UDC 004.8 **DOI:** https://doi.org/10.30837/0135-1710.2025.187.103

M. Lapin, K. Bokhan

FEW-SHOT LEARNING OF A GRAPH-BASED NEURAL NETWORK MODEL WITHOUT BACKPROPAGATION

The subject of this article is a structural graph approach to classifying contour images in *few-shot* mode without using backpropagation**. The core idea** is to make the structure the carrier of explanations: the image is encoded as an attributive graph (critical points and lines as nodes with geometric attributes), and generalization is performed through the formation of concept attractors. **The purpose of the study** is to design and experimentally validate an architecture in which class concepts are formed from several examples (5–6 per class) by means of structural and parametric reductions, ensuring transparency of decisions and rejection of backpropagation of error. **Objectives of the work**: 1) define a vocabulary of nodes/edges and a set of attributes for contour graphs; 2) set normalization and invariance; 3) develop structural and parametric reduction operators as a monotonic simplification of the structure; 4) describe the procedure for aggregating examples into stable concepts; 5) build a classification through graph edit distance (Graph Edit Distance) with practical approximations; 6) compare with representative learning approaches on several examples. **Methods used**. Contour vectorization → bipartite graph (Point/Line as nodes); attributes: coordinates (normalized), length, angle, direction, topological degrees. Reductions: elimination of unstable substructures or noise, alignment of paths between critical points. Concepts are formed by iterative composition of samples; classification is based on the best match of the concept graph (GED with approximations). **Results of the study**. On a MNIST subset with 5–6 basic examples per class (one epoch), a consistent accuracy of approximately 82% was obtained with full traceability of solutions: errors are explained by specific structural similarities. An indicative comparison with SVM/MLP/CNN, as well as metric (ProtoNet) and meta-learning (MAML) lines, is presented in the form of a review graph. **Conclusions**. The structural graph scheme with concepts enables learning from multiple examples without backpropagation of error and provides built-in explanations through an explicit graph structure. Limitations relate to the cost of GED and the quality of skeletonization. Research prospects include optimization of classification algorithms, work with static scenes, and associative recognition.

Keywords: explainable artificial intelligence; few-shot machine learning; backpropagation; graph reduction.

Introduction

Recent advances in artificial intelligence (AI), particularly in deep learning and artificial neural networks (ANNs), have led to significant progress in solving complex problems [1–3]. However, the widespread use of these technologies has revealed a number of fundamental limitations that call into question the possibility of creating truly autonomous and adaptive systems [4–6].

These limitations include: the need for massive amounts of data for training, which requires significant time, computational, and energy resources [7, 8]; fundamental problems with generative models related to trust in information, "hallucinations", and the phenomenon of "entropy gap" [4, 7, 9]; and model degradation when training on recursively generated data (model autophagy disorder, MAD) [10, 11].

In this work, we assume that these problems are fundamental in nature, stemming from the current conceptual paradigm. Modern MLMs are based primarily on the statistical nature of learning and a rigid architecture that is optimized using the backpropagation algorithm [2, 3, 6].

Even specialized approaches to few-shot learning, such as meta-learning (MAML, Prototypical Networks) [12–14], are essentially complex methods of statistical optimization. They do not eliminate the fundamental dependence on statistics and cannot truly learn "from scratch" on a few examples, as they rely on models pre-trained on large data sets or require a complex meta-learning step.

This paper considers an alternative approach based on abandoning backpropagation in favor of biologically motivated structural generalizations. This paper presents a practical computational implementation of such an approach.

We demonstrate how visual patterns (contour images) can be represented as attributed graphs [15–17], where nodes (critical points, lines) and edges (spatial connections) encode the topological and geometric properties of an object.

The learning process is implemented as single-pass few-shot learning without backpropagation. It is based on the application of structural and parametric reduction operators, which operate by monotonic structural simplification. Iterative application of these operators on 5–6 unique samples causes the system to converge to a stable, generalized state with minimal structural complexity – a **generalized concept graph** (or prototype graph).

Analysis of recent studies and publications

The development of structural graph models for learning from a few examples lies at the intersection of several key research areas: few-shot learning [14], explainability methods (XAI) [18, 19], graph representations (GED) [20, 21], and alternative architectures (OvA/OvO) [22]. A review of the literature in these areas reveals fundamental conceptual limitations that the proposed approach aims to address [7, 23–25].

Few-shot/Meta-learning

The dominant deep learning models (CNN, MLP, Transformer) are fundamentally statistical and demonstrate low efficiency when trained on critically small datasets, requiring thousands of examples and many training epochs to achieve acceptable accuracy. To solve this problem, few-shot and meta-learning methods have been proposed [2, 7, 14, 26].

Prototypical Networks learn to identify class prototypes based on distance metrics in embedding space [13]. MAML (Model-Agnostic Meta-Learning) attempts to find the optimal initial weight initialization for fast adaptation [12]. Although both methods significantly improve accuracy on small samples, they do not eliminate the fundamental dependence on statistics and backpropagation.

They require a complex and resource-intensive meta-learning phase on large auxiliary datasets [14, 26].

Thus, this is a transfer of knowledge obtained statistically, rather than true one-pass learning "from scratch".

Explainable AI (XAI) and Graph Representations

As models have grown in complexity, the problem of their interpretability has become more acute. Deep learning models function as "black boxes". Popular XAI methods, such as LIME and SHAP, are post-hoc techniques: they attempt to approximate the behavior of an already trained model rather than explain its actual decision-making process [27, 28]. Studies have shown that such explanations can be unreliable, contradictory, and vulnerable to adversarial attacks [18, 19, 29, 30].

An alternative is "explainability by design", where the internal representation of the model is semantically meaningful [16, 18, 19]. Graph structures are ideal candidates for this because they allow semantics to be explicitly encoded in nodes and edges. Graph Edit Distance (GED) [20, 21] is used to compare such structures. However, GED is an NP-hard problem, which remains a challenge for practical application [31, 32].

Alternative architectures (OvA/OvO) and the problem of feature generalization

To solve classification problems, alternative ANN architectures have long been considered: "one-vs-all" (OvA) and "one-vs-one" (OvO) [22]. This is an approach where, instead of one large network, specialized networks are used (for example, one for each class). This approach is conceptually similar to the one we propose, where we build a single separate "neuron" (concept graph) for each class.

However, in classical implementations of One-vs-All / One-vs-One architectures (OvA/OvO), which rely on the backpropagation algorithm, there are noticeable limitations in detecting examples that go beyond the boundaries of the training data. One-vs-One, OvA/OvO) architectures, which rely on the backpropagation algorithm, there are noticeable limitations in detecting examples that fall outside the training distribution (Out-of-Distribution Detection, OOD) [33, 34]. Networks trained on limited examples do not form stable class separation boundaries.

This is because traditional ANNs generalize only **local recognition features** (e.g., individual textures or angles) and cannot generalize features at the level of the entire structure [23, 24]. Their fully connected and combinatorial nature with stochastic initialization makes it impossible to generalize global topological properties. Our approach solves this problem because generalization occurs not through stochastic optimization of local weights, but through deterministic structural reduction of the graph, which captures **global** topological features.

Synthesis: Identified conceptual gaps

A review of the literature reveals three distinct but interrelated problems:

1. Dependence of few-shot learning methods on the backpropagation algorithm: Leading few-shot methods (MAML, ProtoNets) are not true "zero-shot" learning, but rather knowledge transfer methods that require intensive prior training using backpropagation.

2. Unreliability of XAI: Existing XAI methods (LIME, SHAP) remain mostly post-hoc, unreliable, and vulnerable to attacks.

Feature locality in OvA: Classical architectures (including OvA/OvO) are unable to generalize global/structural features, leading to OOD problems and unstable decision boundaries

Research gaps, Purpose and Objectives

Research gaps

1. Reliability of explanations: Approaches with "by design" explainability are needed, rather than post-hoc approximations (LIME/SHAP).

2. Training on multiple examples without backpropagation: Leading methods (MAML, ProtoNets) still rely on gradient updates. Alternatives are needed that work in low-data modes without backprop.

3. Generalization of global features: Classical ANNs (including OvA) capture local patterns but are not capable of generalizing global topological structure, which is key to shape recognition.

4. GED complexity: Graph edit distance (GED) is NP-hard, which limits its practical application.

The purpose of the work is to develop and experimentally validate a structural graph approach to few-shot classification of contour images without backpropagation, in which the generalization of several class examples is performed through a sequence of structural and parametric reductions, and decision-making has built-in explainability due to the explicit graph structure.

Objectives

1. Representation. Define the representation of a contour image as an attributed graph (node/edge types, geometric attributes, normalization, and invariance) taking into account skeletonization/vectorization properties.

2. Reduction operators. Develop a set of structural (removal of unstable branches, merging of intersections, normalization of paths) and parametric (min-max-center ranges for numerical features) operators that simplify the set of examples into a concept attractor.

3. Aggregation of examples. Build a procedure for forming a concept from 5–6 examples per class in few-shot mode, fixing attribute tolerances and filtering random structures.

4. Classification. Design a concept matching scheme (GED with heuristics based on bipartite matching/local searches) with strict time and quality constraints.

5. Experimental protocol. Conduct tests on a subset of MNIST/similar contour sets: one epoch, 5–6 basic examples/class (+augmentations); evaluate accuracy, concept stability, computation time.

6. Comparison with FSL databases. Compare with representative methods (Prototypical Networks, MAML) as examples of metric and meta-learning approaches; provide an indicative graph (caution regarding different protocols).

Explainability and risks. Explicitly record structural subgraphs/attributes that support decisions; compare with post-hoc explanations and discuss limitations of applicability (when structure "does not explain").

Materials and Methods

This section describes in detail the methodological pipeline used to convert two-dimensional contour images into stable concept graphs and their subsequent classification. The methodology is based on the principles of structural generalization and rejects gradient optimization.

Representation of contours as attributed graphs

To achieve transparency and move away from the "opaque" weight matrices characteristic of traditional neural networks, a representation is proposed where "structure is the carrier of explanations".

The input contour image, obtained after the binarization and skeletonization stages, is transformed into an attributed graph.

The system encodes contours as bipartite graphs, whose structure strictly alternates between *Point* type nodes and *Line* type nodes. This architectural differentiation is fundamental because it allows the topological structure (critical points) to be clearly separated from the geometric properties (the segments that connect them).

Point nodes: Represent the topological structure and critical points of the contour. They are ontologically classified into four main types:

· *EndPoint*: Terminal nodes that mark the beginning or end of an open contour.

· *CornerPoint*: Nodes that mark sharp changes in direction (corners).

· *IntersectionPoint*: Nodes where three or more segments meet.

· *StartPoint*: A designated anchor node that defines the canonical starting point of the graph traversal to ensure consistency of comparisons.

Line nodes: Represent geometric properties. Importantly, line segments are represented as first-class nodes rather than edges. This allows them to be assigned rich semantic and geometric attributes on par with *Point* nodes, which is critical for subsequent parametric reduction operations.

Edges (Interconnections): *Point* and *Line* nodes are connected exclusively by bidirectional edges of type *CONNECTED_TO*. This creates a strict traversal pattern *Point* → *Line* → *Point* → …

Each node carries a set of attributes that encode measurable geometry and topology parameters, including: *normalized_x*, *normalized_y* (coordinates normalized to the invariant range [-1, 1]), *length* (segment length), *angle* (angle for CornerPoint), *quadrant* (discretized direction), *horizontal_direction*, and *vertical_direction*.

Invariance through normalization

To ensure invariance of representation to scale and shift, which is a necessary condition for the formation of stable attractors, all coordinates and related metrics (e.g., *length*) are normalized. Point coordinates are transformed into a centered system with a range of [-1, 1] using the formula:

$$normalized_x = \left(x - center_x\right) / center_x .$$

A similar formula is applied to y.

This process is the first step in parametric reduction ($R_{u,c}$), which converts absolute values specific to a particular instance into relative, generalized parameters.

The learning process as a structural reduction of a graph

The learning process (concept formation) in this work is fundamentally different from traditional statistical optimization (e.g., gradient descent along the loss function). It is viewed as a deterministic process of structural generalization that strives for a state of minimal structural complexity. This most stable, generalized state of the system, representing the invariant essence of the class (e.g., all variants of writing the Fgure "3"), is called a generalized concept graph.

The transition from a set of individual sample graphs $(G_1,\ldots,G_n)$ to a single concept graph C is a process of controlled simplification (reduction) of the structure. This process is controlled by a set of special reduction operators (Custom Reduction Operations, CRO), which act by reducing structural complexity or parametric variability, attempting to simplify the graph to a stable prototype in a finite number of steps.

The general reduction process can be described as a composition of three classes of operators:

$$R = R_w\left(R_{sp}\left(R_{u,c}\left(G_{input}\right)\right)\right),$$

where G_{input} – input graph;

$R_{u,c}$, R_{sp}, R_w – theoretical reduction operators.

A key aspect of our methodology is the direct comparison of these theoretical operators with specific CRO algorithms implemented in the system, as detailed in Table 1.

Table 1. *Structural and parametric reduction (CRO) operators*

Theoretical Operator	Name and Purpose	Practical Implementation (CRO)	Algorithm Details
$R_{u,c}$ (Parametric Reduction)	Minimization of parametric variability. Transition from quantitative values to generalized qualitative ranges.	Parametric Generalization	Numeric properties $v_1,\ldots,v_n$: merge into a rang $min:\min(v_i), max:\max(v_i), center:\mathrm{avg}(v_i)$; this generalizes variations (e.g., length, angle). Categorical properties $s_1,\ldots,s_n$: merge into s_1 only if $s_i = s_1$ for all i; otherwise, the attribute is removed (filtering of inconsistent parameters). List properties $L_1,\ldots,L_n$: merge due to the intersection of sets $L_1 \cap \ldots \cap L_n$; this preserves only universal labels (e.g., Point).
R_{sp} (Structural-Parametric Reduction)	Simplification (reduction) of the structure based on the stability of its parameters.	Path Pruning	For two aligned critical points, the algorithm finds all simple paths between them. It selects the "best match" of paths based on the similarity of nodes and uses the shorter path as a template. Nodes from the longer path that do not have a match are removed. This "eliminates length variations".

Continuation of the table 1

Theoretical Operator	Name and Purpose	Practical Implementation (CRO)	Algorithm Details
R_w (Structural Reduction)	Removal of topological elements that are statistically insignificant (noise).	Endpoint Removal and Intersection Point Merging	Endpoint removal: The algorithm calculates the similarity matrix of endpoints between the concept C_i and the sample G_{i+1}. Endpoints with low similarity (below the threshold) or "extra" points are removed along with the entire path to the nearest critical point. Intersection merging: Consolidates *IntersectionPoint* nodes representing the same structural feature. Applies semantic reduction (e.g., *IntersectionPoint* with a degree < 2 becomes *CornerPoin*t or *EndPoint*).

Iterative algorithm for attractor formation

The learning process is one-pass and does not require backpropagation of error. It iteratively builds an attractor based on a very small sample consisting of 5–6 unique training samples per class.

The concept is initialized with the first sample graph: $C_0 = G_1$. This sample acts as an initial hypothesis about the class structure. Each subsequent sample G_{i+1} is integrated into the current concept C_i using a reduction operation $C_{i+1} = CRO\left(C_i, G_{i+1}\right)$.

Each operation CRO is a five-step process that applies the reduction operators from Table 1:

1. Alignment of starting points: Establishing a common origin for graph traversal C_i and G_{i+1} by clustering and selecting StartPoint.

2. Preprocessing of critical points: Applying structural operators R_w (*Endpoint removal, Intersection merging*) to achieve basic structural compatibility.

3. Traversal synchronization: Generating synchronized paths between corresponding critical points in both graphs.

4. Common structure identification: Applying a structural-parametric operator R_{sp} (*Path pruning*) to normalize paths and eliminate length variations between critical points.

5. Parametric merging: Application of a parametric operator $R_{u,c}$ (*Parametric Generalization*) to merge node attributes that remain after structural reduction.

This iterative process is path-dependent; the order in which samples are submitted affects the final concept graph. This mimics a process where the initial hypothesis (G_1) is iteratively refined under the influence of new data (G_{i+1}), which acts as a reduction force, eliminating sample-specific variations (noise) and leaving only the generalized core.

Classification through approximate graph matching (GED)

The classification (inference) process consists of comparing the graph G_{test} obtained from an unknown input image with each concept graph C_k from the trained library, minimizing the graph edit distance (GED) to the input graph G_{test}:

$$Class\left(G_{test}\right) = \arg\min_k GED\left(G_{test}, C_k\right).$$

GED is defined as the minimum cost of a sequence of operations (insertion, deletion, replacement of nodes/edges) required to transform G_{test} into C_k.

In order for the GED metric to correctly take into account the generalized nature of concepts, we use our own cost functions.

Node Substitution Cost: The cost of substitution a node $v \in G_{test}$ with a node $u \in C_k$ is calculated based on range-based cost functions.

- For numerical attributes (e.g., $length$, $angle$): If the attribute v value (e.g., $v_{length} = 5.5$) falls within the trained attribute u range (e.g., $u_{length} = \{min: 4, max: 7, \ldots\}$), the substitution cost for this attribute is 0. If the value is outside the range, the cost is proportional to the distance to the nearest range boundary.
- For categorical attributes: The cost is 0 in case of exact match or infinite (high) in case of mismatch.
- Label compatibility: The replacement cost is infinite if the base node types are incompatible (e.g., *Line* to *Point*).

Edge editing cost: Reduced cost to prioritize topological differences (presence/absence of nodes) over connectivity differences. Calculating the exact GED is an NP-hard problem. To ensure practical applicability, an approximation is used via a hard 60-second timeout for each individual comparison $GED\left(G_{test}, C_k\right)$.

This timeout acts as a heuristic approximation that interrupts the search for the optimal editing path if it takes too long and returns the best distance found at that moment.

Classification and winner selection mechanism

The proposed architecture implements an approach that is conceptually similar to One-vs-All, where each class k is represented by a separate "neuron" that is a generalized concept graph C_k. The classification (inference) process consists of comparing the contour in the form of a graph G_i with each concept graph C_k from the trained library.

Unlike stochastic networks, where the "excitation" of a neuron is a numerical output (e.g., softmax), in our system, the "excitation" of a k-th neuron is the process of calculating the editorial distance $GED\left(G_{test}, C_k\right)$. To select the final classification result, we apply the Winner-Takes-All concept.

The winner is the class (concept) C_k with the smallest editorial distance from the input graph (G_{test}).

$$Class\left(G_{test}\right) = \arg\min_k \{GED\left(G_{test}, C_1\right), \ldots, GED(G_{test}, C_N)\}.$$

If the distance is the same for several classes, the conflict resolution rule is applied. The class that is structurally more complex is selected.

Complexity is calculated as the sum of the nodes and edges of the graph (Fig. 1).

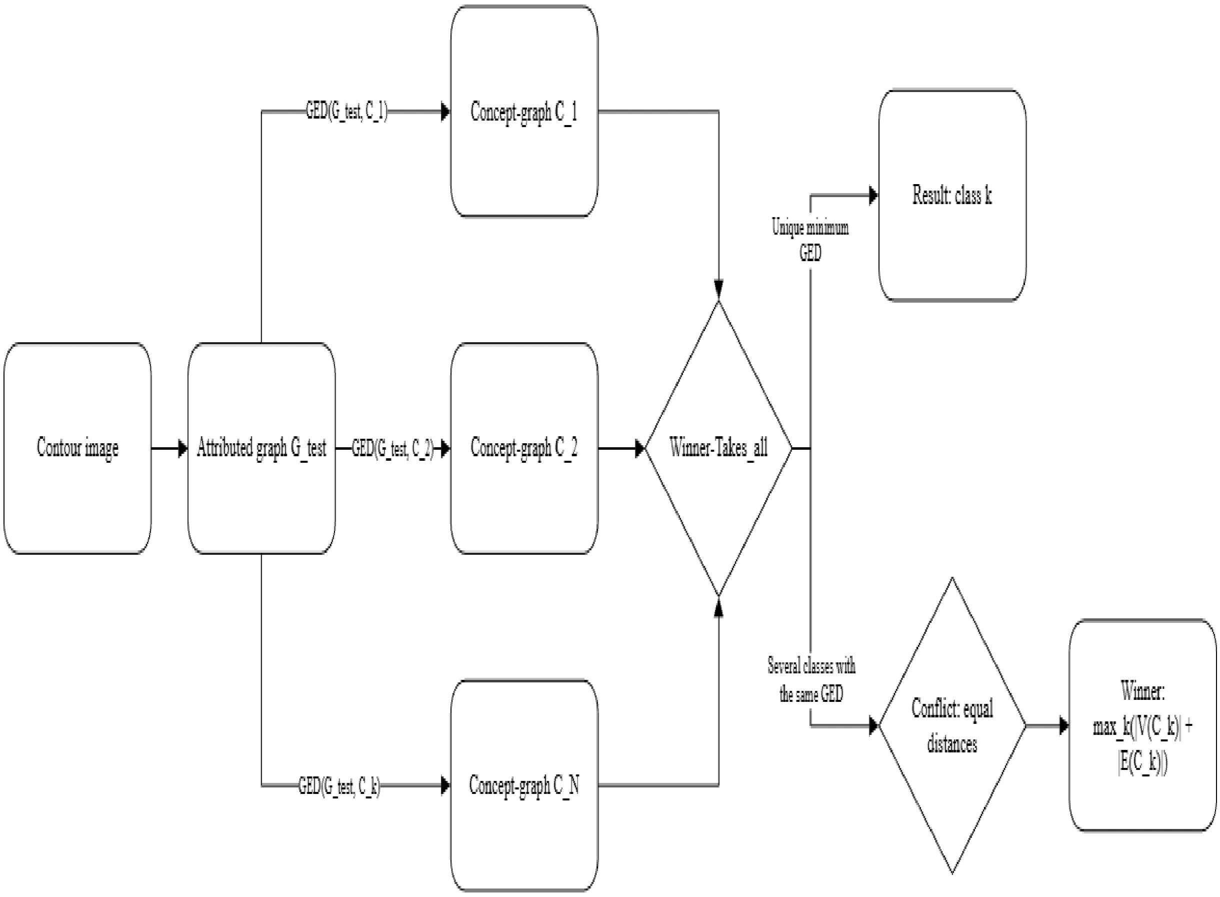


Fig. 1. Classification scheme using GED and WTA

Results and discussion

This section presents empirical validation of the proposed graph-based approach to concept formation.

The goal is not to optimize absolute accuracy, but to demonstrate that stable, explainable concept attractors can be formed from extremely limited data (few-shot learning) and that their performance and error patterns directly follow from their topological and parametric structure.

Experiments are conducted on a subset of MNIST-6 (classes "1", "2", "3", "6", "7", "9"), using 5–6 unique training samples per subclass.

Classification performance on MNIST-6 in Few-Shot mode

The system was trained on 8 concepts covering 6 classes (some classes, such as "1" and "2", had two concepts to represent different writing styles).

Training consisted of iterative structural reduction of 5–6 base samples (with 10 augmentation variants per sample, for a total of about 350 examples) for each concept.

Evaluation was performed on a test sample of 5.467 images that were not used in the formation of concepts.

Overall performance metrics are presented in Table 2.

Table 2. *Overall classification performance (5,467 test images)*

Metrics	Value (%)
Accuracy	82.35
Precision	83.28
Recall	82.35
F1 Score	82.16

These results are conceptually significant. The accuracy of 82.35 % demonstrates that the approach based on the formation of canonical structural attractors without gradient optimization is viable and provides meaningful classification.

The processing pipeline showed high reliability, with a success rate of 100 %, except for 10 images (0.18 %) that could not be processed due to skeletonization errors that resulted in disconnected graphs.

Analysis of class-wise performance and topological distinctiveness

An in-depth analysis of metrics for each class (Table 3) reveals a direct dependence of performance on the structural uniqueness of digits.

Table 3. *Class-based classification metrics*

Digit	Precision (%)	Recall (%)	F1 (%)	Quantity
1	81.46	96.49	88.34	997
2	84.17	60.02	70.07	948
3	78.21	87.28	82.50	983
6	94.23	78.09	85.40	753
7	74.38	82.12	78.06	990
9	91.55	89.57	90.55	786

Key observations:

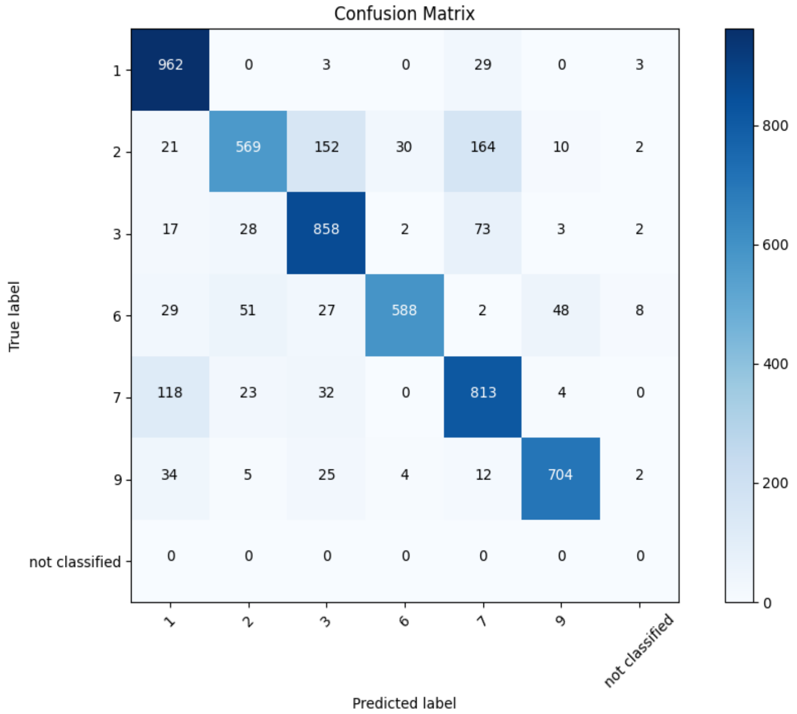
1. High Precision for "6" (94.23 %) and "9" (91.55 %): These classes have the most unique topological signatures – closed cycles represented by *IntersectionPoint* nodes. Their attractors are very specific, which minimizes false positives.

2. Low Recall for "2" (60.02 %): This indicator shows that a significant portion (almost 40 %) of true "2" digits were not recognized. This indicates a high morphological variability in the writing of "2", which the formed concepts ("2_1" and "2_2") were unable to fully cover. Their parametric ranges, studied from only 5–6 samples, proved to be too rigid.

3. Low Precision for "7" (74.38 %): This class was most often confused with others, indicating its structural ambiguity, especially with regard to the digit "1".

Confusion Matrix Analysis

The confusion matrix (Figure 2) provides a deep understanding of how the model makes decisions by visualizing systematic errors that are a direct result of structural and topological similarities.



	1	2	3	6	7	9	not classified
1	962	0	3	0	29	0	3
2	21	569	152	30	164	10	2
3	17	28	858	2	73	3	2
6	29	51	27	588	2	48	8
7	118	23	32	0	813	4	0
9	34	5	25	4	12	704	2
not classified	0	0	0	0	0	0	0

Fig. 2. Matrix of mismatches for the 6-class digit MNIST classification
(The primary mismatch occurs between digits 7 and 1 (angular open contours), and the secondary mismatch occurs between digits 2 and 3 (curved open contours). Digits with closed contours (6, 9) show strong discrimination.)

Primary mismatch: 152 samples of digits "2" were classified as "3". 28 samples of "3" were classified as "2". Secondary discrepancy: 118 samples of digits "7" were classified as "1". Classes "6" and "9" show minimal discrepancy between themselves and other open contours (for example, only 48 samples of "6" were misclassified as "9").

Unlike "black boxes", where the causes of errors are hidden in millions of weights, the errors in this model are fully interpretable. Analysis shows that errors are concentrated along structurally similar pairs:

1. **"2" vs "3"**: Both digits have similar "curved morphology". They are open contours that start on one side, have several bends (represented by CornerPoint nodes), and end on the other side.

2. **"7" vs "1"**: Both digits are "angular open contours". They are both simple paths consisting of a StartPoint, CornerPoint, and EndPoint. The mismatch occurs when the writing of "7" is less curved, or "1" has a more pronounced angle at the beginning.

The fact that the model confuses "7" with "1" (structurally similar) but does not confuse "7" with "6" (structurally different – open contour vs. closed) is strong evidence that the graph matching mechanism works correctly and makes decisions based on topology, as designed.

Stability of concept attractors and structural explainability (XAI)

This section analyzes the final result of the learning process – stable concept attractors, which are the carriers of explanations in the system.

The process of structural reduction transforms multiple training graphs into single canonical structures. Their metrics (Table 4) quantitatively determine the "ideal" form of each digit.

Table 4. *Structural metrics of concept attractors*

Concept	Nods	Edgrs	Av. Degree	Critical points (EP, CP, IP, SP)
1_1	3	2	1.33	1 EP, 1 SP
1_3	3	2	1.33	1 EP, 1 SP
2_1	7	6	1.71	1 EP, 2 CP, 1 SP
2_2	12	12	2.00	1 EP, 3 CP, 1 IP, 1 SP
3_1	7	6	1.71	1 EP, 2 CP, 1 SP
6_1	10	10	2.00	3 CP, 1 IP, 1 SP
7_1	5	4	1.60	1 EP, 1 CP, 1 SP
9_2	8	8	2.00	2 CP, 1 IP, 1 SP

EP = EndPoint; CP = CornerPoint; IP = IntersectionPoint; SP = StartPoint

Analysis of Table I demonstrates a direct correlation between digit topology and the complexity of its attractor.

Simple linear structures ("1"): Concepts "1_1" and "1_3" are minimal, consisting of only 3 nodes (StartPoint, Line, EndPoint). This perfectly reflects their topology as a simple, unbranched path.

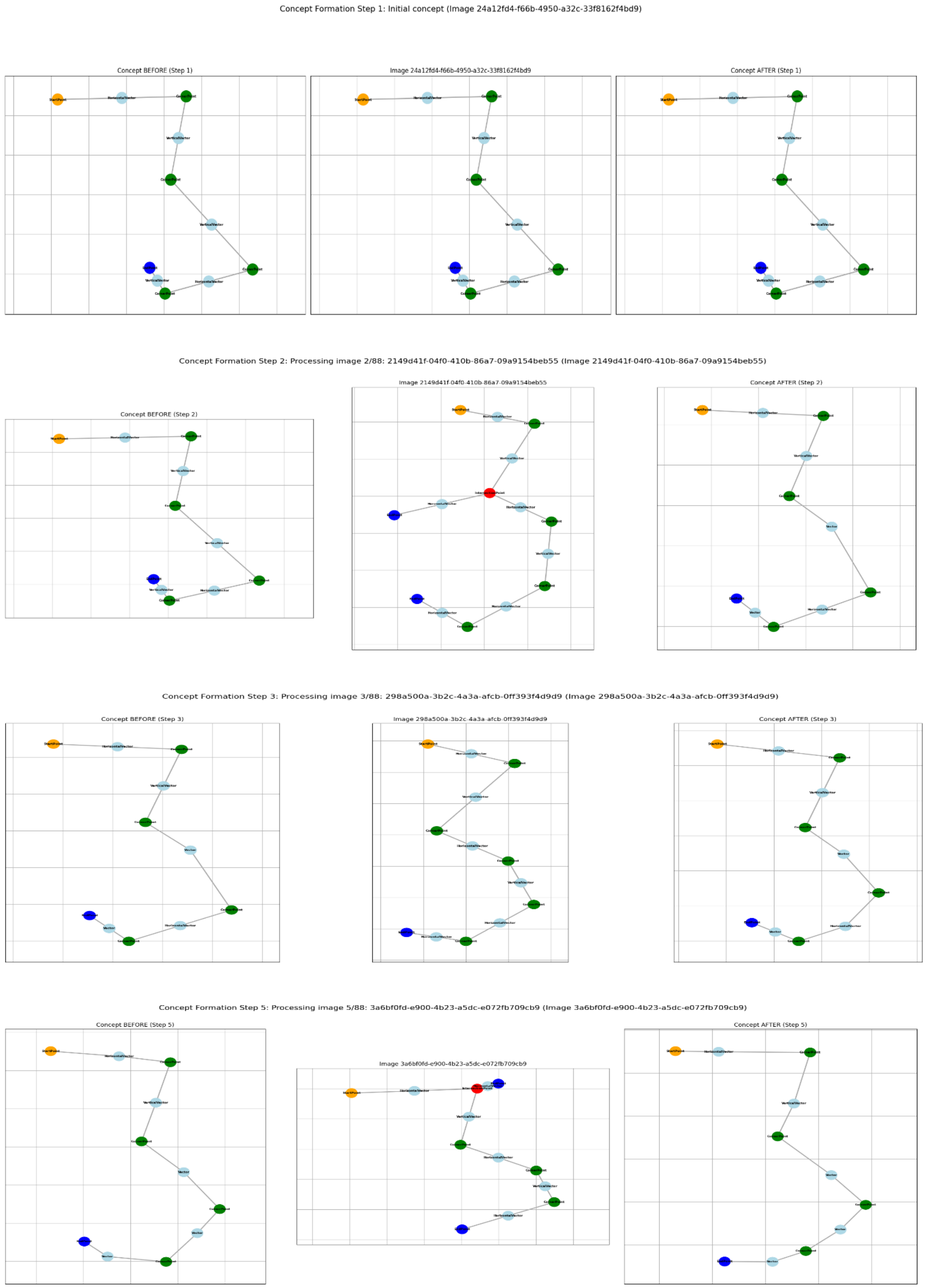
Closed contours ("6", "9"): These concepts have a higher average degree (2.00), indicating the presence of cycles. Importantly, they do not contain an EndPoint (EP = 0), but they do contain an IntersectionPoint (IP = 1) where the cycle closes.

Open curved contours ("2", "3", "7"): These concepts have intermediate complexity (5–12 nodes). They all contain exactly one EndPoint (EP = 1), which topologically marks them as open contours. The number of CornerPoints (CP) encodes the number of bends (e.g., "7_1" has 1 CP, "2_1" has 2 CP).

This table is essentially a dictionary for XAI. The explanation for the "9" classification is that the input image graph successfully matched the "9_2" concept, which is canonically defined as an 8-node structure with 1 IntersectionPoint (cycle) and 0 EndPoints (no free ends).

Case Study: Iterative Stabilization of the Attractor (Digit "3")

The process of concept formation (Figures 3, *a*–*d*) is an empirical demonstration of theoretical reduction operators.



Concept Formation Step 1: Initial concept (Image 24a12fd4-f66b-4950-a32c-33f8162f4bd9)

Concept BEFORE (Step 1)

Image 24a12fd4-f66b-4950-a32c-33f8162f4bd9

Concept AFTER (Step 1)

Concept Formation Step 2: Processing image 2/88: 2149d41f-04f0-410b-86a7-09a9154beb55 (Image 2149d41f-04f0-410b-86a7-09a9154beb55)

Concept BEFORE (Step 2)

Image 2149d41f-04f0-410b-86a7-09a9154beb55

Concept AFTER (Step 2)

Concept Formation Step 3: Processing image 3/88: 298a500a-3b2c-4a3a-afcb-0ff393f4d9d9 (Image 298a500a-3b2c-4a3a-afcb-0ff393f4d9d9)

Concept BEFORE (Step 3)

Image 298a500a-3b2c-4a3a-afcb-0ff393f4d9d9

Concept AFTER (Step 3)

Concept Formation Step 5: Processing image 5/88: 3a6bf0fd-e900-4b23-a5dc-e072fb709cb9 (Image 3a6bf0fd-e900-4b23-a5dc-e072fb709cb9)

Concept BEFORE (Step 5)

Image 3a6bf0fd-e900-4b23-a5dc-e072fb709cb9

Concept AFTER (Step 5)

Fig. 3. Concept formation process

Step 1 ($C_0 = G_1$): The first pattern (G1) establishes the initial concept C_0. It is overly specific and contains all the structural details and noise of the initial pattern (Fig. 3, *a*).

Step 2 ($C_1 = CRO(C_0, G_2)$): Integration of the second sample (G2) reveals a discrepancy – a "redundant endpoint branch". The structural reduction operator (Endpoint removal) is applied, which removes this noise specific to G1. This is a practical implementation of the operator R_w that finds a common substructure (Fig. 3, *b*).

Steps 3 and 4 (C_2, C_3): Further iterations continue this process, removing the "redundant corner point" (Fig. 3, *c*) and another "noise substructure" (Fig. 3, *d*).

The final concept C_3 (Fig. 3, *d*) is a stable attractor representing the most general topological structure ("curved S-shape") common to all training samples. This process is a form of learning without backpropagation of error, where it is not the weight vector that is optimized, but the representation structure itself.

Example of parametric generalization (Digit "3")

Structural reduction determines which nodes remain, while parametric generalization determines how their attributes are generalized to encode variability. Using the example of the concept "3_1" (formed from 3 samples):

Numeric Properties: Attributes such as coordinates are not averaged but converted to ranges ({min, max, center}). This creates flexible decision boundaries.

- $normalized_x$: [–0.7, 0.2] (center –0.33)
- $normalized_y$: [0.3, 0.9] (center 0.63)

Count Properties: Topological variations are also encoded as ranges.

- endpoint_counts: {min: 2, max: 4, center: 2.67}
- intersection_point_counts: {min: 0, max: 2, center: 0.67}

Categorical Properties: Only stored if there is a 100% match.

- contour_type: “OPEN” (all samples were open).
- horizontal_direction: Removed (values were contradictory, e.g., "Left", "Right").

This process is a powerful XAI tool. The range endpoint_counts: {min: 2, max: 4} is a transparent, interpretable boundary. It shows that the model learned from training samples (which had, for example, 2, 4, and 2 endpoints) to expect that valid instances of "3" can have between 2 and 4 endpoints, with an ideal value (center) of 2.67.

This provides recognition flexibility while maintaining verified structural constraints.

Comparative Analysis in the Context of Few-Shot Learning

To evaluate the effectiveness of the proposed approach (referred to as ComAN in the experimental materials), its results are compared with other machine learning models under severely limited data (few-shot) conditions. The data for comparison is taken from experimental reports.

Since the request requires a visual comparison, the following table (Table 5) serves as the data source for a conceptual graph (bar chart) comparing accuracy.

Table 5. *Comparison table*

Model	**Unique samples**	**Epochs of learning**	**Source**	**Accuracy (%)**
ComAN (Our model)	**Up to 36 (5–6/class)**	**1**	**This work**	**82.44**
Nielsen RMNIST/5 (CNN)	50 (5/class)	10–50	Nielsen (2017)	84.38
Prototypical Networks	50–100	Purpose-learning	Snell et al. (2017)	80–90
MAML	50–100	Purpose-learning	Finn et al. (2017)	80–95
CNN (Standard)	500–1000	10–50	Krizhevsky et al. (2012)	74–78
SVM (RBF)	500–600	1	LeCun et al. (1998)	69–75
MLP (Standard)	400–600	10–50	Goodfellow et al. (2016)	53–61

Analysis of this comparison reveals three key conclusions:

1. Competitive accuracy: The accuracy of the ComAN model (82.44 %) is highly competitive. It significantly outperforms standard approaches such as MLP (53–61 %) and SVM (69–75 %), which demonstrate low performance or collapse on such small datasets.

2. Fundamental difference from Meta-Learning: At first glance, MAML (up to 95 %) and Prototypical Networks (up to 90 %) outperform ComAN. However, these models are not "few-shot" in the same sense.

They are meta-learners. They require extensive "pre-training on task distribution" or "on base classes" using backpropagation to "learn to learn". The ComAN model does not require any pre-training. It builds its concepts (attractors) from scratch, de novo, in a single pass (single-epoch training).

This is a radically different learning paradigm based on structural reduction rather than statistical optimization.

3. Comparison with a direct competitor (Nielsen CNN): The most relevant comparison is with Nielsen RMNIST/5, where CNN was trained on the same number of samples (5 per class). CNN Nielsen (84.38 %) shows a slight advantage in accuracy (~ 2 %) over ComAN (82.44 %). However, this advantage comes at the cost of complete loss of interpretability and significantly higher training costs: Nielsen requires 10–50 epochs, backpropagation, dropout, and hyperparameter tuning.

Our model achieves ~ 98 % (82.44/84.38) of SOTA accuracy using only 1 epoch, 0 backpropagation, and providing 100 % transparency.

This comparison empirically confirms the central thesis of the study: the system maintains competitive performance in few-shot mode while providing full structural interpretability.

Conclusions and prospects

Recent advances in artificial intelligence (AI), particularly in deep learning and artificial neural networks (ANNs), have led to significant progress. However, the widespread application of these technologies has revealed fundamental limitations that call into question the viability of the current approach.

Current ANN paradigms face a number of conceptual crises. They require enormous amounts of data for training, as well as significant time, computational, and energy resources. In addition to their high cost, these models, especially generative ones, exhibit significant reliability issues, generating errors and "hallucinations" that significantly undermine confidence in their results.

This directly leads to the phenomenon of "data inbreeding", also known as "Model Autophagic Disorder" (MAD). When models trained to favor statistical probability begin to learn from synthetic data generated by themselves, they enter a recursive cycle.

This process inevitably leads to rapid "information degradation and model collapse" as the entropy of the system continuously decreases, reinforcing averaging and eliminating any novelty.

Conclusions and prospects for further research

This study presents a comprehensive approach to AI that moves away from purely statistical methods in favor of biologically grounded principles of structural generalization.

The paper successfully presents and experimentally validates a unified theoretical and practical framework. This framework combines the principles of structural generalization with a practical, transparent, and highly efficient XAI system based on generalized graph concepts (prototypes).

The main contribution is to demonstrate that abandoning statistical optimization (backpropagation algorithm) in favor of deterministic graph reduction allows:

1. Achieving competitive classification accuracy (82.35 %).
2. Work in training mode on small samples (5–6 samples per class).
3. Perform training in a single pass without backpropagation.
4. Ensure complete internal explainability and transparency of decision-making.

Despite the successful validation of the concept, the current implementation has clear bottlenecks that outline directions for future research.

Computational limitation. The classification (inference) process relies on graph matching, which in general uses graph edit distance (GED), which is an NP-complete problem. This creates a significant computational load at the inference stage, resulting in an average processing time of ~ 3.5 seconds per image and the need for timeouts (e.g., 60 seconds).

In fact, a compromise was made: the computational complexity of training (backpropagation) was replaced by the combinatorial complexity of inference (GED).

Sensory limitation (preprocessing). The model is "fragile" and depends on the quality of the input "sensory" data:

1. Errors in preprocessing lead to a complete failure in processing, since the model cannot construct a correct graph.

2. Invariance is limited by the range used in augmentation (). More significant rotations destroy the structural alignment because they change the attributes (e.g., quadrants) of line nodes.

Representation limitation. The model is "blind" to any information not related to shape. The current approach "discards texture and gradient information", limiting its application exclusively to shape and contour recognition tasks.

The identified limitations directly point to prospects for further research:

1. Short-term prospects include solving immediate engineering problems: researching fast GED approximation algorithms to speed up inference; developing more robust skeletonization methods; and extending the graph representation to include texture and gradient attributes, transforming the model into a multimodal one (in terms of physical parameters).

2. The long-term vision addresses the most fundamental limitation of the current research: "the lack of modeling of evolutionary biological inter-neuronal connections". The current ComAN model successfully implements the concept of a "grandmother cell" – one static concept (neuron) is responsible for one class.

The next fundamental step is to move from modeling individual neurons to modeling dynamic networks of these neurons. This will require the development of mechanisms by which these graph concepts can dynamically interact, compete (e.g., through "Winner Take All" mechanisms), and form more complex, hierarchical "models of the world".

This is the path to creating AI systems that not only mimic biological efficiency but also approach true biological plausibility.

References

1. Goodfellow, I., Courville, A., Bengio, Y. (2016), "Deep learning", *The MIT Press*, Cambridge, Massachusetts, 800 p. ISBN: 978-0-262-03561-3.
2. Heaton, J. (2018), "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning", *Genetic Programming and Evolvable Machines*, Vol. 19, No. 1–2, P. 305–307. DOI: https://doi.org/10.1007/s10710-017-9314-z
3. LeCun, Y., Bengio, Y., Hinton, G. (2015), "Deep learning", *Nature*, Vol. 521, No. 7553, P. 436–444. DOI: https://doi.org/10.1038/nature14539.
4. Bender, E. M., Gebru, T., McMillan-Major, A., Shmitchell, S. (2021), "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?", *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, ACM, P. 610–623. DOI: https://doi.org/10.1145/3442188.3445922
5. Zador, A. M. (2019), "A critique of pure learning and what artificial neural networks can learn from animal brains", *Nature Communications*, Vol. 10, No. 1, P. 3770. DOI: https://doi.org/10.1038/s41467-019-11786-6
6. Marcus, G. (2018), "Deep Learning: A Critical Appraisal", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1801.00631.

7. Parzhyn, Y., Lapin, M., Bokhan, K. (2025), "A New Approach to Building Energy Models of Neural Networks", *Advanced Information Systems*, Vol. 9, Issue 4, P. 100–119. DOI: https://doi.org/10.20998/2522-9052.2025.4.13.
8. Strubell, E., Ganesh, A., McCallum, A. (2019), "Energy and Policy Considerations for Deep Learning in NLP", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1906.02243
9. Ji, Z., Lee, N., Frieske, R. et al. (2023), "Survey of Hallucination in Natural Language Generation", *ACM Computing Surveys*, Vol. 55, Issue 12, P. 1–38. DOI: https://doi.org/10.1145/3571730
10. Alemohammad, S., Casco-Rodriguez, J., Luzi, L. et al. (2023), "Self-Consuming Generative Models Go MAD", *arXiv*. DOI: https://doi.org/10.48550/arXiv.2307.01850.
11. Shumailov, I., Shumaylov, Z., Zhao, Y. et al. (2024), "The Curse of Recursion: Training on Generated Data Makes Models Forget", *arXiv*. DOI: https://doi.org/10.48550/arXiv.2305.17493.
12. Finn, C., Abbeel, P., Levine, S. (2017), "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1703.03400.
13. Snell, J., Swersky, K., Zemel, R. S. (2017), "Prototypical Networks for Few-shot Learning", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1703.05175.
14. Wang, Y., Yao, Q., Kwok, J. et al. (2020), "Generalizing from a Few Examples: A Survey on Few-Shot Learning", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1904.05046.
15. Bai, X., Yang, X., Latecki, L. J., Liu, W., Tu, Z. (2015), "A comparative study using contours and skeletons as shape representations for binary image matching", *Pattern Recognition Letters*, Vol. 65, P. 159–165. DOI: https://doi.org/10.1016/j.patrec.2015.04.007.
16. Parzhin, Y., Galkyn, S., 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)*, Kharkiv, Ukraine, P. 1–6. DOI: https://doi.org/10.1109/KhPIWeek57572.2022.9916331
17. Shen, W., Jiang, Y., Gao, W. et al. (2016), "Shape recognition by bag of skeleton-associated contour parts", *Pattern Recognition Letters*, Vol. 83, P. 321–329. DOI: https://doi.org/10.1016/j.patrec.2016.02.002.
18. Adadi, A., Berrada, M. (2018), "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)", *IEEE Access*, Vol. 6, P. 52138–52160. DOI: https://doi.org/10.1109/ACCESS.2018.2870052.
19. Guidotti, R., Monreale, A., Ruggieri, S. et al. (2018), "A Survey Of Methods For Explaining Black Box Models", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1802.01933.
20. Gao, X., Xiao, B., Tao, D. et al. (2010), "A survey of graph edit distance", *Pattern Analysis and Applications*, Vol. 13, No. 1, P. 113–129. DOI: https://doi.org/10.1007/s10044-008-0141-y.
21. Sanfeliu, A., Fu, K.-S. (1983), "A distance measure between attributed relational graphs for pattern recognition", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, Issue 3, P. 353–362. DOI: https://doi.org/10.1109/TSMC.1983.6313167.
22. Rifkin, R., Klautau, A. (2004), "In Defense of One-Vs-All Classification", *Journal of Machine Learning Research*, Vol. 5, P. 101–141.
23. Parzhin, Y. (2014), "Hypotheses of neural code and the information model of the neuron-detector", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1411.6768.
24. Parzhin, Y. (2017), "The detector principle of constructing artificial neural networks as an alternative to the connectionist paradigm", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1707.03623.
25. Parzhin, Y. (2025), "Architecture of Information", *arXiv*. DOI: https://doi.org/10.48550/arXiv.2503.21794.

26. Chen, W.-Y., Liu, Y.-C., Kira, Z. et al. (2020), "A Closer Look at Few-shot Classification", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1904.04232.
27. Lundberg, S., Lee, S.-I. (2017), "A Unified Approach to Interpreting Model Predictions", *arXiv*. DOI: https://doi.org/10.48550/ARXIV.1705.07874.
28. Ribeiro, M. T., Singh, S., Guestrin, C. (2016), "«Why Should I Trust You?»: Explaining the Predictions of Any Classifier", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1602.04938.
29. Rudin, C. (2019), "Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1811.10154.
30. Slack, D., Hilgard, S., Jia, E. et al. (2020), "Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1911.02508.
31. Blumenthal, D. B., Gamper, J. (2020), "On the exact computation of the graph edit distance", *Pattern Recognition Letters*, Vol. 134, P. 46–57. DOI: https://doi.org/10.1016/j.patrec.2018.05.002.
32. Bougleux, S., Brun, L., Carletti, V. et al. (2017), "Graph edit distance as a quadratic assignment problem", *Pattern Recognition Letters*, Vol. 87, P. 38–46. DOI: https://doi.org/10.1016/j.patrec.2016.10.001.
33. Hendrycks, D., Gimpel, K. (2018), "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks", *arXiv*. DOI: https://doi.org/10.48550/arXiv.1610.02136.
34. Yang, J., Zhou, K., Li, Y. et al. (2024), "Generalized Out-of-Distribution Detection: A Survey", *arXiv*. DOI: https://doi.org/10.48550/arXiv.2110.11334.

Received (Надійшла) 18.11.2025
Accepted for publication (Прийнята до друку) 30.11.2025
Publication date (Дата публікації) 28.12.2025

About the Authors / Відомості про авторів

Lapin Mykyta – National Technical University "Kharkiv Polytechnic Institute", PhD Student, Department of System Analysis and Information and Analytical Technologies, Kharkiv, Ukraine; e-mail: Mykyta.Lapin@cit.khpi.edu.ua; ORCID ID: https://orcid.org/0009-0003-6307-1172

Bokhan Kostiantyn – PhD (Engineering Sciences), National Technical University "Kharkiv Polytechnic Institute", Associate Professor at the Department of System Analysis and Information and Analytical Technologies, Kharkiv, Ukraine; e-mail: kostiantyn.bokhan@khpi.edu.ua; ORCID ID: https://orcid.org/0000-0003-3375-2527

Лапін Микита Олексійович – Національний технічний університет "Харківський політехнічний інститут", аспірант кафедри системного аналізу та інформаційно-аналітичних технологій, Харків, Україна.

Бохан Костянтин Олександрович – кандидат технічних наук, Національний технічний університет "Харківський політехнічний інститут", доцент кафедри системного аналізу та інформаційно-аналітичних технологій, Харків, Україна.

НАВЧАННЯ ЗА КІЛЬКОМА ПРИКЛАДАМИ (*FEW-SHOT*) ГРАФОВОЇ МОДЕЛІ НЕЙРОННОЇ МЕРЕЖІ БЕЗ ВИКОРИСТАННЯ ЗВОРОТНОГО ПОШИРЕННЯ ПОМИЛКИ

Предметом роботи є структурно-графовий підхід до класифікації контурних зображень у режимі *few-shot* без використання зворотного поширення похибки**. Основна ідея** – зробити структуру носієм пояснень: зображення кодується у вигляді атрибутивного графа (критичні точки й лінії як вузли з геометричними атрибутами), а узагальнення виконується через формування концепт-атракторів. **Мета дослідження** – спроєктувати та експериментально підтвердити архітектуру, у якій концепти класів утворюються з кількох прикладів (5–6 на клас) способом структурних і параметричних редукцій, забезпечуючи прозорість рішень і відмову від зворотного поширення помилки. **Завдання роботи**: 1) визначити словник вузлів / ребер і набір атрибутів для контурних графів; 2) задати нормалізацію та інваріантності; 3) розробити структурні та параметричні редукційні оператори як монотонне спрощення структури; 4) описати процедуру агрегації прикладів у стабільні концепти; 5) побудувати класифікацію через відстань редагування графа (*Graph Edit Distance*) з практичними апроксимаціями; 6) порівняти з репрезентативними підходами навчання за кількома прикладами. **Застосовані методи**. Векторизація контуру → двочастковий граф (*Point/Line* як вузли); атрибути: координати (нормовані), довжина, кут, напрям, топологічні степені. Редукції: усунення нестабільних підструктур або шумів, узгодження шляхів між критичними точками. Концепти утворюються ітеративною композицією зразків; класифікація – за найкращою відповідністю графа концепту (GED з апроксимаціями). **Результати дослідження**. На підмножині MNIST із 5–6 базовими прикладами на клас (одна епоха) отримано узгоджувану точність приблизно 82 % за повної трасованості рішень: помилки пояснюються конкретними структурними подібностями. Подано індикативне порівняння з SVM/MLP/CNN, а також метричною (*ProtoNet*) і метанавчальною (MAML) лініями у вигляді оглядового графіка. **Висновки**. Структурно-графова схема з концептами забезпечує навчання за кількома прикладами без зворотного поширення помилки й надає вбудовані пояснення через явну графову структуру. Обмеження стосуються вартості GED та якості скелетизації. Перспективи дослідження – оптимізація алгоритмів класифікації, робота зі статичними сценами й асоціативне розпізнавання.

Ключові слова: зрозумілий штучний інтелект; *few-shot* машинне навчання; зворотне поширення помилки; редукція графів.

Bibliographic descriptions / Бібліографічні описи

Lapin, M., Bokhan, K. (2025), "Few-shot learning of a graph-based neural network model without backpropagation", *Management Information Systems and Devises*, No. 4 (187), P. 103–122. DOI: https://doi.org/10.30837/0135-1710.2025.187.103

Лапін М. О., Бохан К. О. Навчання за кількома прикладами (*few-shot*) графової моделі нейронної мережі без використання зворотного поширення помилки. *Автоматизовані системи управління та прилади автоматики*. 2025. № 4 (187). С. 103–122.
DOI: https://doi.org/10.30837/0135-1710.2025.187.103