept attractor 7_1 after structural reduction: a linear chain consisting of three Point nodes and two Vector nodes – one of the smallest non-trivial attractors in the alphabet

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

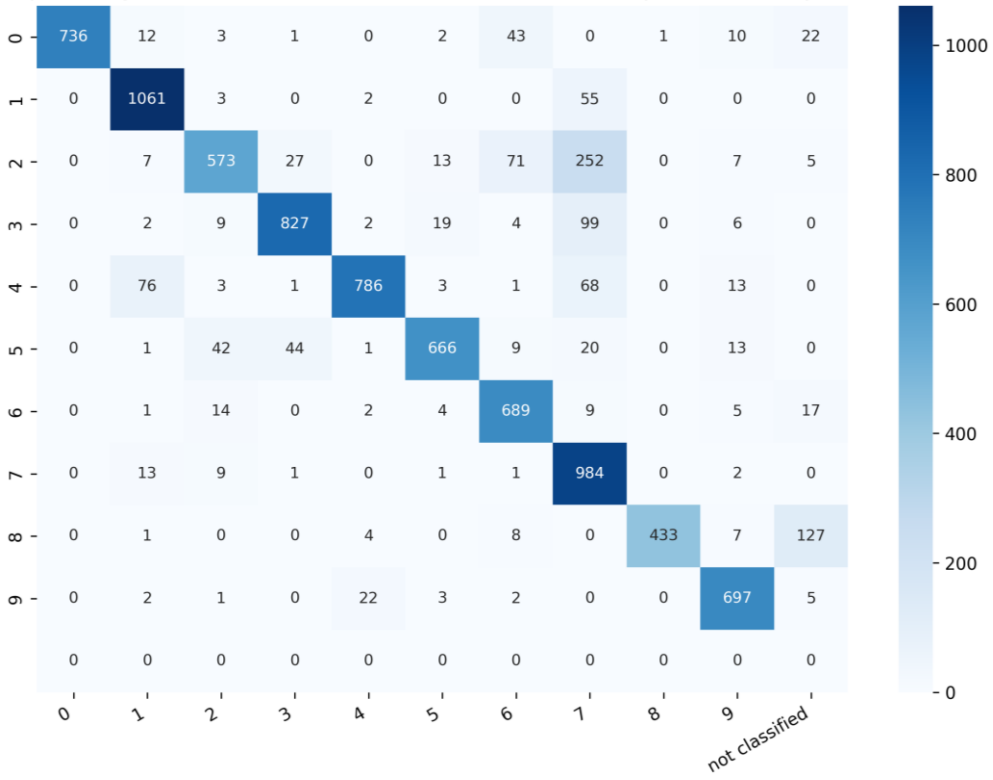

Fig 4. Mismatch matrix on a subset of MNIST with closed contours (8,685 successfully classified images across 10 classes). The "not classified" column contains images that did not match any concept

Table 2. *Concept alphabet: number of originals per concept, number of augmented variants, training set size after augmentation, number of concept nodes, and class membership*

Concept	Class	Originals	Augmented	Training instances	Number of concept nodes
0_1	0	3	30	33	12
1_1	1	3	30	33	7
1_2	1	4	40	44	3
2_1	2	6	60	66	7
2_2	2	3	30	33	10
3_1	3	6	30	36	11
4_1	4	7	70	77	8
4_2	4	6	59	65	9
5_1	5	6	60	66	9
6_1	6	7	70	77	10
7_1	7	9	90	99	5
8_1	8	8	80	88	15
9_1	9	8	80	88	8
Total	–	76	729	805	–

Table 3. *Class-wise classifier metrics on the MNIST subset with connected contours: precision, recall, F1, and volume*

Class	Precision, %	Recall, %	F1, %	Volume
0	100.0	88.7	94.0	830
1	90.2	94.6	92.4	1121
2	87.2	60.0	71.1	955
3	91.8	85.4	88.5	968
4	96.0	82.6	88.8	951
5	93.7	83.7	88.4	796
6	83.2	93.0	87.8	741
7	66.2	97.3	78.8	1011
8	99.8	74.7	85.4	580
9	91.7	95.2	93.4	732

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

Table 4. *Top 10 pairs of discrepancies (true → predicted): absolute count and proportion of class errors*

True class	Predicted class	Count	% of class errors
2	7	252	66.0
8	unclassified	127	86.4
3	7	99	70.2
4	1	76	46.1
2	6	71	18.6
4	7	68	41.2
1	7	55	91.7
5	3	44	33.8
0	6	43	45.7
5	2	42	32.3

6. Discussion of results

This work is conceptual in nature and requires further research. The following discussion addresses six points that arose during the theoretical exposition, followed by the experimental section on the failure modes observed in Chapter 5 and the areas of application where the proposed approach offers a competitive advantage.

First, exogenous primitives detectors. Although the work focuses on self-organizing systems, primitives detectors are specified exogenously rather than learned. This is a model-specific limitation; in the brain, the formation of corresponding sensory areas is genetically programmed rather than learned. Future research should include a model of endogenous formation of primitives – for example, through hierarchical autoencoders or self-organizing maps.

Second, exogenous coordinates and scales. Coordinate systems and measurement scales are also specified exogenously. Endogenous scales and coordinates are likely formed in the brain based on both genetics and experience; a full exploration of this lies beyond the scope of the current study. In the experiment, the spatial coordinate system is fixed on the MNIST dataset, and the feature scales (e.g., normalized coordinates in [–1, +1]) are specified by deterministic normalizers.

Third, the selection of anchors. The central problem of the theory is the search for critical (anchor) structural points that constitute the attractor. The theory assumes that these points are determined by frequency with respect to a hypergraph representation; the efficient search among many candidates is a separate subfield. The current implementation uses a deterministic heuristic for identifying critical points (endpoints, corner points, merge points) as post-hoc preprocessing – this choice is convenient but does not model the endogenous nature of anchor selection.

Fourth, attention as a heuristic. The formal theory of the attention operator is critical, as it is the main mechanism for circumventing the NP-hardness of global hypergraph matching. This work uses only heuristic anchor attention.

Fifth, endogenous ontology and scaling. A key strength of this approach is the natural formation of an endogenous ontology – the system's internal model of the external world. This ontology is formed both by the maps of neurons of individual classes and by their hierarchy [41]. The scalability problem is solved automatically: detector neurons are trained independently, and their hierarchy generates a hierarchy of concepts that has neurobiological support [28].

Sixth, the dendritic computation hypothesis. The key idea is that a neuron's dendritic tree is the primary "computational" system that constructs a structural attractor; the neuron's response is its aggregated characteristic. This is consistent with neurobiological evidence that individual dendritic branches act as independent computational units and that a neuron's firing is determined by the activity of a limited subset of synapses [30, 31, 32].

From an experimental perspective, three failure modes warrant discussion. First, the completeness for class 2 is 60.0%, indicating insufficient coverage by concepts 2_1 and 2_2: the stylistic variability of handwritten 2 (forms with a closed loop versus forms with an open tail) is not captured by the two attractors and would require a third stylistic attractor. This is consistent with the statement in Section 4.1 that intraclass variation should be expressed by additional subclass attractors when the conditions of separability are met. Second, the accuracy on class 7 is 66.2%, reflecting the overfitting of the compact attractor 7_1 (5 nodes, complexity 9), one of the smallest non-trivial attractors in the trained alphabet. The advantage of small attractors in terms of dimension-dependent coverage is a direct

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

consequence of the "structure–parameter" decomposition: when the structural skeleton coincides and the parametric coverage is broad, a small, widely segmented attractor wins many ambiguous matches. Third, the proportion of unclassified items in class 8 is non-trivial, because attractor 8_1 has the highest complexity (15 nodes) in the alphabet, requiring significant structural matching before classification becomes possible. These three patterns confirm that the structure of classification errors based on attractors is interpreted conceptually and across a range of features, satisfying the requirement for explainability set forth in Section 1.

Areas of application. The framework's traceability – where each classification decision points to a named concept attractor and specific ranges of anchor features – makes it a candidate for high-confidence optical symbol recognition (medical forms, legal documents), for industrial defect detection in contour structures (welds, stampings, printed circuit boards), where the number of defect classes is small and labeled examples are scarce, and for educational analytics, where pedagogical explanations are required. Since training requires only a handful of positive examples per class, the approach is also relevant for rapid re-targeting in engineering applications – adapting an existing alphabet to a new font, a new sensor configuration, or a new manufacturing operation – without retraining a large optimization model. In regulated areas covered by the EU AI Act (EU AI Act) [1] and the General Data Protection Regulation (GDPR) [2] regarding the transparency of automated decisions, the ability to automatically generate an interpretable decision log for auditing is a decisive advantage – this aligns with the journal's priority regarding the industrial application of results.

7. Conclusions

This paper proposes a formal learning model based on the structural reduction of hypergraphs and the identification of invariants during the process of aligning a set of observations. Unlike traditional statistical approaches, in which learning is formulated as the minimization of an error functional, the proposed model views learning as convergence to a structural attractor.

The main result is the introduction and formalization of the concept of a structural attractor as a fixed point of a reduction operator acting on the space of coordinated hypergraphs.

It is shown that the attractor has the following properties:

(i) it is the result of a finite reduction process;

(ii) it is unique for a given class of objects;

(iii) it admits an equivalent characterization as the intersection of structural invariants and as the set of single elements of frequency in the cumulative carrier;

(iv) it is invariant with respect to the order of data presentation.

Furthermore, the attractor naturally decomposes into two levels: the structural level (topology of relations) and the parametric level (segments and metrics), which separates the tasks of identifying structural invariants and their parameterization. An alternative formulation based on energy minimization for related neural network models was developed in [46].

The fundamental difference from classical machine learning models lies in the contrast between operators: while statistical neural network models use the minimization operator of the error/divergence functional, the proposed approach employs the algebra of structural invariants – specifically, the fixed-point equation $R(C) = C$ on a partially ordered set of hypergraphs. Learning is interpreted not as a search for the minimum of a functional, but as the computation of the fixed point of the operator on the space of structures. This change in formalism has several consequences.

First, there is no longer a need to specify an error functional or an external optimality criterion.

Second, learning becomes endogenous: it is determined by the data structure.

Third, a natural mechanism for robustness to variations in input representations arises from the identification of invariants.

The model opens up prospects for further research: conditions for the existence and stability of attractors, the design of efficient hypergraph matching algorithms, and the systematic study of learning with a limited number of examples (few-shot and zero-shot modes) without statistical optimization procedures. In summary, this work lays the foundations for an alternative theoretical paradigm of learning, in which structural invariants, reduction, and the dynamics of attractors play a central role.

The empirically proposed transfer achieved 85.80% accuracy (89.31% weighted precision, 85.80% recall, 86.66% F1-score) on the MNIST subset with complete contours (8,707 valid images), having been trained on 76 selected originals – ranging from three to nine

ISSN 2522-9818 (print) Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)

per concept – expanded to 805 instances via using parametric rotation $(\pm 10°)$ and translation $(\pm 10\%)$. The trained alphabet of 13 attractors had a conceptual number of nodes ranging from 3 to 15. These metrics confirm the "structure–parameter" decomposition hypothesis from Sections 4.1.7–4.1.8: a structural skeleton extracted from a small sample is sufficient to cover the class, whereas the metric is induced endogenously through augmentation-driven segmentation rather than learned through statistical regularization.

Three directions for further work emerge from the experimental analysis.

(i) Restore the incompleteness of class 2 by introducing a third stylistic attractor, 2_3 – so that open-tail and closed-loop forms become two distinct subclass detectors rather than sharing a single common attractor.

(ii) Replace heuristic attention based on anchors with an endogenous attention operator that searches for anchors dynamically rather than computing them via deterministic post-hoc preprocessing; this directly addresses the fourth point of Section 6 and operationalizes Definition 6 / Axiom 4 in the implementation.

(iii) Extend the framework to EMNIST and Omniglot to verify whether trained concept-attractors transfer across alphabets – this verifies the claim made in Section 6 that the framework forms a hierarchy of endogenous concepts.

Conflict of interest

The authors declare that they have no conflicts of interest, including financial, personal, authorial, or any other conflicts that could influence the research or the results published in this article.

Funding

The study was conducted without financial support.

Data availability

Data will be provided upon reasonable request. The model's source code, experiment configuration, manifest of valid images, per-concept metrics, error log, and misclassification matrix will be provided upon reasonable request to the corresponding author (Y. Parzhin, e-mail: yparzhyn@augusta.edu); the request must include a justification of the scientific purpose.

Use of artificial intelligence

The authors used AI tools (Anthropic Claude Opus 4.7, model ID "claude-opus-4-7", accessed via the Anthropic API) during four stages of this manuscript's preparation.

1. Preliminary domain analysis – queries regarding related fields (graph editing distance, structural image recognition, few-shot learning, post-hoc explainability) to delineate the scope of related work. The authors independently verified each AI-generated statement against primary sources before inclusion.

2. Literature search – generation of candidate reference lists, which were then verified by the authors via Crossref / OpenAlex / Scopus for DOI, author identity, and publication metadata before inclusion in the citations.

3. Cross-validation of results – systematic comparison of experimental findings (85.80% accuracy on the MNIST subset with connected contours, per-concept F1 scores, top-10 discrepancies) with published results for MNIST (CNN ~99%, SVM ~98%, MLP ~97–98%) and few-shot learning (Prototypical Networks ~95–97% on Omniglot/miniImageNet) to ensure the correct contextualization of statements.

4. Translation editing – correction of errors in the translation of draft materials from Ukrainian and Russian into English; the author reviewed each phrase suggested by the AI before acceptance. All conceptual content (problem formulation, theoretical framework, experimental design, analysis, and conclusions) was written by human authors. AI tools did not participate in the formulation of theorems, proofs, or experimental hypotheses and did not influence the scientific conclusions of the work.

Acknowledgments

The authors express their sincere gratitude to Alexander Schwartzman, Dean of the School of Computer and Cyber Sciences at Augusta University (Georgia, USA), for his invaluable assistance and support in conducting this research.

Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36) *ISSN 2522-9818 (print)*
Innovative technologies and scientific solutions for industries. 2026. No. 2 (36) *ISSN 2524-2296 (online)*

References

1. European Parliament, Council of the European Union (2024), "Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)", Official Journal of the European Union, L series, 12 July 2024.
2. European Parliament, Council of the European Union (2016), "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)", Official Journal of the European Union, L 119, 4 May 2016, pp. 1–88
3. Parzhyn, Y. (2025), "Architecture of information", *arXiv preprint arXiv:2503.21794*, 81 p. DOI: https://doi.org/10.48550/arXiv.2503.21794
4. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998), "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324. DOI: https://doi.org/10.1109/5.726791
5. Snell, J., Swersky, K., and Zemel, R. (2017), "Prototypical networks for few-shot learning", *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, pp. 4077–4087. DOI: https://doi.org/10.48550/arXiv.1703.05175
6. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. (2016), "Matching networks for one-shot learning", *Advances in Neural Information Processing Systems* 29 (NeurIPS 2016), pp. 3630–3638. DOI: https://doi.org/10.48550/arXiv.1606.04080
7. Finn, C., Abbeel, P., and Levine, S. (2017), "Model-agnostic meta-learning for fast adaptation of deep networks", *Proceedings of the 34th International Conference on Machine Learning* (ICML 2017), pp. 1126–1135. DOI: https://doi.org/10.48550/arXiv.1703.03400
8. Minh, D., Wang, H. X., Li, Y. F., and Nguyen, T. N. (2022), "Explainable artificial intelligence: a comprehensive review", *Artificial Intelligence Review*, Vol. 55, No. 5, pp. 3503–3568. DOI: https://doi.org/10.1007/s10462-021-10088-y
9. Hooshyar, D., and Yang, Y. (2024), "Problems with SHAP and LIME in interpretable AI for education: a comparative study of post-hoc explanations and neural-symbolic rule extraction", *IEEE Access*, Vol. 12, pp. 137472–137490. DOI: https://doi.org/10.1109/ACCESS.2024.3463948
10. Rajabi, E., and Etminani, K. (2024), "Knowledge-graph-based explainable AI: a systematic review", *Journal of Information Science*, Vol. 50, No. 4, pp. 1019–1029. DOI: https://doi.org/10.1177/01655515221112844
11. Nawaz, U., Anees-ur-Rahaman, M., and Saeed, Z. (2025), "A review of neuro-symbolic AI integrating reasoning and learning for advanced cognitive systems", *Intelligent Systems with Applications*, Vol. 26, Article 200541. DOI: https://doi.org/10.1016/j.iswa.2025.200541
12. Ribeiro, M. T., Singh, S., and Guestrin, C. (2016), "Why should I trust you?: Explaining the predictions of any classifier", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD 2016), pp. 1135–1144. DOI: https://doi.org/10.1145/2939672.2939778
13. Lundberg, S. M., and Lee, S.-I. (2017), "A unified approach to interpreting model predictions", *Advances in Neural Information Processing Systems 30* (NeurIPS 2017), pp. 4765–4774. DOI: https://doi.org/10.48550/arXiv.1705.07874
14. Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. (2020), "Fooling LIME and SHAP: adversarial attacks on post hoc explanation methods", *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society* (AIES '20), pp. 180–186. DOI: https://doi.org/10.1145/3375627.3375830
15. Ding, K., Wang, J., Li, J., Shu, K., Liu, C., and Liu, H. (2020), "Graph prototypical networks for few-shot learning on attributed networks", *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 295–304. DOI: https://doi.org/10.1145/3340531.3411922
16. Lapin, M., and Bokhan, K. (2025), "Few-shot learning of a graph-based neural network model without backpropagation", *Management Information System and Devices*, No. 187, pp. 103–122. DOI: https://doi.org/10.30837/0135-1710.2025.187.103
17. Sanfeliu, A., and Fu, K.-S. (1983), "A distance measure between attributed relational graphs for pattern recognition", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, No. 3, pp. 353–362. DOI: https://doi.org/10.1109/TSMC.1983.6313167
18. Conte, D., Foggia, P., Sansone, C., and Vento, M. (2004), "Thirty years of graph matching in pattern recognition", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 18, No. 3, pp. 265–298. DOI: https://doi.org/10.1142/S0218001404003228
19. Riesen, K., and Bunke, H. (2009), "Approximate graph edit distance computation by means of bipartite graph matching", *Image and Vision Computing*, Vol. 27, No. 7, pp. 950–959. DOI: https://doi.org/10.1016/j.imavis.2008.04.004
20. Piao, C., Xu, T., Sun, X., Rong, Y., Zhao, K., and Cheng, H. (2023), "Computing graph edit distance via neural graph matching", *Proceedings of the VLDB Endowment*, Vol. 16, No. 8, pp. 1817–1829. DOI: https://doi.org/10.14778/3594512.3594514
21. Moscatelli, A., Piquenot, J., Berar, M., Heroux, P., and Adam, S. (2024), "Graph node matching for edit distance", *Pattern Recognition Letters*, Vol. 184, pp. 14–20. DOI: https://doi.org/10.1016/j.patrec.2024.05.020

22. Xie, Y., Liang, Y., Wen, C., Qin, A. K., and Gong, M. (2024), "Federated collaborative graph neural networks for few-shot graph classification", *Machine Intelligence Research*, Vol. 21, No. 6, pp. 1077–1091. DOI: https://doi.org/10.1007/s11633-023-1463-3
23. Chukhran, I., Udovenko, S., Shergin, V., and Chala, L. (2025), "Method for augmenting 3D point cloud models using graph neural networks", *Innovative Technologies and Scientific Solutions for Industries*, No. 2(32), pp. 129–150. DOI: https://doi.org/10.30837/2522-9818.2025.2.129
24. von der Malsburg, C. (1999), "The what and why of binding: the modeler's perspective", *Neuron*, Vol. 24, No. 1, pp. 95–104. DOI: https://doi.org/10.1016/S0896-6273(00)80825-9
25. Buzsáki, G. (2010), "Neural syntax: cell assemblies, synapsembles, and readers", *Neuron*, Vol. 68, No. 3, pp. 362–385. DOI: https://doi.org/10.1016/j.neuron.2010.09.023
26. Yarbus, A. L. (1967), *Eye Movements and Vision*, Plenum Press, New York, 222 p. DOI: https://doi.org/10.1007/978-1-4899-5379-7
27. Bajcsy, R., Aloimonos, Y., and Tsotsos, J. K. (2018), "Revisiting active perception", *Autonomous Robots*, Vol. 42, No. 2, pp. 177–196. DOI: https://doi.org/10.1007/s10514-017-9615-3
28. Rucci, M., and Victor, J. D. (2015), "The unsteady eye: an information-processing stage, not a bug", *Trends in Neurosciences*, Vol. 38, No. 4, pp. 195–206. DOI: https://doi.org/10.1016/j.tins.2015.01.005
29. Quiroga, R. Q., Reddy, L., Kreiman, G., Koch, C., and Fried, I. (2005), "Invariant visual representation by single neurons in the human brain", *Nature*, Vol. 435, pp. 1102–1107. DOI: https://doi.org/10.1038/nature03687
30. Quiroga, R. Q. (2012), "Concept cells: the building blocks of declarative memory functions", *Nature Reviews Neuroscience*, Vol. 13, No. 8, pp. 587–597. DOI: https://doi.org/10.1038/nrn3251
31. Magee, J. C. (2000), "Dendritic integration of excitatory synaptic input", *Nature Reviews Neuroscience*, Vol. 1, No. 3, pp. 181–190. DOI: https://doi.org/10.1038/35044552
32. Häusser, M. (2001), "Synaptic function: dendritic democracy", *Current Biology*, Vol. 11, No. 1, pp. R10–R12. DOI: https://doi.org/10.1016/S0960-9822(00)00034-8
33. Branco, T., and Häusser, M. (2010), "The single dendritic branch as a fundamental functional unit in the nervous system", *Current Opinion in Neurobiology*, Vol. 20, No. 4, pp. 494–502. DOI: https://doi.org/10.1016/j.conb.2010.07.009
34. Larkum, M. E. (2022), "Are dendrites conceptually useful?", *Neuroscience*, Vol. 489, pp. 4–14. DOI: https://doi.org/10.1016/j.neuroscience.2022.03.008
35. Parzhin, Y., Galkyn, S., and Sobol, M. (2022), "Method for binary contour images vectorization of handwritten characters for recognition by detector neural networks", *2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek)*, pp. 1–6. DOI: https://doi.org/10.1109/KhPIWeek57572.2022.9916331
36. Caspard, N., Leclerc, B., and Monjardet, B. (2012), *Finite Ordered Sets: Concepts, Results and Uses*, Cambridge University Press, Cambridge, 337 p. DOI: https://doi.org/10.1017/CBO9781139005135
37. Hanika, T., and Hirth, J. (2022), "Knowledge cores in large formal contexts", *Annals of Mathematics and Artificial Intelligence*, Vol. 90, No. 6, pp. 537–567. DOI: https://doi.org/10.1007/s10472-022-09790-6
38. Krotov, D. (2023), "A new frontier for Hopfield networks", *Nature Reviews Physics*, Vol. 5, No. 7, pp. 366–367. DOI: https://doi.org/10.1038/s42254-023-00595-y
39. Ramsauer, H., Schäfl, B., Lehner, J., Seidl, P., Widrich, M., Adler, T., Gruber, L., Holzleitner, M., Pavlović, M., Sandve, G. K., Greiff, V., Kreil, D., Kopp, M., Klambauer, G., Brandstetter, J., and Hochreiter, S. (2021), "Hopfield Networks is All You Need", *International Conference on Learning Representations (ICLR 2021)*. DOI: https://doi.org/10.48550/arXiv.2008.02217
40. Bodnar, C., Di Giovanni, F., Chamberlain, B. P., Liò, P., and Bronstein, M. M. (2022), "Neural Sheaf Diffusion: A Topological Perspective on Heterophily and Oversmoothing in GNNs", *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*. DOI: https://doi.org/10.48550/arXiv.2202.04579
41. Parzhin, Y., Kosenko, V., Podorozhniak, A., Malyeyeva, O., and Timofeyev, V. (2020), "Detector neural network vs connectionist artificial neural networks", *Neurocomputing*, Vol. 414, pp. 191–203. DOI: https://doi.org/10.1016/j.neucom.2020.07.025
42. Riesen, K. (2015), *Structural Pattern Recognition with Graph Edit Distance: Approximation Algorithms and Applications*, Springer International Publishing, Cham. DOI: https://doi.org/10.1007/978-3-319-27252-8
43. Fritzke, B. (1995), "A growing neural gas network learns topologies", *Advances in Neural Information Processing Systems 7* (NeurIPS 1994), pp. 625–632. DOI: https://doi.org/10.5555/2998687.2998765
44. Douglas, D. H., and Peucker, T. K. (1973), "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature", *Cartographica: The International Journal for Geographic Information and Geovisualization*, Vol. 10, No. 2, pp. 112–122. DOI: https://doi.org/10.3138/FM57-6770-U75U-7727
45. Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008), "Exploring network structure, dynamics, and function using NetworkX", *Proceedings of the 7th Python in Science Conference* (SciPy 2008), pp. 11–15. DOI: https://doi.org/10.25080/TCWV9851.

46. Parzhyn, Y., Lapin, M., and Bokhan, K. (2025), "A new approach to building energy models of neural networks", *Advanced Information Systems*, Vol. 9, No. 4, pp. 100–119. DOI: https://doi.org/10.20998/2522-9052.2025.4.13

Received (Надійшла) 08.08.2026
Accepted for publication (Прийнята до друку) 21.05.2026
Publication date (Дата публікації) 27.06.2026

Відомості про авторів / About the Authors

Лапін Микита Олексійович – Національний технічний університ "Харківський політехнічний інститут", здобувач вищої освіти ступеня доктора філософії кафедри системного аналізу та інформаційно-аналітичних технологій, Харків, Україна;
Mykyta Lapin – National Technical University "Kharkiv Polytechnic Institute", Postgraduate Student of the Systems Analysis and Information-Analytical Technologies Department, Kharkiv, Ukraine;
e-mail: Mykyta.Lapin@cit.khpi.edu.ua
ORCID ID: https://orcid.org/0009-0003-6307-1172
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=60171161300

Паржин Юрій Володимирович – доктор технічних наук, професор, Школа Комп'ютерних та Кібер Наук Університету Огасти, постдокторант, Огаста, Джорджія, США.
Yurii Parzhyn – Doctor of Technical Sciences, Professor, School of Computer and Cyber Sciences Augusta University, Postdoctoral Fellow, Augusta, Georgia, USA;
e-mail: yparzhyn@augusta.edu
ORCID ID: https://orcid.org/0000-0001-5727-1918
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57224412390

Бохан Костянтин Олександрович – кандидат технічних наук, Національний технічний університет "Харківський політехнічний інститут", доцент кафедри системного аналізу та інформаційно-аналітичних технологій, Харків, Україна;
Kostiantyn Bokhan – Candidate of Technical Sciences, National Technical University "Kharkiv Polytechnic Institute", Associate Professor of the of Systems Analysis and Information-Analytical Technologies Department, Kharkiv, Ukraine;
e-mail: kostiantyn.bokhan@khpi.edu.ua
ORCID ID: https://orcid.org/0000-0003-3375-2527
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57191592568

Перевозник Кирило Максимович – Національний технічний університет "Харківський політехнічний інститут", здобувач вищої освіти ступеня доктора філософії кафедри системного аналізу та інформаційно-аналітичних технологій, Харків, Україна;
Kyrylo Perevoznyk – National Technical University "Kharkiv Polytechnic Institute", Postgraduate Student of the Systems Analysis and Information-Analytical Technologies Department, Kharkiv, Ukraine;
e-mail: kyrylo.perevoznyk@cs.khpi.edu.ua
ORCID ID: https://orcid.org/0009-0009-2327-1501
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=59564678000

Александрова Тетяна Євгенівна – доктор технічних наук, професор, Національний технічний університет "Харківський політехнічний інститут", завідувачка кафедри системного аналізу та інформаційно-аналітичних технологій, Харків, Україна;
Tetiana Aleksandrova – Doctor of Technical Sciences, Professor, National Technical University "Kharkiv Polytechnic Institute", Head of the Systems Analysis and Information-Analytical Technologies Department, Kharkiv, Ukraine;
e-mail: Tetiana.Aleksandrova@khpi.edu.ua
ORCID ID: https://orcid.org/0000-0001-9596-0669
Scopus Author ID: https://www.scopus.com/authid/detail.uri?authorId=57189376480

ISSN 2522-9818 (print) *Сучасний стан наукових досліджень та технологій в промисловості. 2026. № 2 (36)*
ISSN 2524-2296 (online) *Innovative technologies and scientific solutions for industries. 2026. No. 2 (36)*

ІНВАРІАНТНО-СТРУКТУРНЕ НАВЧАННЯ: ФОРМУВАННЯ КОНЦЕПТІВ ЯК ДИНАМІКА ГІПЕРГРАФОВИХ АТРАКТОРІВ

Предметом дослідження є формування концептів класів об’єктів у системах машинного навчання як процесу структурної та параметричної редукції гіперграфового подання, а не як оптимізації функціонала втрат. **Мета роботи** – побудувати альтернативний підхід статистичного навчання штучних нейронних мереж без використання функції помилки (зворотне поширення), у якому реалізується інваріантне структурне навчання, де концепт класу визначається як структурний атрактор (нерухома точка монотонного оператора редукції на частково впорядкованому просторі гіперграфів), і підтвердити цю теорію на прикладі розпізнавання рукописних цифр. **Завдання:** 1) формалізувати основу з аксіомами стабільності сегментації, строгої редуктивності, навчання за позитивними прикладами й локальності уваги; 2) довести збіжність і єдиність атрактора; 3) встановити інваріантність щодо порядку поглинання прикладів; 4) розкласти атрактор на структурний і параметричний рівні; 5) оцінити конвеєр на підмножині MNIST із цілісними контурами. **Методи дослідження:** гіперграфи отримано за допомогою скелетизації бінарних контурів (*Growing Neural Gas* + Рамер – Дуглас – Пекер); концепти-атрактори сформовано редукцією за анкорами – критичними точками контуру. Класифікація використовує порівнювач за відстанню редагування графа з вартостями ознак і логарифмічним апріорі за складністю. Тренувальна множина – 76 оригіналів MNIST, доповнених аугментацією (ротація ±10°, зсув ±10%) до 805 примірників. **Результати.** Конвеєр навчає 13 концептів-атракторів із кількістю вузлів від 3 до 15. З 8 707 допустимих зображень MNIST з цілісними контурами 8 685 утворили валідні скелетні графи; на цій валідній підмножині досягнуто точність 85,80%, зважену влучність 89,31%, повноту 85,80% і F1-міру 86,66%. Структура помилок інтерпретована поконцептно: кожна помилка простежувана до конкретного атрактора й діапазону ознак. **Висновки.** Зафіксована продуктивність, досягнута без функціонала втрат і з трьох до дев’яти унікальних оригіналів на концепт-атрактор, підтверджує теоретичне твердження, що навчання є побудовою структурного атрактора, а не мінімізацією функціонала помилки. Основа дає принциповий шлях до пояснювальної маловибіркової класифікації та конструктивну альтернативу градієнтному навчанню.

Ключові слова: інваріантно-структурне навчання; структурний атрактор; формування концепту; відстань редагування графа; маловибіркове розпізнавання; пояснювальний штучний інтелект; MNIST; редукція гіперграфа.

Бібліографічні описи / Bibliographic descriptions

Лапін М. О., Паржин Ю. В., Бохан К. О., Перевозник К. М., Александрова Т. Є. Інваріантно-структурне навчання: формування концептів як динаміка гіперграфових атракторів. *Сучасний стан наукових досліджень та технологій в промисловості*. 2026. № 2 (36). С. 70–94. DOI: https://doi.org/10.30837/2522-9818.2026.2.070

Lapin, M., Parzhyn, Y., Bokhan, K., Perevoznyk, K., Aleksandrova, T. (2026), "Invariant structural learning: concept formation as hypergraph attractor dynamics", *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (36), P. 70–94. DOI: https://doi.org/10.30837/2522-9818.2026.2.070