Pairwise Surrogate Learning for Neural Architecture Ranking in Digital Pathology

Zeliha Kaya Akçelik[1], Sümeyye Zülal Dik[1], Semiha Koç[1,2], Zeki Kuş[3], Musa Aydin[4]

[1] Department of Computer Engineering, Fatih Sultan Mehmet Vakif University, İstanbul, Türkiye
[2] Department of Computer Engineering, Marmara University, İstanbul, Türkiye
[3] Department of Artificial Intelligence and Data Engineering, Fatih Sultan Mehmet Vakıf University, İstanbul, Türkiye
[4] Department of Artificial Intelligence and Data Engineering, Samsun University, Samsun, Türkiye
Email: zkaya@fsm.edu.tr, szdik@fsm.edu.tr, semiha.koc@marmara.edu.tr, zkus@fsm.edu.tr, musa.aydin@samsun.edu.tr

2026 11th International Conference on Machine Learning Technologies (ICMLT) | 979-8-3315-4865-0/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICMLT69916.2026.11688928

***Abstract*—Neural Architecture Search can automate the design of accurate and efficient deep networks for whole-slide image analysis in digital pathology, but exhaustive evaluation of candidate architectures is computationally expensive. This study proposes a pairwise surrogate-learning framework to predict relative performance between architectures generated from two NAS formulations: Non-Plato NAS and Path-Based NAS. For each of four pathology datasets (EBHI, BCNB, TCGA, and SPIDER), 200 architectures per search space are sampled via Random Search and trained to obtain F1-based fitness. Ordered architecture pairs are then labeled by which model performs better, enabling supervised learning of auxiliary predictors. We compare tree-based models and graph neural networks (GNNs) using both pairwise classification metrics and ranking/selection metrics after aggregating pairwise outcomes into global rankings. Results show boosted trees are most robust, while GNNs benefit from structured path-based encodings.**

***Index Terms*—digital pathology, surrogate-based learning, NAS**

I. Introduction

Digital pathology digitizes, stores, and analyzes tissue slides using computational methods, replacing manual microscopy. Advances in slide scanners support high-resolution whole-slide imaging with detailed morphological information for tumor detection, grading, and treatment assessment. However, WSIs' gigapixel resolution, tissue heterogeneity, and computational demands pose challenges. Neural Architecture Search (NAS) automates the design of accurate, efficient models, suitable for large-scale WSI analysis.

NAS has been increasingly explored in digital pathology to overcome the limitations of manually designed neural networks. Early studies showed that automated architecture design can surpass handcrafted models in accuracy and efficiency. PBC-NAS used an evolutionary, cell-based framework for blood cell classification, demonstrating that optimizing network topology improves accuracy–complexity trade-offs over manual designs [1]. Beyond patch-level analysis, NAS has been applied to WSI. Xu et al. [2] proposed a NAS framework for basal cell carcinoma detection in WSIs, where evolutionary search identified task-specific architectures that outperformed ImageNet backbones within the same computational budget.

Surrogate-assisted NAS methods accelerate architecture search by combining evolutionary algorithms with performance prediction, reducing costly evaluations [3]. The Neural Predictor framework learns a regression model from limited data to predict validation accuracy, thereby enabling faster candidate preselection and higher sample efficiency than traditional evolutionary NAS [4]. Recently, surrogate-assisted NAS has shifted from absolute performance regression to modeling relative performance between architectures. SiamNAS [5] employs a Siamese-network surrogate to predict dominance relations in multi-objective NAS, enabling evolutionary search to find Pareto-optimal solutions with fewer costly evaluations. Building on these advances, recent work has made NAS in digital pathology more structured and efficient. A path-based NAS framework with DAG search spaces and continuous encoding was introduced for WSI classification, using a surrogate-assisted multi-objective evolutionary strategy to optimize both accuracy and model complexity [6]. Similarly, MOLiNAS [7] addressed whole-slide multi-class blood cell segmentation with a lightweight, multi-objective NAS framework that combines surrogate-assisted evolutionary optimization, achieving strong performance with significantly reduced FLOPs and model size.

In this study, we analyze and leverage architectures produced by different NAS formulations. Specifically, we consider two distinct NAS search spaces and encoding strategies—Non-Plato NAS and Path-Based NAS—and apply Random Search to generate candidate architectures across four whole-slide image datasets. The resulting architectures are stored and used to construct a surrogate learning problem, in which auxiliary predictors are trained to compare architectures pairwise. We evaluate these predictors using both classification-based metrics on pairwise comparisons and ranking-based metrics that assess their ability to recover global architecture rankings and identify top-performing models.

II. Methodology

A. Non-Plato Neural Architecture Search

The Non-Plato Neural Architecture Search (NP-NAS) approach used in this study is developed based on the search space definition and encoding principles previously introduced within a cell-based NAS framework [8]. In the NP-NAS search space, each cell is represented as a directed acyclic graph

Authorized licensed use limited to: FATIH SULTAN MEHMET UNIVERSITY. Downloaded on November 11,2026 at 07:59:29 UTC from IEEE Xplore. Restrictions apply.

(DAG), where nodes correspond to feature-extraction operations and directed edges represent information flow between nodes. Each cell comprises six nodes: one input node, one output node, and four intermediate nodes. As illustrated in Figure 1, each intermediate node can establish connections with at most two of the subsequent nodes. Consequently, the resulting cell topology consists of a maximum of nine directed edges. These topological constraints control the complexity of the cell architecture and enable a more structured exploration of the search space. In NP-NAS, each candidate architecture is represented by a 16-dimensional vector with values in [0,1]. These continuous values are mapped to discrete rules within the search space. The first nine elements define node connections by mapping to possible successor nodes. The next four specify two-dimensional operations for intermediate nodes, while the last three indicate external architectural decisions. As shown in Figure 1, while cell structures in Path-Based NAS approaches are generated through path selection mechanisms, cell formation in the NP-NAS method is achieved by independently determining the connections between nodes. This comparison is presented not to propose the NAS methods themselves, but rather to conceptually and visually explain how a solution is transformed into a cell structure under different NAS formulations.

B. Path-Based Neural Architecture Search

The Path-Based Neural Architecture Search (PB-NAS) approach adopted in this study is based on the search space definition and architectural encoding principles previously introduced within a cell-based NAS framework [6]. In the PB-NAS search space, each cell is modeled as a DAG, where nodes correspond to feature extraction operations and directed edges represent the flow of information between nodes. The cell structure consists of an input node, an output node, and a fixed number of intermediate nodes. Unlike approaches that determine node connections independently, PB-NAS constructs the cell topology by selecting valid paths on the DAG. In PB-NAS, each candidate architecture is represented by a 10-dimensional vector with values in [0,1]. These continuous values are mapped to discrete rules within the search space. The first three elements define which valid paths are chosen. The next four specify two-dimensional operations for intermediate nodes, while the last three indicate external architectural decisions. The cell output is then made by applying the chosen operations to the intermediate feature maps along the selected paths and combining the node outputs. As illustrated in Figure 1, while PB-NAS generates the cell structure through a path-selection-based mechanism, the NP-NAS approach constructs the cell by independently determining the connections between nodes. This comparison is presented not to propose a new NAS method or search space, but rather to provide a conceptual and visual explanation of how solution vectors are transformed into cell structures under different NAS formulations.

C. Proposed Method

In this study, we propose a pairwise surrogate-based performance prediction framework to compare different predictor families on NAS outputs obtained from two NAS formulations: NP-NAS and PB-NAS. The method is evaluated on four whole-slide digital pathology datasets: EBHI, BCNB, TCGA, and SPIDER. The key idea is to learn predictors that estimate which of two candidate architectures is better (pairwise comparison), and then convert these pairwise predictions into a global ranking for top-architecture selection.

Step 1: Architecture generation via Random Search (RS) For each dataset and each NAS approach, we generate 200 candidate solutions using RS. An architecture encoding represents each solution (see Figure 1).

Step 2: Training candidate architectures and obtaining fitness Each solution is trained on a GPU for 20 epochs to determine its ground-truth performance. It is then assigned a scalar fitness score based on its F1-score.

Step 3: Pairwise dataset construction Instead of directly predicting an absolute performance value from a single architecture, we formulate surrogate learning as a pairwise classification problem. Let $\{x_i\}_{i=1}^{n}$ denote the set of feasible solutions and $\{f_i\}_{i=1}^{n}$ their fitness values (F1-scores). We construct ordered pairs (x_i, x_j) for all $i \neq j$. Each pair is assigned a binary label:

$$y_{ij} = \begin{cases} 1, & \text{if } f_i > f_j, \\ 0, & \text{otherwise.} \end{cases}$$

Thus, each training example consists of:

- **Input:** a pair of architectures $[x_i \,\|\, x_j]$,
- **Label:** $y_{ij} \in \{0, 1\}$.

where $\|$ denotes vector concatenation. For GNN-based predictors, each architecture is first converted into a graph structure before forming the pair.

Step 4: Train/test split The pairwise dataset is split into training and test sets with a $70/30$ ratio, using a fixed random seed for reproducibility.

Step 5: Tree-based surrogate predictors We train multiple tree-based predictors, including XGBoost, LightGBM, CatBoost, and Random Forest. The model learns to predict the probability that x_i outperforms x_j. We evaluate predictors from two perspectives: (1) **Binary classification performance** on the held-out test pairs using Accuracy, Precision, Recall, and F1-score. (2) **Ranking and selection performance** by aggregating predicted pairwise outcomes into a global ranking using a Borda-style method. From the resulting ranking, we compute ranking metrics such as Kendall's τ, Spearman's ρ, and NDCG@K (e.g., $K \in \{1, 3, 5\}$).

Step 6: GNN-based surrogate predictors To explore graph-based surrogate modeling, we additionally train GNNs on the same pairwise supervision. The main difference from tree-based predictors is the input representation: GNNs require graph-structured data. **Architecture-to-graph conversion:** Each architecture is converted into a graph $G = (V, E)$ using:

Authorized licensed use limited to: FATIH SULTAN MEHMET UNIVERSITY. Downloaded on November 11,2026 at 07:59:29 UTC from IEEE Xplore. Restrictions apply.

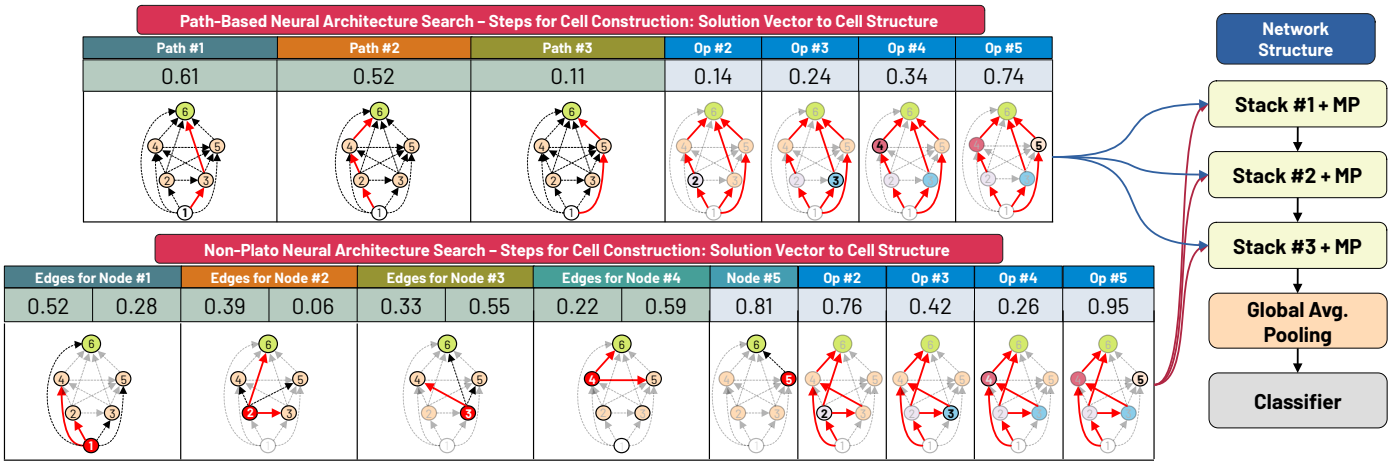


Fig. 1. Comparison of Cell Structure Formation Mechanisms between PB-NAS and NP-NAS. The upper section illustrates the PB-NAS approach, where the architecture is derived by selecting optimal paths and operations based on vector values. The middle section depicts the NP-NAS method, which constructs the cell by independently determining the connectivity (edges) for each node and selecting operations.

- an adjacency matrix to define edges,
- an operation list to define node types.

Evaluation criteria for GNN-based predictors: The GNN-based predictors are evaluated using the same protocol as tree-based predictors: Binary classification performance and Ranking and selection performance.

Overall, across both NAS approaches and all four datasets, our goal is to compare the predictive quality of: tree-based predictors operating on concatenated solution vectors, and GNN-based predictors operating on graph representations of architectures. We evaluate performance in terms of both classification success (pairwise decision accuracy) and ranking/selection success (ability to recover global ordering and identify top-performing architectures).

III. Experimental Design

A. Datasets

The EBHI dataset is an open-source collection of 5,532 Hematoxylin and Eosin (H&E) histopathology images of colorectal cancer captured via electron microscopy [9]. It includes five tumor differentiation stages, suitable for multi-class classification, and can be grouped into two categories for binary tasks. Images are available at 40, 100, 200, and 400x magnifications, and we use 200× and binary tasks. The BCNB dataset comprises 1,058 Wheel Slide Images (WSIs) of early-stage breast cancer from needle biopsies, with tumor areas annotated by experts [10]. The SPIDER dataset supports AI applications in pathology, containing patch-level images from multiple organs labeled for digital pathology [11]. This study uses the SPIDER-Colorectal subset, comprising 13 classes, 77,182 images at 20× magnification, and 224×224-pixel central patches. The dataset from The Cancer Genome Atlas (TCGA) includes molecular data and WSIs from over 20,000 primary tumors across 33 cancer types [12]. Focused on three anatomical sites with eight subtypes, the study analyzes 109 WSIs, chosen for their low computational cost and challenge. In our experiments, we used publicly available train and test splits for a fair comparison.

B. Evaluation Metrics

Ranking quality is measured by how well the surrogate-induced order aligns with the true architecture ranking based on actual fitness values. We use Kendall's tau and Spearman's rho to assess overall agreement: Kendall's tau evaluates the consistency of pairwise orderings, while Spearman's rho measures how closely the predicted ranks match the true ranks. To evaluate top-ranked architectures, we report NDCG@K, which emphasizes correct placement of high-performing architectures at the top and penalizes early mis-orderings. We also assess selection performance using Hit@K, which indicates whether the true best architecture is among the top K predictions, and regret, which quantifies the performance loss from choosing the surrogate's top-ranked architecture over the true best, reflecting surrogate errors in NAS.

IV. Result and Discussion

This study evaluates the performance of surrogate models using architectures generated by NP-NAS and PB-NAS, and tested with classification and ranking metrics. We evaluate surrogate quality from two perspectives: (i) pairwise classification success, i.e., how accurately a predictor decides whether one architecture outperforms another (Tables I and II), and (ii) ranking/selection success, i.e., how well the predictor reconstructs the global ordering and identifies top-performing architectures after aggregating pairwise decisions (Tables III and IV).

Tables I and II consistently show that tree-based surrogate predictors outperform the GNN-based surrogate in terms of Accuracy, Recall, Precision, and F1-score across both NAS encodings and all datasets. Among the tree models, LightGBM and CatBoost emerge as the strongest baselines, frequently achieving the best or second-best results, while Random Forest tends to lag behind the boosted methods.

Authorized licensed use limited to: FATIH SULTAN MEHMET UNIVERSITY. Downloaded on November 11,2026 at 07:59:29 UTC from IEEE Xplore. Restrictions apply.

TABLE I

COMPARISON OF THE CLASSIFICATION PERFORMANCE OF DIFFERENT PREDICTORS ON THE NON-PLATO ODE ACROSS FOUR DATASETS. FOR EACH DATASET AND METRIC, THE BEST RESULT IS HIGHLIGHTED IN **BOLD**, AND THE SECOND-BEST RESULT IS UNDERLINED.

	ACC	REC	PREC	F1
EBHI				
XGBoost	0.9645	0.9600	0.9681	0.9640
LightGBM	**0.9760**	**0.9703**	**0.9812**	**0.9757**
CatBoost	0.9661	0.9633	0.9681	0.9657
RandomForest	0.9163	0.8883	0.9396	0.9132
GNN	0.9347	0.9544	0.9172	0.9354
BCNB				
XGBoost	0.9678	0.9683	0.9671	0.9677
LightGBM	0.9783	0.9780	0.9785	0.9783
CatBoost	**0.9804**	**0.9799**	**0.9799**	**0.9804**
RandomForest	0.9524	0.9490	0.9554	0.9521
GNN	0.9187	0.9417	0.8999	0.9203
SPIDER				
XGBoost	0.9613	0.9618	0.9615	0.9617
LightGBM	0.9760	0.9756	0.9769	0.9763
CatBoost	**0.9785**	**0.9784**	**0.9789**	**0.9787**
RandomForest	0.9513	0.9491	0.9543	0.9517
GNN	0.9429	0.9534	0.9348	0.9440
TCGA				
XGBoost	0.9601	0.9558	0.9649	0.9604
LightGBM	**0.9790**	**0.9760**	0.9823	**0.9792**
CatBoost	0.9777	0.9732	**0.9826**	0.9779
RandomForest	0.9515	0.9424	0.9610	0.9516
GNN	0.9306	0.9287	0.9338	0.9312

TABLE II

COMPARISON OF THE CLASSIFICATION PERFORMANCE OF DIFFERENT PREDICTORS ON THE PATH-BASED ODE ACROSS FOUR DATASETS. FOR EACH DATASET AND METRIC, THE BEST RESULT IS HIGHLIGHTED IN **BOLD**, AND THE SECOND-BEST RESULT IS UNDERLINED.

	ACC	REC	PREC	F1
EBHI				
XGBoost	0.9434	0.9469	0.9396	0.9433
LightGBM	0.9779	0.9796	0.9760	0.9778
CatBoost	**0.9793**	**0.9810**	**0.9775**	**0.9792**
RandomForest	0.9301	0.9250	0.9337	0.9293
GNN	0.9350	0.9584	0.9149	0.9361
BCNB				
XGBoost	0.9534	0.9537	0.9524	0.9530
LightGBM	0.9775	0.9770	**0.9775**	0.9772
CatBoost	**0.9781**	**0.9785**	0.9774	**0.9779**
RandomForest	0.9317	0.9257	0.9357	0.9307
GNN	0.9419	0.9472	0.9361	0.9417
SPIDER				
XGBoost	0.9545	0.9523	0.9570	0.9549
LightGBM	0.9806	**0.9808**	0.9808	0.9808
CatBoost	**0.9807**	**0.9808**	**0.9811**	**0.9809**
RandomForest	0.9359	0.9326	0.9402	0.9364
GNN	0.9289	0.8979	0.9588	0.9274
TCGA				
XGBoost	0.9560	0.9545	0.9584	0.9565
LightGBM	**0.9786**	0.9790	**0.9787**	**0.9788**
CatBoost	0.9785	**0.9801**	0.9774	**0.9788**
RandomForest	0.9370	0.9288	0.9457	0.9372
GNN	0.9400	0.9421	0.9507	0.9464

In Table I, boosted-tree surrogates dominate the NP-NAS setting: LightGBM is best on EBHI and TCGA, with CatBoost close behind and XGBoost consistently strong. In contrast, the GNN underperforms (most notably on BCNB and TCGA), suggesting that graph conversion of the NP-NAS encoding (with independently determined edges) yields higher structural variability that is harder to learn than the original vector representation for tree ensembles. Table II shows the same hierarchy for PB-NAS, with LightGBM/CatBoost again near the ceiling. However, the GNN becomes comparatively more competitive, supporting the view that encoding choice affects surrogate learnability: the path-selection constraints in PB-NAS likely induce more regular graph structures, which better match GNN inductive biases than the more heterogeneous NP-NAS graphs.

While pairwise classification performance is necessary, NAS ultimately requires identifying the best architectures. Therefore, we evaluate whether predicted pairwise outcomes can be aggregated into meaningful global rankings and top-K selections (Tables III and IV).

In NP-NAS (Table III), LightGBM/CatBoost yield the most consistent rankings (Kendall's $\tau \approx 0.90$, Spearman's $\rho \approx 0.98$–0.99), with XGBoost close behind, confirming boosted trees as the strongest ranking baselines. These high correlations indicate that boosted-tree predictors are effective not only at making individual pairwise decisions but also at preserving the global ordering of architectures after aggregation. The GNN attains lower correlations (typically $\tau \approx 0.80$–0.85, and particularly weak on TCGA), indicating that pairwise errors amplify during aggregation. Notably, Hit@1/Hit@3 are often 0 despite high correlation, suggesting that recovering the very top architecture in NP-NAS is harder than approximating the overall order (e.g., due to tight fitness gaps among top candidates). In PB-NAS (Table IV), boosted trees again lead in τ/ρ, but top-K selection is substantially stronger: on SPIDER and TCGA several models achieve Hit@1/3/5 $= 1$ with regret $= 0$, implying reliable best-architecture identification under the more structured path-based search space. The GNN also improves (including perfect Hit@K and zero regret in some cases), supporting that path-based encodings produce graph structures better aligned with GNN inductive biases.

Comparing NP-NAS and PB-NAS, the results suggest that PB-NAS yields a more predictable and surrogate-friendly architecture manifold, enabling not only high pairwise accuracy but also reliable recovery of top architectures. In contrast, the NP-NAS space, where connections are independently determined, appears to pose a more challenging ranking problem: even strong pairwise predictors may struggle to capture the extreme top architectures after aggregation consistently. Overall, these results demonstrate that the proposed pairwise surrogate framework can successfully support NAS evaluation and top-architecture selection in computational pathology, with tree-based predictors providing the most robust and consistently strong performance, while GNN-based predictors become promising when paired with a path-based architectural encoding that yields more regular graph structures.

V. CONCLUSION

In this study, we investigated the effectiveness of different surrogate model families on architectures generated by two distinct NAS formulations, namely NP-NAS and PB-NAS, using a pairwise comparison framework. Experiments conducted

Authorized licensed use limited to: FATIH SULTAN MEHMET UNIVERSITY. Downloaded on November 11,2026 at 07:59:29 UTC from IEEE Xplore. Restrictions apply.

TABLE III

RANKING-BASED PERFORMANCE COMPARISON OF DIFFERENT PREDICTORS ON THE NON-PLATO ODE ACROSS FOUR DATASETS. WE REPORT KENDALL'S τ, SPEARMAN'S ρ, NDCG@3, HIT@1/3/5, AND REGRET (LOWER IS BETTER, ↓). FOR EACH DATASET AND METRIC, THE BEST RESULT IS HIGHLIGHTED IN **BOLD** AND THE SECOND-BEST RESULT IS UNDERLINED. XGB: XGBOOST; LGBM: LIGHTGBM; CB: CATBOOST; RF: RANDOM FOREST

	EBHI							BCNB						
	Kendall's τ	Spearman's ρ	NDCG@3	Hit@1	Hit@3	Hit@5	Regret (↓)	Kendall's τ	Spearman's ρ	NDCG@3	Hit@1	Hit@3	Hit@5	Regret (↓)
XGB	0.8896	0.9835	0.9542	0	0	**1**	**0.0230**	0.8808	0.9808	0.9683	0	0	0	0.331
LGBM	**0.8994**	**0.9860**	**0.9583**	0	0	**1**	**0.0230**	0.8923	**0.9844**	0.9692	0	0	0	0.331
CB	0.8890	0.9837	**0.9583**	0	0	**1**	**0.0230**	**0.8930**	0.9843	0.9692	0	0	0	0.331
RF	0.8703	0.9784	**0.9583**	0	0	0	**0.0230**	0.8705	0.9776	**0.9806**	0	0	0	**0.0331**
GNN	0.8346	0.9686	**0.9583**	0	0	0	0.0260	0.8033	0.9509	0.9692	0	0	0	**0.0331**
	SPIDER							TCGA						
XGB	0.8758	0.9790	0.9644	0	0	0	**0.0358**	0.8730	0.9785	**0.9641**	0	0	0	0.0334
LGBM	0.8894	0.9828	0.9687	0	0	0	0.0443	0.8889	0.9832	**0.9641**	0	0	0	0.0334
CB	**0.8908**	**0.9833**	0.9644	0	0	0	0.0443	**0.8925**	**0.9841**	**0.9641**	0	0	0	0.0334
RF	0.8721	0.9775	**0.9706**	0	0	0	0.0405	0.8675	0.9767	**0.9641**	0	0	0	**0.0156**
GNN	0.8488	0.9705	0.9673	0	0	0	0.0443	0.8317	0.9658	0.9457	0	0	0	0.412

TABLE IV

RANKING-BASED PERFORMANCE COMPARISON OF DIFFERENT PREDICTORS ON THE PATH-BASED ODE ACROSS FOUR DATASETS. WE REPORT KENDALL'S τ, SPEARMAN'S ρ, NDCG@3, HIT@1/3/5, AND REGRET (LOWER IS BETTER, ↓). FOR EACH DATASET AND METRIC, THE BEST RESULT IS HIGHLIGHTED IN **BOLD** AND THE SECOND-BEST RESULT IS UNDERLINED.

	EBHI							BCNB						
	Kendall's τ	Spearman's ρ	NDCG@3	Hit@1	Hit@3	Hit@5	Regret (↓)	Kendall's τ	Spearman's ρ	NDCG@3	Hit@1	Hit@3	Hit@5	Regret (↓)
XGB	0.8626	0.9761	0.9477	0	0	0	0.0518	0.8829	0.9814	0.9757	0	0	0	**0.0238**
LGBM	**0.8990**	**0.9856**	**0.9634**	0	0	0	0.0449	0.9049	**0.9871**	**0.9784**	0	0	0	**0.0238**
CB	0.8977	0.9854	0.9568	0	0	0	**0.0438**	**0.9053**	**0.9871**	0.9754	0	0	0	**0.0238**
RF	0.8472	0.9712	0.9431	0	0	0	0.0518	0.8586	0.9740	0.9757	0	0	0	**0.0238**
GNN	0.8442	0.9689	0.9494	0	0	0	0.0482	0.8524	0.9722	**0.9784**	0	0	0	0.0251
	SPIDER							TCGA						
XGB	0.8763	0.9789	0.9935	**1**	**1**	**1**	**0**	0.8733	0.9788	0.9820	0	**1**	**1**	0.0810
LGBM	0.8976	0.9853	**0.9939**	**1**	**1**	**1**	**0**	**0.8976**	**0.9853**	**0.9939**	0	**1**	**1**	0.0810
CB	**0.9034**	**0.9861**	0.9934	**1**	**1**	**1**	**0**	0.8941	0.9844	0.9820	**1**	**1**	**1**	**0**
RF	0.8606	0.9745	0.9935	**1**	**1**	**1**	**0**	0.8634	0.9754	0.9820	0	**1**	**1**	0.0810
GNN	0.8244	0.9609	0.9904	**1**	**1**	**1**	**0**	0.8511	0.9719	0.9820	**1**	**1**	**1**	**0**

on four digital pathology datasets demonstrated that tree-based surrogate models, particularly LightGBM and CatBoost, consistently achieved superior performance in both classification and ranking tasks, reaching up to 98% accuracy and F1-score on several datasets. Beyond classification accuracy, the quality of NAS evaluation further analyzed using ranking-based metrics, including Kendall's τ and Spearman's ρ, which quantify the consistency between predicted and ground-truth architecture orderings. The high correlation values obtained for tree-based surrogate models indicate that these approaches are effective not only in pairwise decision-making but also in preserving the global ranking of candidate architectures, which is critical for reliable top-architecture selection. While GNN-based surrogate models generally underperformed compared to tree-based approaches, they exhibited more stable and balanced behavior on PB-NAS architectures. This observation highlights the importance of architectural encoding strategies and suggests that path-based representations provide a more suitable structure for graph-based surrogate learning. Overall, the results show that tree-based models offer a strong and reliable baseline for pairwise surrogate learning in NAS, while graph-based models remain promising for more structured and complex architecture representations.

ACKNOWLEDGEMENT

This study was supported by the Scientific and Technological Research Council of Türkiye (TUBITAK) under the Grant number 124E493.

REFERENCES

[1] Z. Kuş *et al.*, "Pbc-nas: Neural architecture search for peripheral blood cells classification," in *2024 32nd Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2024, pp. 1–4.

[2] H. Xu *et al.*, "Detection of basal cell carcinoma in whole slide images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2023, pp. 263–272.

[3] Y. Xue and A. Zhu, "An effective surrogate-assisted rank method for evolutionary neural architecture search," *Applied Soft Computing*, vol. 167, p. 112392, 2024.

[4] W. Wen *et al.*, "Neural predictor for neural architecture search," in *European conference on computer vision*, 2020, pp. 660–676.

[5] Y. Zhou *et al.*, "Siamnas: Siamese surrogate model for dominance relation prediction in multi-objective neural architecture search," in *GECCO*, 2025, pp. 1309–1318.

[6] Z. K. Akçelik *et al.*, "Multi-objective surrogate-assisted neural architecture search for digital pathology: A tree-based regression approach," in *2025 ELECO*, 2025, pp. 1–5.

[7] Z. Kuş *et al.*, "Molinas: multi-objective lightweight neural architecture search for whole-slide multi-class blood cell segmentation," *Health Information Science and Systems*, vol. 13, no. 1, p. 78, 2025.

[8] S. Koç *et al.*, "Neural architecture search for histopathological image classification," in *2025 33rd Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2025, pp. 1–4.

[9] W. Hu *et al.*, "Ebhi: A new enteroscope biopsy histopathological h&e image dataset for image classification evaluation," *Physica Medica*, vol. 107, p. 102534, 2023.

[10] F. Xu *et al.*, "Predicting axillary lymph node metastasis in early breast cancer using deep learning on primary tumor biopsy slides," *Frontiers in oncology*, vol. 11, p. 759007, 2021.

[11] D. Nechaev *et al.*, "Spider: A comprehensive multi-organ supervised pathology dataset and baseline models," *arXiv preprint arXiv:2503.02876*, 2025.

[12] A. Riasatian *et al.*, "Fine-tuning and training of densenet for histopathology image representation using tcga diagnostic slides," *Medical image analysis*, vol. 70, p. 102032, 2021.

Authorized licensed use limited to: FATIH SULTAN MEHMET UNIVERSITY. Downloaded on November 11,2026 at 07:59:29 UTC from IEEE Xplore. Restrictions apply.