Efficient Graph Neural Architecture Search for Medical Imaging in Real-World Clinical Settings

Hadjer Benmeziane [1] **Abderaouf Gacem** [2] **Kaoutar El Maghraoui** [3] **Sarra Benmeziane** [4]

Abstract

Deploying deep learning in clinical settings requires balancing accuracy with limited computational resources. This is especially challenging in multitask medical imaging, where shared encoders reduce redundancy but task-specific heads remain memory-intensive. We propose Efficient Graph Neural Architecture Search (EGNAS), a gradient-based method that explores a graph-structured space to find compact, task-specific predictors. EGNAS jointly optimizes accuracy and model size using a Pareto-efficient strategy. Evaluated on six MedNIST tasks, it reduces head size by 2.1x on average without performance loss. We further validate EGNAS in a real-world deployment on a low-resource clinical laptop in Algeria, demonstrating its practical utility for resource-constrained healthcare.

1. Introduction

Deep learning has demonstrated remarkable success in medical imaging, enabling automated and accurate diagnosis across a wide range of clinical tasks (Antonelli et al., 2022). However, deploying these models in real-world healthcare settings—especially in resource-constrained environments—remains a significant challenge due to limitations in computational capacity, memory availability, and power consumption (Benmeziane et al., 2024; Isensee et al., 2021). One of the most practical strategies to reduce model size and enable on-device inference is the use of a *shared encoder* across multiple tasks (Mikhailov et al., 2023; Kiechle et al., 2024).

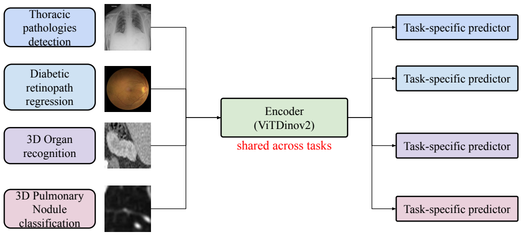

Figure 1. High-level model architecture used in this work. A shared encoder processes input medical images and feeds into multiple task-specific prediction heads.

This approach is commonly framed within the paradigm of *multi-task learning* (MTL) (Thung & Wee, 2018), where a single encoder processes the input data and captures a unified feature representation, which is then fed into task-specific heads. Sharing the encoder reduces redundant computation and improves training efficiency. Despite these advantages, the task-specific prediction heads, which are often implemented as multi-layer perceptrons (MLPs), can still be memory-intensive and pose a bottleneck for deployment on limited hardware. These heads must remain task-specific to preserve performance, which limits opportunities for further parameter sharing. An illustration of this architecture is shown in Figure 1.

To better illustrate this imbalance, Figure 2 shows the relative parameter share of the encoder versus the task-specific heads in our setup. While each head individually contributes only a small percentage of the total parameters, their cumulative cost becomes significant as the number of tasks grows. In fact, across all tasks, task-specific heads account for over 23% of the total model size, making them a key target for optimization.

To improve the representational power of these heads while maintaining task-specific behavior, recent works have proposed using *graph neural networks* (GNNs) (Kiechle et al., 2024) as an alternative to standard MLP predictors. GNNs offer a more structured and expressive way to model inter-feature dependencies and can adapt well to heterogeneous task requirements in a multi-task setup. However, designing effective and lightweight GNN architectures for task-specific prediction remains a complex challenge, especially under strict memory constraints.

In this work, we propose **Efficient Graph Neural Archi-**

[1] IBM Research Europe, 8803 Rüschlikon, Switzerland. [2] Institut National des Sciences Appliquées Lyon [3] IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA. [4] EPS Mohamed Boudiaf Ouargla. Correspondence to: Hadjer Benmeziane <hadjer.benmeziane@ibm.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

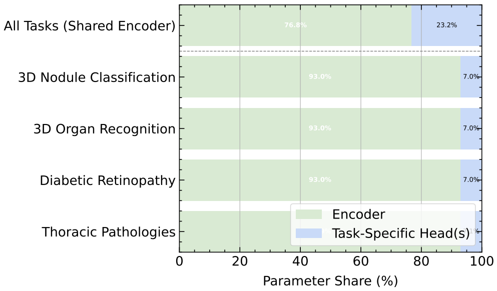

Figure 2. Relative parameter share between the shared encoder and task-specific heads across individual tasks and in aggregate. While each head is lightweight, their cumulative size becomes substantial as the number of tasks increases.

tecture Search (EGNAS), a method for discovering compact, task-specific GNN-based prediction heads tailored for multi-task medical imaging. EGNAS formulates architecture search as a graph-structured optimization problem and uses a Pareto-efficient, gradient-based strategy to jointly optimize for predictive accuracy and model size. The resulting architectures are well-suited for deployment in real-world, low-resource clinical environments.

We validate EGNAS on six medical imaging tasks from the MedNIST dataset (Yang et al., 2023) and demonstrate that it identifies lightweight task-specific GNN heads that maintain high accuracy while reducing memory footprint. We further illustrate the practical value of our approach through deployment on a low-resource clinical laptop in Algeria, highlighting its potential for scalable, high-performance medical AI in underserved regions.

2. Related Work

Neural Architecture Search (NAS) (Elsken et al., 2019; Baymurzina et al., 2022) is a technique for automatically discovering high-performing neural network architectures, often outperforming manually designed models in both accuracy and efficiency. In the context of medical imaging, NAS has been used to discover entire models tailored to specific clinical tasks, such as organ segmentation, tumor detection, and disease classification (Yang et al., 2024; Bargagna et al., 2024). These approaches typically search for encoder-decoder structures or end-to-end CNN architectures suited for full-volume or 2D slice-based medical inputs.

However, the overwhelming majority of prior work in medical NAS focuses solely on maximizing predictive performance. Given the life-critical nature of clinical decisions, these methods often optimize accuracy or AUROC exclusively, with little to no regard for computational efficiency. This narrow objective leads to models that are often too large for practical use in real-world, resource-constrained environments.

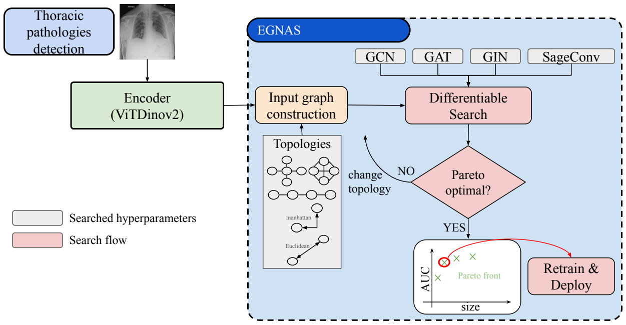

Figure 3. EGNAS framework high-level steps.

A recent study (Kiechle et al., 2024) explored the efficacy of Graph Neural Networks (GNNs) as alternatives to traditional Multi-Layer Perceptrons (MLPs) for classification tasks in 3D medical imaging. By constructing subject-level graphs from DINOv2-encoded slice representations, they demonstrated that GNNs can outperform MLPs in both accuracy and inference efficiency, highlighting the potential of graph-based models in medical applications. Inspired by (Kiechle et al., 2024), this is the first work to introduce a multi-objective neural architecture search tailored for medical imaging that explicitly considers both task performance and computational efficiency.

3. Methodology

Our goal is to discover lightweight, task-specific GNN-based prediction heads within a shared encoder framework for multi-task medical imaging. To this end, we propose EGNAS, a differentiable and Pareto-efficient architecture search framework (Liu et al., 2018) that optimizes prediction heads for both accuracy and compactness. An overview of our method is shown in Figure 3.

3.1. Problem Setup

We assume a multi-task learning setting with a shared encoder $f_\theta(\cdot)$ and T task-specific prediction heads $\{g_{\phi_t}(\cdot)\}_{t=1}^{T}$. Each task t has its own dataset $\mathcal{D}_t = \{(x_t^i, y_t^i)\}_{i=1}^{N_t}$, where x_t^i is a medical image and y_t^i is the corresponding label. The shared encoder extracts a latent representation $z = f_\theta(x)$, which is then passed to a task-specific head g_{ϕ_t} to produce an output $\hat{y}_t = g_{\phi_t}(z)$. Our objective is to learn compact, expressive architectures for each g_{ϕ_t} that maximizes task-specific performance while minimizing model size.

3.2. Search Space Design

Each task-specific prediction head is modeled as a GNN, and EGNAS performs joint optimization over three architectural dimensions: (1) the choice of message-passing operator, (2) the topological structure of the computation graph, and (3) key hyperparameters controlling depth and connectivity.

The operator space includes five options: Graph Convolutional Networks (GCN) (Zhang et al., 2019), Graph Attention Network (GAT) (Veličković et al., 2018), Graph Isomorphism Network (GIN) (Liu & Wang, 2021), GraphSAGE (Hamilton et al., 2017), and a baseline MLP; used as a fallback in cases where GNNs offer no performance benefit. Related work section B describes these GNN options. For each operator, we search over the number of layers $\in \{1, 2, 3\}$, hidden dimension size $\in \{32, 64, 128\}$, and whether to apply batch normalization and residual connections (each as binary decisions). Additionally, for GAT, we search over the number of attention heads $\in \{1, 4, 8\}$; for GIN, we search over the learnable aggregation function type.

The topology space defines how nodes are connected in the prediction head's computational graph. We support two classes of topologies: slice-based and encoding-based. Slice-based topologies include fully connected, line (1D chain), star, and custom manually defined structures; these reflect spatial priors such as anatomical slices or organ zones. Encoding-based topologies are derived from pairwise similarity metrics computed over latent representations, including Manhattan (L_1), Euclidean (L_2), Chebyshev (L_∞), and cosine similarity.

Each candidate architecture is represented as a directed acyclic graph (DAG) over hidden feature states, with edges encoding softmax-weighted mixtures of operations, parameterized by architecture weights α_t. Topology selection is treated as a categorical variableand is jointly optimized with operator choices and layer-wise parameters during the search phase. To support end-to-end search, topology selection is integrated using a Gumbel-softmax relaxation over discrete graph templates.

3.3. Pareto-Efficient Bi-Objective Optimization

EGNAS formulates the architecture search as a bi-objective optimization problem (Lampinen, 2000), balancing predictive accuracy and computational efficiency. For each task t, we define a composite loss function:

$$\mathcal{L}_t(\theta, \phi_t, \alpha_t) = \mathcal{L}_{\text{task}}(y_t, \hat{y}_t) + \lambda \cdot \mathcal{C}(\alpha_t), \tag{1}$$

where $\mathcal{L}_{\text{task}}$ is the task-specific prediction loss (e.g., cross-entropy or mean squared error), and $\mathcal{C}(\alpha_t)$ is a cost term that quantifies the expected *memory footprint* of the architecture induced by α_t. EGNAS explicitly incorporates both the number of parameters and the size of intermediate activations:

$$\mathcal{C}(\alpha_t) = \gamma \cdot \text{Params}(\alpha_t) + (1 - \gamma) \cdot \text{Activations}(\alpha_t), \tag{2}$$

where $\text{Params}(\alpha_t)$ is the total number of parameters in the architecture and $\text{Activations}(\alpha_t)$ estimates the total size of activations during inference, both computed as expectations over the soft architecture weights. The trade-off coefficient $\gamma \in [0, 1]$ controls the relative importance of model size versus runtime memory usage.

To estimate activation size, we consider the dimensions of hidden states generated at each node in the DAG, scaled by batch size and feature width.

We approximate the Pareto front by maintaining a pool of non-dominated architectures during search. This pool spans a range of accuracy-efficiency trade-offs, enabling informed selection of architectures suitable for deployment in resource-limited environments. This is reflected in Figure 3, where the condition evaluates whether the current architecture is Pareto-optimal with respect to accuracy and model size. If the current candidate is non-dominated, it is added to the Pareto front; otherwise, the search controller modifies the topology and re-enters the search loop. This iterative mechanism enables EGNAS to explore diverse graph structures while maintaining a strong trade-off between performance and resource efficiency.

3.4. Search Algorithm

The EGNAS algorithm proceeds in two phases: (1) *Joint Search Phase.* We fix the shared encoder f_θ and jointly optimize architecture weights α_t and predictor parameters ϕ_t for each task. Using Gumbel-Softmax relaxation, we enable differentiable sampling of both operator types and edge connectivity, allowing end-to-end optimization of topology and operation choice. (2) *Discrete Evaluation Phase.* After convergence, we discretize the architecture for each task head by selecting the highest-probability operation and connectivity per edge. The resulting GNN architecture is then re-instantiated and retrained from scratch to ensure a clean evaluation of its performance.

This two-phase pipeline allows EGNAS to effectively explore a large and expressive space of GNN architectures, producing accurate and compact task-specific predictors suitable for deployment in resource-constrained clinical environments.

4. Experiments

4.1. Dataset and Tasks

We evaluate EGNAS on the publicly available **MedNIST** dataset, which consists of over 50,000 labeled medical images from six categories: Chest X-ray, Hand, HeadCT, AbdomenCT, BreastMRI, and CXR. Each category corresponds to a distinct classification task, forming a six-task multi-task learning setup. We follow standard preprocessing

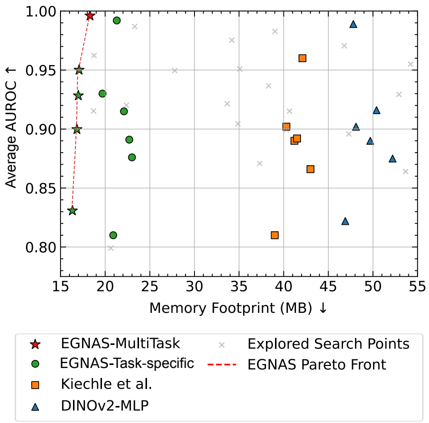

Figure 4. Task-wise Pareto front analysis on MedNIST3D. Each point shows accuracy vs. memory usage for a model on a given task. EGNAS models occupy or closely approach the Pareto front, demonstrating superior performance-efficiency trade-offs.

protocols, resizing images to 64×64 pixels and normalizing intensities. Data is split into 70% training, 15% validation, and 15% test sets for each task, ensuring no patient overlap.

4.2. Search and Training Procedure

We run EGNAS for 50 epochs of joint architecture and weight optimization using Adam with learning rate 10^{-3}. Architecture parameters α_t are updated every 5 steps using a held-out validation set.

After convergence, we discretize each task head by selecting the top-performing configuration on the Pareto front (maximizing accuracy under a memory threshold). Final GNN architectures are retrained from scratch using full training data for 100 epochs with early stopping.

4.3. Results

We compare EGNAS against two strong baselines on the MedNIST3D benchmark: a shared encoder with standard MLP heads (**DINOv2-MLP**) and the GNN-based architecture of Kiechle et al. (2024). Figure 4 visualizes the performance-efficiency trade-offs across six medical imaging tasks, plotting accuracy versus memory footprint for each method. Full numerical results are included in the supplementary material (Table S1).

EGNAS achieves state-of-the-art accuracy on all tasks, while significantly reducing the runtime and memory footprint. In each subplot, EGNAS configurations lie on or near the Pareto frontier, outperforming baselines in both accuracy and memory usage. On average, EGNAS reduces head memory usage by 2.1× compared to DINOv2-MLP and by 1.9× compared to Kiechle et al., without sacrificing predictive quality.

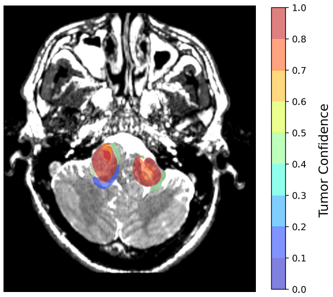

Figure 5. Tumor confidence overlay on T1-weighted axial MRI. EGNAS output matched expert labels with 78% IoU.

EGNAS also offers consistently faster inference, with an average runtime reduction of 54% compared to DINOv2-MLP and 45% compared to Kiechle et al. These savings stem from EGNAS's ability to tailor architectural complexity to each task's needs via Pareto-efficient search.

In addition, we evaluate a multi-task setting with a shared encoder and all six EGNAS heads (**EGNAS-MultiTask**). This configuration delivers high aggregate performance (AUROC = 0.996, ACC = 0.939), while maintaining a total memory footprint of just 18.3 MB, making it ideal for deployment in memory-constrained environments.

Conclusion & Clinical Use

We introduced EGNAS, a Pareto-efficient neural architecture search method that discovers compact, task-specific GNN heads for multi-task medical imaging. EGNAS balances accuracy and memory efficiency through a differentiable, graph-structured search space and demonstrates strong performance across six tasks on the MedNIST dataset. It reduces head memory usage by over 2× without sacrificing accuracy and achieves fast inference, making it ideal for deployment in low-resource settings.

Furthermore, we validated EGNAS in collaboration with clinicians at an Algerian ENT department on a brain MRI case involving bilateral vestibular schwannomas. The patient presented with cranial nerve symptoms, and imaging revealed enhancing extra-axial masses in the cerebellopontine angles (30mm right, 38mm left), extending into the internal auditory canals and displacing the brainstem.

To support diagnosis under hardware constraints (no GPU, limited RAM), EGNAS optimized compact GNN-based models for brain tumor detection on T1-weighted, contrast-enhanced sequences. The final model operated within a 300 MB memory budget and achieved inference times under 1.5 seconds/image on a dual-core Intel i5 laptop.

Predicted tumor regions showed 78% IoU with expert annotations, confirming diagnostic reliability. Figure 5 shows a representative segmentation with confidence overlay.

References

Antonelli, M., Reinke, A., Bakas, S., Farahani, K., Kopp-Schneider, A., Landman, B. A., Litjens, G., Menze, B., Ronneberger, O., Summers, R. M., et al. The medical segmentation decathlon. *Nature communications*, 13(1): 4128, 2022.

Bargagna, F., Zigrino, D., De Santi, L. A., Genovesi, D., Scipioni, M., Favilli, B., Vergaro, G., Emdin, M., Giorgetti, A., Positano, V., et al. Automated neural architecture search for cardiac amyloidosis classification from [18f]-florbetaben pet images. *Journal of Imaging Informatics in Medicine*, pp. 1–15, 2024.

Baymurzina, D., Golikov, E., and Burtsev, M. A review of neural architecture search. *Neurocomputing*, 474:82–93, 2022.

Benmeziane, H., Hamzaoui, I., Cherif, Z., and El Maghraoui, K. Medical neural architecture search: Survey and taxonomy. In *International Joint Conference on Artificial Intelligence*, 2024.

Elsken, T., Metzen, J. H., and Hutter, F. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 1024–1034, 2017.

Isensee, F., Jaeger, P. F., Kohl, S. A., Petersen, J., and Maier-Hein, K. H. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

Kiechle, J., Lang, D. M., Fischer, S. M., Felsner, L., Peeken, J. C., and Schnabel, J. A. Graph neural networks: A suitable alternative to mlps in latent 3d medical image classification? In *International Workshop on Graphs in Biomedical Image Analysis*, pp. 12–22. Springer, 2024.

Lampinen, J. Multiobjective nonlinear pareto-optimization. *Pre-investigation Report, Lappeenranta University of Technology*, 114:125, 2000.

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Liu, J. and Wang, H. Graph isomorphism network for speech emotion recognition. In *Interspeech*, pp. 3405–3409, 2021.

Mikhailov, I., Chauveau, B., Bourdel, N., and Bartoli, A. Sharing is caring: Concurrent interactive segmentation and model training using a joint model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2432–2441, 2023.

Thung, K.-H. and Wee, C.-Y. A brief review on multi-task learning. *Multimedia Tools and Applications*, 77(22): 29705–29725, 2018.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., and Ni, B. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Yang, Y., Wei, J., Yu, Z., and Zhang, R. A trustworthy neural architecture search framework for pneumonia image classification utilizing blockchain technology. *The Journal of Supercomputing*, 80(2):1694–1727, 2024.

Zhang, S., Tong, H., Xu, J., and Maciejewski, R. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.

A. Full result

This table show the full result of task-specific EGNAS search and multi-task, including AUROC, accuracy, memory footprint and runtime. The runtime computes the time to run full inference on each dataset validation sets.

Table 1. Comparison of EGNAS against state-of-the-art models on MedNIST3D tasks. Best results per row in **bold**.

Dataset	**Model**	**AUROC ↑**	**ACC ↑**	Runtime (min)	Memory (MB) ↓
OrganMNIST3D	DINOv2-MLP	0.997 ± 0.001	0.933 ± 0.002	3.9	47.8
	Kiechle et al. (2024)	0.991	0.932	3.0	42.1
	EGNAS	**0.997 ± 0.001**	**0.943 ± 0.003**	**1.6**	**21.3**
NoduleMNIST3D	DINOv2-MLP	0.905 ± 0.001	0.866 ± 0.003	3.8	48.1
	Kiechle et al. (2024)	0.912	0.869	2.8	40.3
	EGNAS	**0.918 ± 0.004**	**0.876 ± 0.004**	**1.4**	**19.7**
FractureMNIST3D	DINOv2-MLP	0.812 ± 0.003	0.641 ± 0.004	3.6	46.9
	Kiechle et al. (2024)	0.801	0.611	3.1	39.0
	EGNAS	**0.819 ± 0.009**	**0.637 ± 0.011**	**1.5**	**20.9**
AdrenalMNIST3D	DINOv2-MLP	0.926 ± 0.003	0.870 ± 0.004	3.9	50.4
	Kiechle et al. (2024)	0.902	0.851	3.3	41.2
	EGNAS	**0.938 ± 0.002**	**0.883 ± 0.011**	**1.7**	**22.1**
SynapseMNIST3D	DINOv2-MLP	**0.885 ± 0.001**	**0.873 ± 0.003**	4.1	52.2
	Kiechle et al. (2024)	0.871	0.855	3.5	43.0
	EGNAS	0.892 ± 0.004	0.868 ± 0.004	**1.6**	**23.0**
VesselMNIST3D	DINOv2-MLP	0.909 ± 0.001	0.899 ± 0.005	4.2	49.7
	Kiechle et al. (2024)	0.905	0.889	3.7	41.5
	EGNAS	**0.918 ± 0.002**	**0.901 ± 0.004**	**2.0**	**22.7**
Multiple	**EGNAS-MultiTask (Ours)**	0.996	0.939	**1.6**	**18.3**

B. Graph Neural Networks

GNNs are a class of neural networks specifically designed to operate on graph-structured data, making them well-suited for tasks where relational or topological information is essential, such as tumor detection in medical imaging and patient data.

Graph Convolutional Network (GCN) (Zhang et al., 2019), while popular, was not the first attempt to build deep learning models for graphs. GCN simplified earlier spectral methods into more scalable and intuitive operations. It introduced a message-passing framework, where each node updates its representation by collecting information from its neighbors. This idea opened up a new direction in graph learning.

Building on this foundation, GraphSAGE (Hamilton et al., 2017) introduced neighborhood sampling to improve scalability and enable inductive learning, while Graph Attention Networks (GATs) (Veličković et al., 2018) enhanced the aggregation process by learning attention weights for different neighbors, allowing the model to focus on more relevant connections.

Further advancing the field, the Graph Isomorphism Network (GIN) (Liu & Wang, 2021) focused on the expressive power of GNNs, replacing simple averaging or attention aggregation with a multi-layer perceptron (MLP), which gives the model more flexibility in combining neighbor information and makes it as powerful as the Weisfeiler-Lehman (WL) graph isomorphism test.