npj | digital medicine

Article

Published in partnership with Seoul National University Bundang Hospital

https://doi.org/10.1038/s41746-025-02042-x

Large language models driven neural architecture search for universal and lightweight disease diagnosis on histopathology slide images

Xiu Su[1,2], Qinghua Mao[3], Zhongze Wu[1,2], Xi Lin[3] ✉, Shan You[4], Yue Liao[5] & Chang Xu[6]

Artificial Intelligence has revolutionized healthcare by offering smart services and reducing diagnostic burden, particularly facilitating the identification and segmentation of malignant tissues. However, current task-specific approaches require disease-specific models, while universal foundation models demand costly customization for complex cases, hindering practical deployment in clinical environments. We present Pathology-NAS, a universal and lightweight medical analysis framework that leverages LLMs' knowledge to refine the architecture space across diverse scenarios, eliminating the need for exhaustive search. Pathology-NAS is pretrained on 1.3 million images across three supernet architectures, providing a robust visual foundation that generalizes across diverse tasks. Across breast cancer and diabetic retinopathy diagnosis tasks, Pathology-NAS achieves 99.98% classification accuracy while reducing FLOPs by 45% compared to leading methods. Our model delivers near-optimal architectures in just 10 iterations, bypassing the exponential search space. Pathology-NAS provides accurate tumor recognition across diverse tissues with computational efficiency, making AI-assisted diagnosis practical even in resource-constrained clinical environments.

Cancer remains one of the most formidable challenges in contemporary medicine, with its diverse manifestations and multifaceted etiologies. Histopathology image analysis is crucial in risk prevention and crisis management of aggressive tumor progression, commonly characterized by rapid growth, dynamic molecular changes and high metastatic potential across different organs. Historically, the cancer prediction and prognosis have relied heavily on morphological criteria and tissue-based examinations. However, these methods are time-consuming and depend on expertise knowledge and specific behaviors of certain tumors. According to a recent report from the GLOBOCAN[1], over the past 10 years, more than 200 types of all recorded cancers, 18% of related deaths (about 95 million) and 2.3–2.5% of related global economic losses (US$ 20 trillion) were consequences of malignant tumors or their complications.

Artificial intelligence (AI), emerging as a revolutionary technology, has unleashed the potential to provide automated and intelligent solutions for medical imaging analysis. Histopathology slides provide high-resolution tissue observations with rich cellular and morphological information, making them ideal candidates for AI analysis[2]. The conventional approach for analyzing these data is through task-specific deep learning models, which generate diagnostic predictions based on supervised learning from annotated images[3]. However, these methods, even when implemented with state-of-the-art architectures, restrict adaptability to specific cancer types and anatomical regions[4,5], requiring separate models for each diagnostic scenario (Fig. 1a). Alternative approaches include CeoGraph[3], a cell-graph model attempts to address this limitation by modeling cellular spatial organization. These methods identify key morphological features by correlating spatial patterns with patient symptoms, but depend heavily on precise computational staining and cell location detection. They provide valuable insights for specific cancer types but fail to generalize across the spectrum of pathology images due to heavy cost[5].

Large language models (LLMs) and vision foundation models[6,7] have been widely applied in recent years to pathology analysis[4,8]. These methods leverage large corpora of diverse medical images to train generalizable models in an end-to-end fashion, dispensing with task-specific

[1]Big Data Institute, Central South University, Changsha, China. [2]National Engineering Research Center for Medical Big Data Application Technology, Changsha, China. [3]School of Computer Science, Shanghai Jiao Tong University, Shanghai, China. [4]SenseTime Research, Science Park, Hong Kong. [5]Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, China. [6]School of Computer Science, Faculty of Engineering, The University of Sydney, Sydney, NSW, Australia. ✉e-mail: linxi234@sjtu.edu.cn

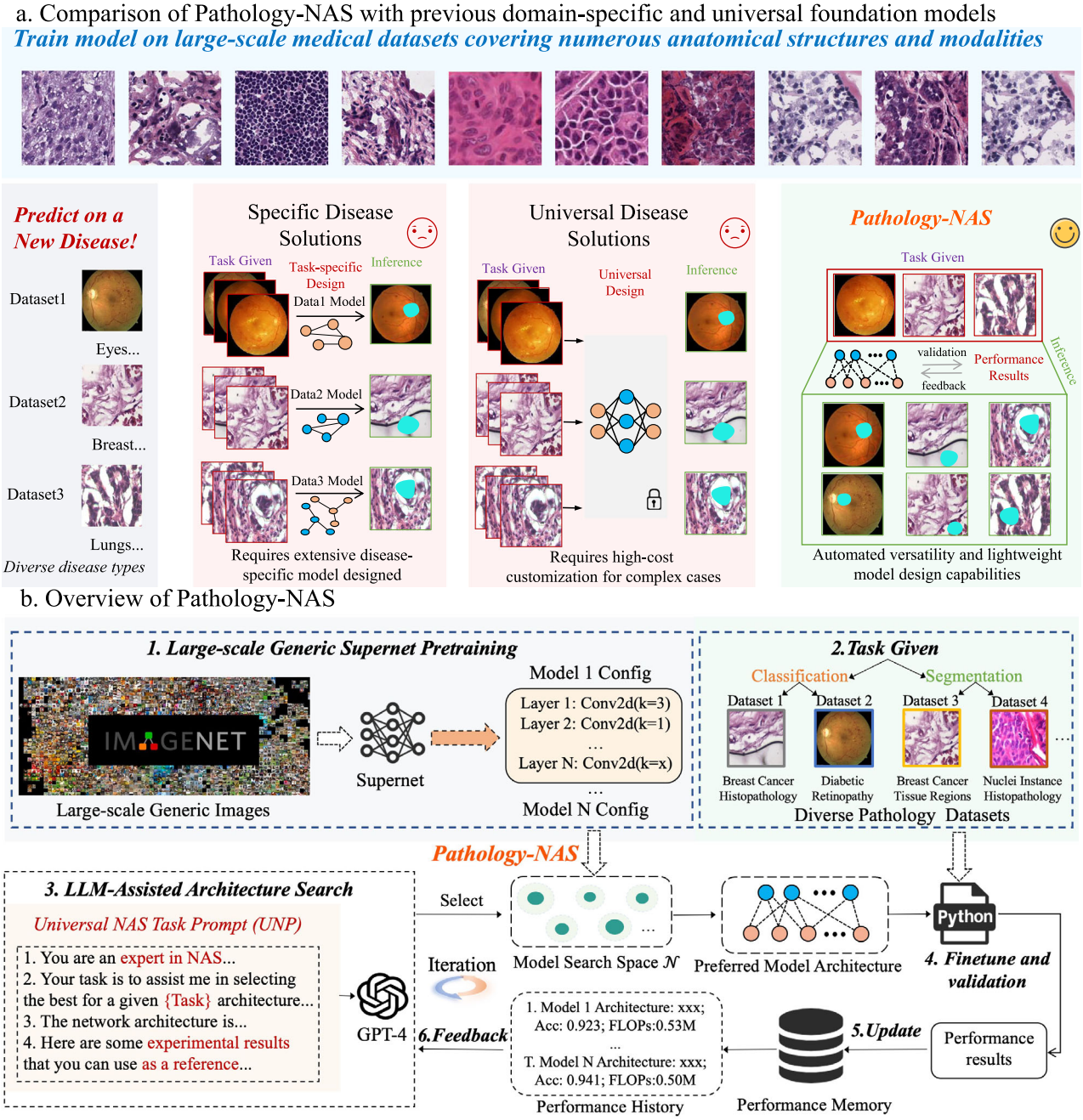


Fig. 1 | The primary strength and overall framework of Pathology-NAS. **a** Pathology NAS is trained on a large-scale corpus of medical image datasets. PrevIoUs specific disease solutions require to design customized model for each disease, lacking enough generalization capabilities. While universal foundation models are trained on data covering numerous anatomical structures, high-cost customization for complex cases is still required. Compared with disease-specific methods and existing universal foundation models, Pathology-NAS holds advantages in automated versatility and lightweight model design. **b** Overview of Pathology-NAS, a universally medical image analysis framework driven by LLM-assisted neural architecture search. Pathology-NAS significantly benefits from large-scale generic supernet pretraining, neural architecture fine-tuning and validation on diverse pathology datasets, and LLM-assisted architecture search.

architectures. They have proved promising for universal medical image analysis as measured by cross-domain generalization metrics[9]. A significant advance in this direction has been the Segment Anything Model (SAM) and its medical adaptation MedSAM[4], which generates prompt-based segmentations adaptable to diverse medical contexts. In evaluations across multiple datasets, MedSAM demonstrated superior versatility compared to task-specific models[10]. However, for complex pathological analysis, foundation models face significant challenges in practical deployment, particularly due to the need for high-cost customization to address diverse and intricate cases. This customization, coupled with immense web-scale data requirements and billions of model training parameters, demands substantial computational resources, energy, and storage[11,12], limiting their deployment into clinical practice. Therefore, it is of great urgency and importance to develop a universal and lightweight pathology image analysis solution tailored for clinical practice.

In this work, we present Pathology-NAS, a universally lightweight medical image analysis framework (Fig. 1b) that accurately identifies malignant tissue sections and predicts tumor categories across diverse histopathology slide images. Pathology-NAS aims to automatically optimize model design across diverse diagnostic tasks, achieving 99.98% classification accuracy in breast cancer and diabetic retinopathy diagnosis while reducing computational cost by 45% compared to leading methods. Pretrained on

Table 1 | Image classification performance compared with SOTA models

Models	BreakHis			Diabetic		
	Prec@1 (%)↑	FLOPs↓	Params (M)↓	Prec@1 (%)↑	FLOPs↓	Params (M)↓
EfficientNet	88.63	384.60M	3.97	69.52	384.61M	3.97
ResNet	95.10	4.13G	23.51	70.48	4.13G	23.52
Pathology-NAS ShuffleNet(ours)	**99.98**	**213.30M**	**1.80**	**73.22**	**240.25M**	**2.10**
ViT-small	87.33	**4.25G**	25.19	67.71	4.25G	21.59
Swin-Transformer	83.59	15.17G	86.68	54.69	15.17G	86.68
Pathology-NAS ViT(ours)	**98.08**	4.95G	**25.12**	**70.38**	**4.13G**	**20.99**

For ShuffleNet backbone, Pathology-NAS is compared with EfficientNet and ResNet in terms of Top-1 accuracy, FLOPs and Params. For ViT backbone, Pathology-NAS is compared with ViT-small and Swin-Transformer in terms of Top-1 accuracy, FLOPs and Params. The best performance is highlighted in bold. Pathology-NAS achieves the highest performance with the lowest FLOPs and parameters.

Table 2 | Image segmentation performance compared with SOTA models

Search strategy	BCSS			PanNuke				
	Dice (%)↑	IoU (%)↑	FLOPs (G)↓	Params (M)↓	Dice (%)↑	IoU (%)↑	FLOPs (G)↓	Params (M)↓
U-Net	71.56	56.33	14.80	18.44	84.93	74.99	14.80	18.44
FPN	72.30	57.21	17.00	11.49	88.45	80.07	17.00	11.49
Pathology-NAS U-Net(ours)	**74.33**	**59.68**	**10.58**	**11.37**	**89.24**	**81.25**	**14.33**	**8.34**

For U-Net backbone, Pathology-NAS is compared with U-Net and FPN in terms of dice score, IoU score, FLOPs and Params. The best performance is highlighted in bold. Among all methods, Pathology-NAS holds the optimal segmentation performance with the lowest FLOPs and parameters.

large-scale generic images via a supernet, Pathology-NAS identifies near-optimal architectures in just 10 iterations guided by insights from large language models (LLMs), bypassing the 4^{20} configurations of traditional exhaustive search. Deployable in resource-constrained clinical settings, it recognizes tumors across varied tissues, which is a promising solution to enhance AI-driven cancer prediction and prognosis worldwide.

Results

Pathology-NAS: a universally lightweight medical image analysis framework

Pathology-NAS is designed as a versatile and lightweight medical image analysis framework, leveraging Large Language Model (LLM)-driven neural architecture search (NAS) to achieve optimal performance across diverse pathology datasets, as presented in Fig. 1. The primary goal of Pathology-NAS is to address the limitations of existing deep learning models that are often tailored to specific tasks, lacking generalization across different pathology types and imaging modalities.

To achieve this, we employ a novel one-shot NAS strategy guided by insights from LLMs, specifically utilizing GPT-4 to drive the architecture search process. The framework begins by pretraining a Supernet on a large-scale pathology image dataset, capturing a wide range of visual features. Each subnet path is uniformly sampled during this phase to ensure a fair estimation of its performance. After pre-training, GPT-4 assists in identifying the optimal architecture by iteratively refining the model configurations based on performance feedback.

Through extensive evaluations, Pathology-NAS has demonstrated superior performance in both image classification and segmentation tasks, as presented in Tables 1 and 2. Compared with competitive baselines, Pathology-NAS achieves 99.98% classification accuracy while reducing FLOPs by 45% across breast cancer and diabetic retinopathy diagnosis tasks. This approach significantly reduces the computational burden typically associated with exhaustive search strategies while enhancing the adaptability of the model to various pathology tasks. The lightweight nature of Pathology-NAS ensures that it can be effectively deployed in practical clinical environments where computational resources may be limited.

Qualitative and quantitative analysis

We evaluated Pathology-NAS on medical image recognition and segmentation tasks across four histopathology slide datasets. Particularly, we compare our approach with specialized SOTA classification models ResNet[13], EfficientNet[14] and Swin-Transformer[15], as well as segmentation models U-Net[16] and FPN[17]. For convolution-based models, each model was trained from scratch on a modality-wise image dataset, such as diabetic retinopathy dataset. The training protocol follows the same setting in Pathology-NAS approaches. During inference, these specialized models were leveraged to conduct cancer diagnosis and malignant tissue segmentation. For convolution-based classification and segmentation models, we trained them from scratch on pathology datasets. For ViT classification models Swin-Transformer, we finetune it on pathology datasets, while the pretrained weights were downloaded from the official implementation of timm library (https://huggingface.co/docs/hub/timm).

Tables 1 and 2 shows the overall performance of Pathology-NAS against SOTA classification and segmentation models, respectively. As shown in 1, our method have demonstrated superior performance under the constraint of model complexity. With the optimal architectures searched by GPT-4, ShuffleNet achieves the 99.98% top-1 accuracy on BreakHis and the highest 73.22% top-1 accuracy on diabetic datasets. ViT achieves the 98.08% top-1 accuracy on BreakHis and the highest 70.38% top-1 accuracy on diabetic datasets. Meanwhile, the searched models holds relatively lower FLOPs and model parameters. Note that swin-transformer shows extraordinary performance after finetuning, due to the rich inherent knowledge from pretraining on large-scale datasets. However, the optimal architectures in our method were retrained from scratch on downstream datasets, further showcasing excellent adaptability and performance through NAS. Similarly, the searched U-net architecture have achieved the best performance in terms of dice coefficient and IoU score, which might be attributed to the capacity of capturing multi-scale context and searching the optimal architectures. A suite of visualized segmentation examples are presented in Fig. 2, which showcases that Our segmentation model achieved results closely resembling the ground truth segmentation masks, surpassing the other two methods.

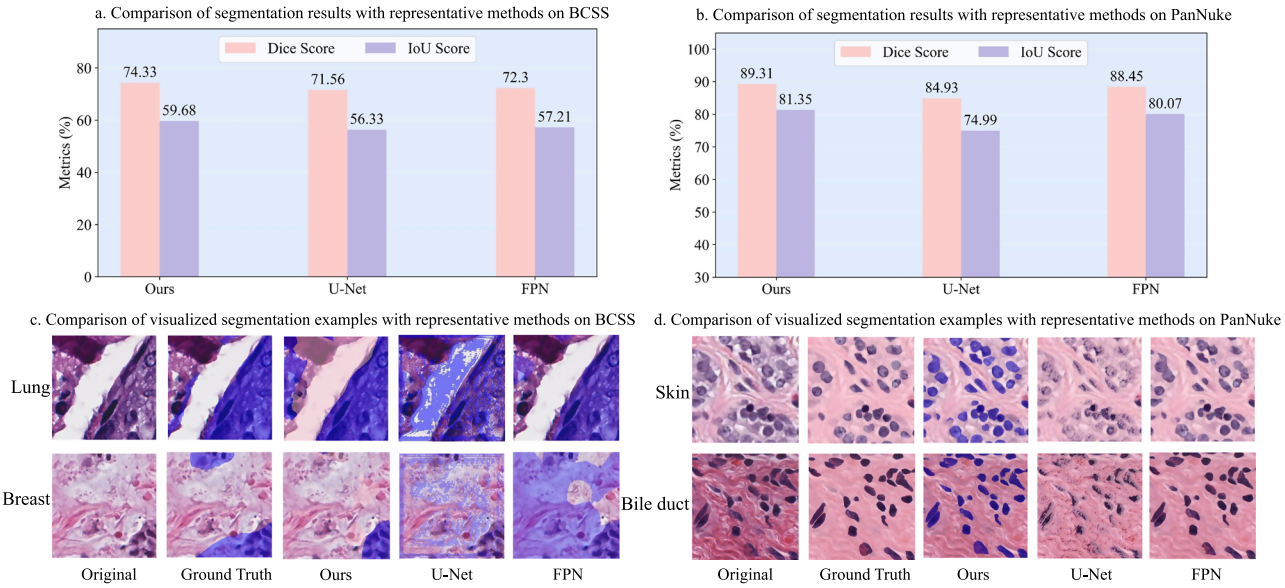


Fig. 2 | Quantitative and qualitative segmentation examples on BCSS and PanNuke segmentation datasets. a Comparison of segmentation results with U-Net and FPN on BCSS. **b** Comparison of segmentation results with U-Net and FPN on PanNuke. Our Pathology-NAS outperforms other two representative methods on both datasets. **c** Comparison of visualized segmentation examples on BCSS, including lung and breast tissues. **d** Comparison of visualized segmentation examples on PanNuke, including skin and bile duct tissues. It can be observed from segmentation examples that Pathology-NAS most closely resemble the ground truth, accurately delineating the distribution of different anatomical structures. Our method not only fully covers the target regions that need to be labeled but also meticulously distinguishes the boundaries between the target regions and the background regions.

The quantitative results on the PanNuke dataset across its 16 diverse tissue types, demonstrating that Pathology-NAS effectively supports a wide range of tissues by discovering superior and efficient architectures. Compared to Random Search (Supplementary Table 3), Pathology-NAS identifies more efficient architectures (i.e., reducing FLOPs by approx. 19.1%) while achieving superior segmentation, such as in Bile duct tissue (Dice: 88.53% vs. 81.71%). Furthermore, against non-searched baselines like U-Net and FPN (Supplementary Table 4), Pathology-NAS demonstrates improved performance with significantly fewer resources; for instance, it achieved a 5.4% Dice improvement over U-Net on Bile duct tissue with 12.1% fewer FLOPs, and a 71.5% FLOPs reduction compared to FPN while maintaining better performance

As shown in Fig. 3, the qualitative visual results presented further substantiate the effectiveness of Pathology-NAS. As highlighted by the red boxes in the visualizations, both U-Net and FPN exhibit common failure modes, including inaccurate delineation in Bile duct and Thyroid tissues, missed segmentations in Stomach and HeadNeck samples, and excessive adhesion between targets in Ovarian and Skin tissues. Pathology-NAS successfully mitigates these failure modes, demonstrating more robust and accurate segmentation.

Pathology-NAS shows competitive performance with higher computational efficiency

One-shot NAS approaches are a set of NAS strategies that leverages weight sharing for architecture search in discrete space. The weight sharing strategy is of great significance to avoid training each subnet from scratch. However, the search process is still exhausting. For example, NSE-NAS[18] conducts architecture search within 27 OPs and 5 layers, the total number of possible architectures $N_{arch} \approx 1.4 \times 10^{110}$. The most obvious distinction between previous one-shot approaches and our approaches is the architecture search process driven by GPT-4. We resort to inherent knowledge of large language models for selecting network candidates, not other manual search strategy. In this experiment, we compare the retraining performance of searched architectures obtained from our approaches and representative one-shot NAS approaches, including evolutionary search in SPOS[19], AutoFormer[20] and Cream[21]. The implementation of evolutionary search in SPOS and AutoFormer follow the same setting. The population size is set to 50, while the number of generations is set to 20. Each generation we pick the top 10 architectures as the parents to generate alternative subsets by mutation and crossover. The mutation probability P_d and P_m are set to 0.2 and 0.4 as in AutoFormer. The search space in Cream includes mobile inverted bottleneck MBConv[22] and squeeze-and-excitation networks[23]. The experimental setup is consistent with the original paper, including training and search process of hypernetwork and subnetwork retraining.

We provided a detailed breakdown of the computational costs, including the NAS time (GPU hours), total time (GPU hours), GPT-4 API calls (which occur only during the NAS phase), search iterations, model latency (hours), and API cost ($) for these respective tasks. As shown in Tables 3–5, our results show that Pathology-NAS achieves significant search efficiency improvements, requiring 3–9 × less search time while discovering architectures with superior performance compared to traditional methods. As seen from the Total Time (TT) in Table 3, the GPT-4 search latency is negligible.

For instance, for ShuffleNet on BreakHis (Table 3), Pathology-NAS identifies superior architectures in 10 iterations (search time 7.42 GPU hrs), achieving 99.98% ± 0.27 Top-1 accuracy, incurring a total latency of 0.001 over 10 calls and a total API cost of $0.13. This is achieved with 213.30M FLOPs for the final model's inference, which does not involve GPT-4. In comparison, other NAS methods (i.e., Random Search, Cream, AutoFormer) detailed in Tables 3–5 often require more iterations (i.e., Random Search: 500; Cream: 120; AutoFormer: 300) and longer search times (see tables for details) for comparable or lower accuracies. Furthermore, we also conducted a comprehensive cost analysis for the Zenodo lung cancer dataset (Table 5). Similarly, we performed a cost analysis for the Gastric Cancer dataset, with results presented in Table 4. Our results demonstrate that Pathology-NAS effectively discovers superior architectures while maintaining computational efficiency.

LLM response with controllable sampling helps stabilize performance improvement

Recent advanced alignment techniques, such as Reinforcement Learning with Human Feedback (RLHF)[24] and Direct Preference Optimization (DPO)[25] algorithms, have been adopted to generate controllable and adjustable content. In essence, these operations are to selectively utilize LLM knowledge at different level and scale by setting various parameters. The

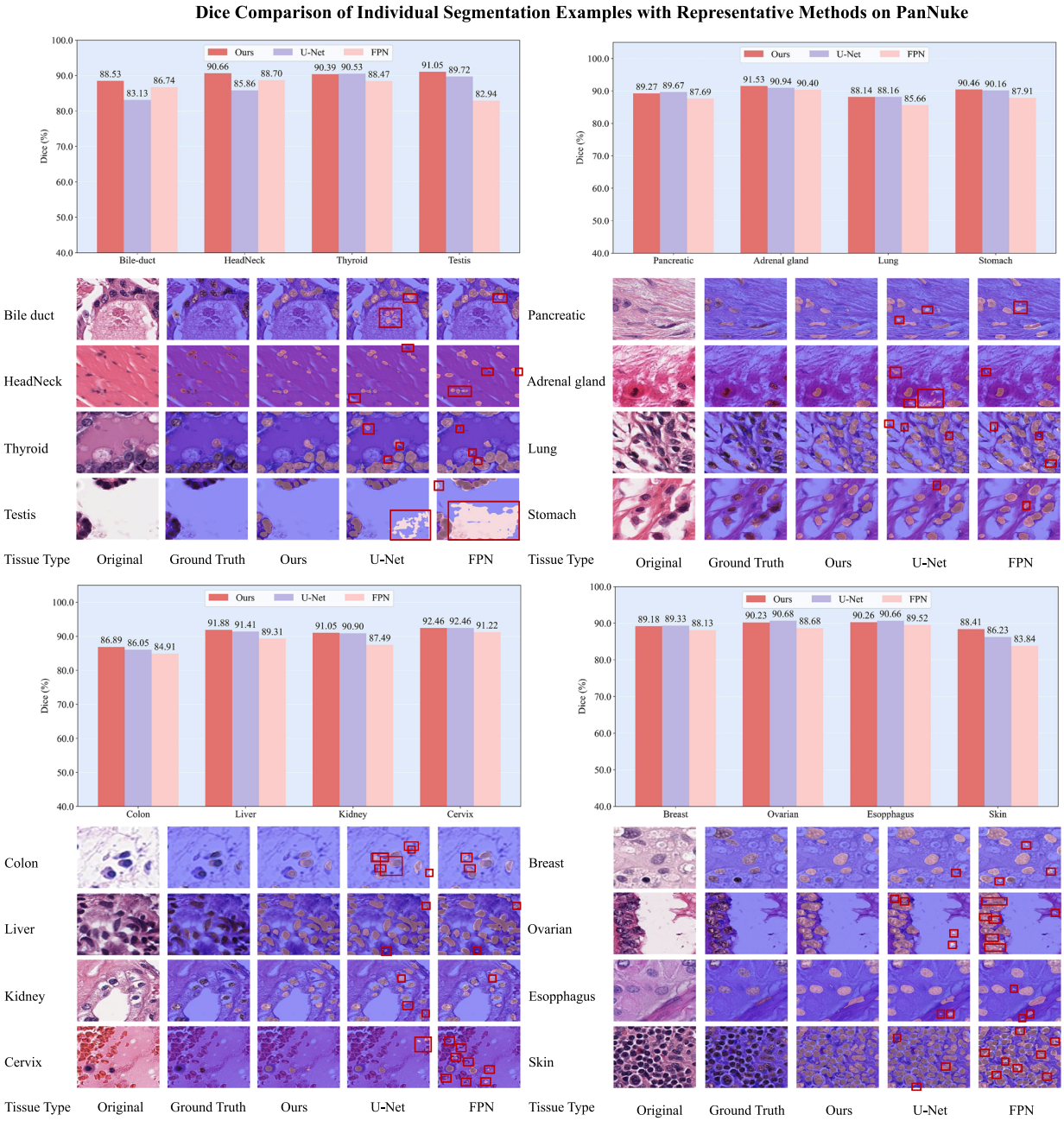


Fig. 3 | Individual quantitative and qualitative segmentation examples on the PanNuke dataset. The figure presents Dice score comparisons (bar charts) and visualized segmentation examples (Original, Ground Truth, Pathology-NAS (Ours), U-Net, FPN) for sixteen distinct tissue types. The masks generated by Pathology-NAS (Ours) generally exhibit a closer alignment with the ground truth annotations, particularly in accurately delineating intricate structures and reducing errors such as over-segmentation when compared to U-Net and FPN. This visual assessment is supported by the quantitative Dice scores, where our method often shows improved or competitive performance across the various tissue types. The red boxes are used to highlight common failure cases in the FPN and U-Net results, such as missed segmentation, incorrect segmentation, and excessive adhesion between different targets, areas where Pathology-NAS (Ours) often demonstrates more robust results.

sampling temperature in GPT-4 is used to control the randomness and diversity of LLM output. Larger values like 1.5 increase the randomness of response, while smaller values like 0.8 will make it more controllable. Consequently, the diversity of LLM response will inevitably affect network candidate recommendation and the process of architecture search. In this experiment, we seek to investigate the impact of GPT sampling temperature on architecture search. We choose the value of 0 and 1 respectively, where 0 means greedy sampling for deterministic results and 1 allows partial diversity. We apply these two temperature values on classification and segmentation tasks across all of the datasets in this study.

The iterative validation results during search and fine-tuning stage are presented in Fig. 4. From Fig. 4a, b, it can be observed that neural architectures of ViT are iteratively refined for better performance via the interaction with LLMs. The optimal architecture with best performance is obtained at the final round. This can be attributed to massive knowledge and complex reasoning abilities of LLMs. Those two curves in each figure demonstrate generation diversity controlled by sampling temperature. Stable and monotonically increasing performance improvement is obtained under the condition of temperature 0. By contrast, temperature 1 indicates more diversity and randomness, resulting in more obvIoUs performance fluctuations. As shown in Fig. 4c–f, architecture search for ShuffleNet and U-Net also exhibit similar tendency. Nevertheless, the performance fluctuations of these two models are smaller, which implicitly highlights the challenging stable training of vision transformers.

Table 3 | Detailed search cost and classification performance comparison for one-shot NAS methods on the BreakHis and Diabetic datasets

Dataset: BreakHis						
	ShuffleNet backbone			ViT backbone		
Metric	Random search	Cream	Pathology-NAS	Random search	AutoFormer	Pathology-NAS
Iterations ↓	500	300	**10**	500	300	**10**
GPT-4 API Calls	0	0	10	0	0	10
FLOPs ↓	275.96M	442.99M	**213.30M**	4.42G	1.28G	4.95G
Prec@1 (%) ↑	95.21 ± 0.34	97.13 ± 0.41	**99.98 ± 0.27*****	95.67 ± 0.22	96.21 ± 0.39	**98.08 ± 0.26*****
API Cost ($)	0.00	0.00	0.13	0.00	0.00	0.17
Latency (hrs)	0.000	0.000	0,001	0.000	0.000	0.001
ST (GPU hrs) ↓	32.40	10.63	**7.42**	67.28	31.72	**14.88**
TT (GPU hrs) ↓	32.400	10.630	**7.421**	67.280	31.720	**14.881**
Dataset: Diabetic						
	ShuffleNet Backbone			ViT Backbone		
Metric	Random search	Cream	Pathology-NAS	Random search	AutoFormer	Pathology-NAS
Iterations ↓	500	120	**10**	500	300	**10**
GPT-4 API Calls	0	0	10	0	0	10
FLOPs ↓	246.32M	440.07M	**240.25M**	4.77G	1.28G	**4.13G**
Prec@1 (%) ↑	65.03 ± 0.59	70.31 ± 0.38	**73.22 ± 0.34*****	58.47 ± 0.57	67.62 ± 0.24	**70.38 ± 0.22*****
API Cost ($)	0.00	0.00	0.12	0.00	0.00	0.18
Latency (hrs)	0.000	0.000	0.001	0.000	0.000	0.001
ST (GPU hrs) ↓	10.90	1.43	**1.16**	22.76	11.24	**6.12**
TT (GPU hrs) ↓	10.900	1.430	**1.161**	22.760	11.240	**6.121**

Results are averaged over 5 independent runs. The table presents a comprehensive breakdown of search costs (Iterations, GPT-4 API Calls, Latency (hrs), API Cost ($), ST (GPU hrs), TT (GPU hrs)) alongside key performance metrics (FLOPs, Prec@1 (%)) for Pathology-NAS compared with Random Search, Cream (for the ShuffleNet backbone), and AutoFormer (for the ViT backbone). Optimal values for performance and lower values for costs are typically highlighted in bold where applicable. Statistical significance of Pathology-NAS Prec@1 (%) performance compared to Random Search (assessed by independent two-sample Welch's *t* tests) is denoted by: $^{***}p < 0.001$ (very highly significant). Prec@1: Top-1 accuracy.
ST Search Time, *TT* Total Time, TT = ST + Latency, *Prec@1* Top-1 accuracy.

Table 4 | Detailed search cost and classification performance comparison for one-shot NAS methods on the Gastric Cancer datasets

Dataset: Gastric Cancer						
	ShuffleNet backbone			ViT backbone		
Metric	Random search	Cream	Pathology-NAS	Random search	AutoFormer	Pathology-NAS
Iterations ↓	500	120	**10**	500	300	**10**
GPT-4 API Calls	0	0	10	0	0	10
FLOPs ↓	286.16M	430.04M	**259.14M**	4.25G	4.82G	**4.25G**
Prec@1 (%) ↑	62.47 ± 0.21	54.88 ± 0.28	**63.15 ± 0.25*****	40.62 ± 0.30	41.40 ± 0.12	**43.04 ± 0.23*****
Prec@5 (%)	98.61 ± 0.25	98.05 ± 0.31	**98.99 ± 0.31****	93.57 ± 0.33	93.75 ± 0.25	**94.28 ± 0.02*****
API Cost ($)	0.00	0.00	0.15	0.00	0.00	0.16
Latency (hrs)	0.0000	0.0000	0.0011	0.0000	0.0000	0.0010
ST (GPU hrs) ↓	9.16	8.46	**3.00**	111.10	70.96	**8.02**
TT (GPU hrs) ↓	9.160	8.460	**3.001**	111.100	70.960	**8.021**

Results are averaged over 5 independent runs. The table presents a comprehensive breakdown of search costs (Iterations, GPT-4 API Calls, Latency (hrs), API Cost ($), ST (GPU hrs), TT (GPU hrs)) alongside key performance metrics (FLOPs, Prec@1 (%), Prec@5 (%)). Optimal values for performance and lower values for costs are typically highlighted in bold where applicable. Statistical significance of Pathology-NAS Prec@1 (%) and Prec@5 (%) performance compared to Random Search (assessed by independent two-sample Welch's *t* tests) is denoted by: $^{**}p < 0.01$ (highly significant), $^{***}p < 0.001$ (very highly significant).
ST Search Time, *TT* Total Time, TT = ST + Latency, *Prec@1* Top-1 accuracy, *Prec@5* Top-5 accuracy.

Expert prompt identified high-performing architectures with fewer search iterations

We conducted a comprehensive study to investigate the impact of prompt strategies on the search process and retrained architecture's performance, comparing three distinct approaches:

- Expert Prompt(Ep): Our proposed strategy, where GPT-4 is explicitly instructed to act as an "AI expert specializing in Neural Architecture Search for medical image analysis", equipped with detailed context about the Supernet, search space, task, and historical performance data.

Table 5 | Detailed search cost and segmentation performance comparison for one-shot NAS methods on BCSS, PanNuke and Zenodo Lung datasets, with U-Net based backbone

	Dataset: BCSS		Dataset: PanNuke		Dataset: Zenodo Lung	
Metric	**Random search**	**Pathology-NAS**	**Random search**	**Pathology-NAS**	**Random search**	**Pathology-NAS**
Iterations ↓	500	**10**	500	**10**	500	**10**
GPT-4 API Calls	0	10	0	10	0	10
FLOPs (G) ↓	12.63	**10.58**	17.72	**14.33**	38.45	**18.52**
Dice (%)	70.41 ± 0.18	**74.12 ± 0.22*****	88.24 ± 0.38	**89.31 ± 0.44****	71.77 ± 0.46	**73.94 ± 0.46*****
IoU (%)	55.38 ± 0.20	**59.45 ± 0.23*****	80.61 ± 0.47	**81.30 ± 0.45***	59.97 ± 0.39	**62.05 ± 0.31*****
API Cost ($)	0.00	0.14	0.00	0.15	0.00	0.16
Latency (hrs)	0.0000	0.0005	0.0000	0.0004	0.0000	0.0005
ST (GPU hrs) ↓	194.68	**12.14**	13.44	**2.14**	7.46	**0.72**
TT (GPU hrs) ↓	194.680	**12.141**	13.440	**2.140**	7.460	**0.721**

Results are averaged over 5 independent runs. The table presents a comprehensive breakdown of search costs (Iterations, GPT-4 API Calls, Latency (hrs), API Cost ($), ST (GPU hrs), TT (GPU hrs)) alongside key performance metrics (FLOPs, Dice (%)). Optimal values for performance and lower values for costs are typically highlighted in bold where applicable. Statistical significance of Pathology-NAS Dice (%) and IoU (%) performance compared to Random Search (assessed by independent two-sample Welch's *t* tests) is denoted by: $^{***}p < 0.001$ (very highly significant).
ST Search Time, *TT* Total Time, TT = ST + Latency.

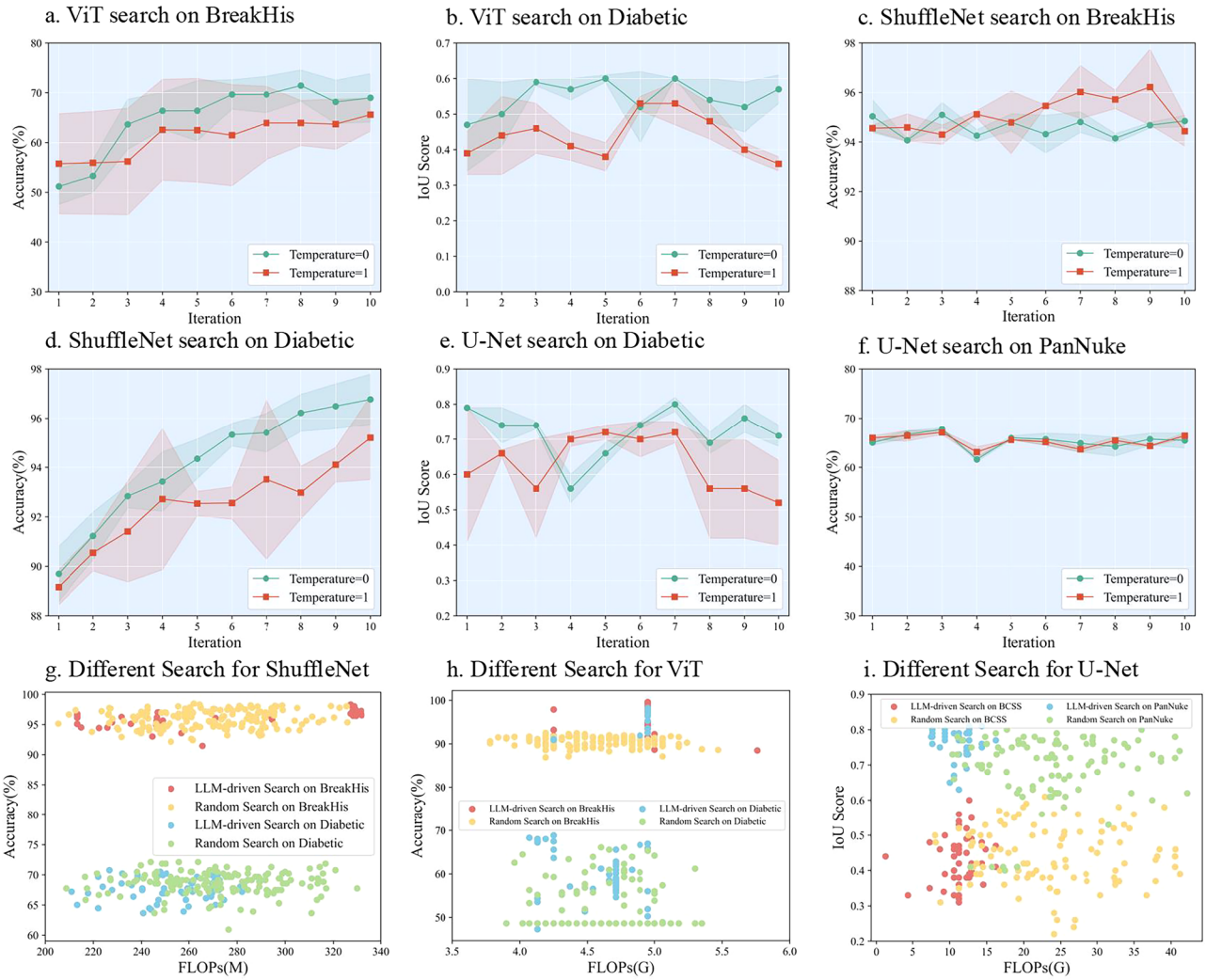


Fig. 4 | Quantitative evaluation results on pathology classification and segmentation tasks. a–f The iterative performance for searching backbone model architectures on classification and segmentation task with a temperature 0 and 1. We repeat each experiment with 10 iterations for 3 times. It can be found that stable increasing performance can be obtained under the condition of lower sampling temperature. **g–i** The comparison of our search strategy with random search strategy in terms of performance (accuracy or IoU score) and FLOPs of the model. It can be found that our LLM-driven search strategy has mostly achieved the optimal performance with low model complexity.

- Generic Assistant Prompt (GAP): A baseline where GPT-4 is prompted as a generic AI assistant, asked to suggest architectures based on the provided information without the specialized "expert" role.
- No System Prompt (NSP): A minimal approach where only the task-specific information and historical data are provided, without any explicit system-level instruction or role assignment.

As shown in Tables 9 and 10, EP-discovered architectures achieved superior or comparable performance. For instance, with ShuffleNet on BreakHis (Table 9), EP (ours) achieved 99.98% Top-1 accuracy, surpassing GAP (95.32%) and NSP (94.73%). EP also often identified architectures with competitive or improved computational efficiency. As shown in Table 9, for ViT on Diabetic, EP (ours) found an architecture with 4.13G FLOPs and 20.99M Params (achieving 70.38% Prec@1). In comparison, GAP yielded an architecture with 4.95G FLOPs and 25.12M Params but achieved only 52.17% Prec@1, and NSP resulted in 4.95G FLOPs and 25.12M Params for 63.86% Prec@1. Furthermore, EP consistently found high-performing architectures with fewer iterations and API calls. As shown in Table 10, with U-Net on PanNuke, EP (ours) averaged only 10 iterations and 1.6 API calls. This was significantly more efficient than GAP, which required 15 iterations (+50%) and 4.1 API calls (+156%), and NSP, which used 15 iterations (+50%) and 2.3 API calls (+44%). The above observations indicate that assigning an "expert" role could more efficiently exploit the potential NAS-related knowledge of LLM.

Moderate fine-tuning after LLM reference accelerates the searching process

Balancing the search cost (e.g, FLOPs and Params) and the final performance of optimal architectures is often of great significance in NAS. Especially for large-scale pathology slide datasets, performing a standard search strategy for each candidate architecture might be exhaustively costly. In the context of LLM-assisted architecture search, it is equally critical to manage the frequency of calling GPT-4 API and total search time. To be specific, the question arises as to how many training epochs should be conducted before feeding performance back to GPT-4. Consequently, we conducted an experiment to investigate the trade-off between architecture performance and training epochs. For each iteration, We finetune the pretrained model for 10, 20, 30, 40 epochs, respectively, and give feedback into LLM. The retraining strategy follows the same setting when training epoch is set to 20.

Tables 6 and 7 show the model performance of re-trained ShuffleNet, ViT, and U-net with varying training epochs, respectively. It can be observed that finetuning for 20 epochs have provided adequate accuracy for informative feedback. After 20 epochs of finetuning for each query-and-response iteration, almost all of the models explored in this paper have achieved the best performance on the medical image datasets.The only exception was fine-tuning the U-Net model for 30 epochs on the diabetic retinopathy dataset. Furthermore, Simply increasing training epochs fail to enhance the final accuracy but lead to a sharp increase during the search stage. As illustrated in architecture performance of re-trained ShuffleNet, ShuffleNet with 20 epochs of finetuning attains a 100% accuracy with 213M FLOPs and 1.80M params, outperforming 30 epochs-finetuning ShuffleNet with 305M FLOPs and 2.76M params by 5.82% in terms of accuracy.

Pathology-NAS achieves a better trade-off between accuracy and efficiency

In this experiment, we compared the accuracy progression of model retraining with the discovered architecture with our method and other search strategies. Figure 4 shows the distribution of top-1 accuracy and FLOPs on BreakHis and Diabetic for searching ShuffleNet architectures. It can be observed from Fig. 4g that ShuffleNet with LLM-driven search achieves the optimal accuracy/FLOPs trade-off. Specifically, our solution obtains 96.77% top-1 accuracy on BreakHis with 213.3M FLOPs, superior to other methods. Figure 4h illustrates the distribution of top-1 accuracy and FLOPs on BreakHis and Diabetic for search ViT architectures. It is shown that model complexities vary widely for architectures achieving comparable performance. However, LLM-driven search still rivals or outperforms random search in terms of performance across almost all FLOPs constraints. As is shown in Fig. 4i, evaluation results tends to exhibit a more dispersed distribution, with a wide range of both IoU and FLOPs values. Among a large number of cases, LLM-driven search generally dominates random search methods with the same level of FLOPs constraints.

In our paper, the reported FLOPs measure the computational cost of the discovered architectures during their inference stage only, without the cost of the neural architecture search. It's important to note that in our approach, GPT-4 is only used during the initial NAS phase as a guiding mechanism, involving a very limited number of calls (only 10 calls in our experiments). GPT-4 is completely uninvolved in the subsequent model training and inference stages. Therefore, there is a small, one-time API usage

Table 6 | Image classification performance of re-trained ShuffleNet and ViT with varying training epochs

Training epochs	BreakHis			Diabetic		
ShuffleNet	Prec@1 (%)↑	FLOPs (M)↓	Params (M)↓	Prec@1 (%)↑	FLOPs (M)↓	Params (M)↓
10	94.50	326.09	2.82	66.94	328.00	2.81
20	**99.98**	**213.30**	**1.80**	**73.22**	**240.25**	**2.10**
30	96.37	305.47	2.76	68.86	250.79	2.23
40	95.32	327.45	2.82	63.93	258.03	2.40
ViT	Prec@1 (%)↑	FLOPs (G)↓	Params (M)↓	Prec@1 (%)↑	FLOPs (G)↓	Params (M)↓
10	97.63	5.35	27.19	48.63	4.95	25.12
20	**98.08**	4.95	25.12	**70.38**	**4.13**	**20.99**
30	96.78	**4.77**	**24.24**	66.12	4.13	20.99
40	96.90	5.00	25.42	64.75	4.95	25.12

We adjust the fine-tuning epochs with coverage from 10 to 40. The best metrics are highlighted in bold. For both ShuffleNet and ViT search, Pathology-NAS generally achieves the optimal re-training performance when fine-tuning 20 epochs for each search iteration.

Table 7 | Image segmentation performance of re-trained U-Net with varying training epochs

Training epochs	BCSS			PanNuke				
	Dice (%)↑	IoU (%)↑	FLOPs (G)↓	Params (M)↓	Dice (%)↑	IoU (%)↑	FLOPs (G)↓	Params (M)↓
10	69.58	53.95	14.53	**6.64**	89.28	81.31	16.23	11.39
20	**74.33**	**59.68**	10.58	11.37	89.24	81.25	14.33	**8.34**
30	70.14	54.65	12.63	12.76	**89.31**	**81.35**	**12.63**	12.76
40	71.93	56.76	**8.67**	10.43	89.04	80.93	12.63	12.76

We adjust the fine-tuning epochs with coverage from 10 to 40. The best metrics are highlighted in bold. Pathology-NAS achieves the optimal dice score and IoU score when fine-tuning for 20 epochs on BCSS and 30 epochs on PanNuke. FLOPs and Params of different fine-tuning epochs are on the same order of magnitude.

Table 8 | Domain generalization performance: evaluation of architectures (discovered on source datasets) on unseen external target datasets without retraining

Source dataset	Target external dataset	Backbone	Method	Performance (%)↑
BreakHis	SkinTumor[27]	ShuffleNet	Random Search	39.21 ± 0.57 (Prec@1)
BreakHis	SkinTumor[27]	ShuffleNet	Pathology-NAS	74.50 ± 0.98 (Prec@1)
BreakHis	SkinTumor[27]	ViT	Random Search	73.52 ± 0.94 (Prec@1)
BreakHis	SkinTumor[27]	ViT	Pathology-NAS	82.35 ± 0.29 (Prec@1)
BreakHis	SkinTumor[27]	MobileNetV3	Random Search	45.30 ± 0.70 (Prec@1)
BreakHis	SkinTumor[27]	MobileNetV3	Pathology-NAS	78.10 ± 0.60 (Prec@1)
PanNuke	Polyp[28]	U-Net	Random Search	39.18 ± 0.27 (Dice)
PanNuke	Polyp[28]	U-Net	Pathology-NAS	62.07 ± 0.45 (Dice)

Performance metrics are reported as mean ± std. dev.

Table 9 | Ablation study: impact of prompting strategies on classification NAS performance

Backbone	Dataset	Prompt strategy	Iterations (↓)	API Calls (↓)	Prec@1 (%)↑	FLOPs↓	Params (M)↓
ShuffleNet	BreakHis	EP (Ours)	**10**	**3.5**	**99.98*****	**213.30M**	**1.80**
		GAP	10	3.9	95.32	238.27M	2.38
		NSP	14	3.5	94.73	225.86M	2.07
	Diabetic	EP (Ours)	**10**	**4.2**	**73.22*****	**240.25M**	**2.10**
		GAP	11	4.5	66.85	328.80M	2.81
		NSP	10	4.3	66.30	249.54M	2.44
ViT	BreakHis	EP (Ours)	**10**	4.7	**98.08*****	4.95G	25.12
		GAP	15	7.5	96.09	4.83G	24.53
		NSP	15	**4.6**	95.30	**4.60G**	**23.35**
	Diabetic	EP (Ours)	**10**	**4.6**	**70.38*****	**4.13G**	**20.99**
		GAP	10	4.6	52.17	4.95G	25.12
		NSP	10	5.4	63.86	4.95G	25.12

Results are averaged over 5 runs. Best results for each metric within a group are typically achieved by EP. The *Prec@1* of *EP* consistently outperforms the counterpart of *GAP* and *NSP* with very high significance ($p < 0.001$).
EP Expert Prompt (Ours), *GAP* Generic Assistant Prompt, *NSP* No System Prompt, *Prec@1:* Top-1 accuracy.

cost during the search phase, and this does not affect the inference efficiency of the final discovered models.

Consistent performance of Pathology-NAS across various datasets

To further evaluate the segmentation performance and generalizability of Pathology-NAS, we conducted additional experiments on the CoNSeP dataset, including a direct comparison with HoVer-Net. As shown in Supplementary Table 1, Pathology-NAS achieved a Dice score of 94.83% and an Aggregated Jaccard Index (AJI) of 64.71%, representing an improvement of 10.01% in Dice and 10.08% in AJI over HoVer-Net (Dice: 84.82%, AJI: 54.63%). Furthermore, Pathology-NAS demonstrated significantly higher efficiency, using only 25.9 GFLOPs and 15.75M parameters. This was ~7.6 times fewer FLOPs and 3.5 times fewer parameters compared to HoVer-Net (197.05 GFLOPs and 54.74M parameters).

The generalizability of Pathology-NAS for image segmentation tasks was also further assessed by incorporating the Zenodo lung cancer dataset into our evaluations. As detailed in Supplementary Table 2, Pathology-NAS (U-Net backbone) demonstrated excellent performance. Specifically, our method achieved a Dice score of 73.48% and an IoU of 61.74%, outperforming standard U-Net (Dice: 70.97%, IoU: 58.63%) and FPN (Dice: 69.52%, IoU: 57.22%). Notably, Pathology-NAS was also significantly more efficient, utilizing only 18.51G FLOPs and 14.68M parameters, compared to U-Net (23.61G FLOPs, 24.44M params) and FPN (27.36G FLOPs, 26.01M params). These findings highlight the effectiveness of Pathology-NAS in discovering high-performing, efficient architectures for segmentation tasks on various medical datasets.

Furthermore, we conducted experiments on the Gastric Cancer dataset[26]. Our LLM-guided approach effectively balances high performance with model efficiency, as demonstrated in Table 4. With the ViT backbone, Pathology-NAS achieves a Top-1 accuracy of 43.04% ± 0.23. This notably surpasses both Random Search (Top-1: 40.62% ± 0.30) and AutoFormer (Top-1: 41.40% ± 0.12). Notably, it is achieved with FLOPs lower than AutoFormer (4.82G). Furthermore, its Top-5 accuracy for ViT at 94.28% ± 0.02 also outperforms Random Search (93.57% ± 0.33) and AutoFormer (93.75% ± 0.25). For the ShuffleNet backbone, Pathology-NAS also improves Top-1 accuracy to 63.15% ± 0.25 compared with Random Search (62.47% ± 0.21) while utilizing significantly lower inference FLOPs (259.14M) compared to Random Search (286.16M).

Pathology-NAS demonstrates strong domain generalization on unseen external datasets without retraining

To demonstrate the domain generalization of Pathology-NAS, we evaluated architectures discovered on our source datasets-BreakHis for classification and PanNuke for segmentation, by directly validation on unseen external datasets: SkinTumor[27] (for BreakHis-derived models) and Polyp[28] (for PanNuke-derived models), respectively, without any retraining or fine-tuning. As presented in Tables 8–10 Pathology-NAS demonstrates strong generalization capabilities. For instance, a Pathology-NAS discovered ShuffleNet architecture (trained on BreakHis) applied to SkinTumor[27] achieves a top-1 accuracy of 74.50% ± 0.98%, which is a 35.29%

Table 10 | Ablation study: impact of prompting strategies on segmentation NAS performance

Backbone	Dataset	Prompt strategy	Iterations (↓)	API calls (↓)	Dice (%)↑	IoU (%)↑	FLOPs (G)↓	Params (M)↓
U-Net	BCSS	EP (Ours)	**10**	**1.8**	**74.33*****	**59.68*****	**10.58**	**11.37**
		GAP	10	1.9	73.78	58.99	19.83	15.75
		NSP	11	2.2	73.90	59.22	17.46	12.73
	PanNuke	EP (Ours)	**10**	**1.6**	**89.31*****	**81.35*****	**14.33**	**8.34**
		GAP	15	4.1	89.26	81.28	23.12	11.30
		NSP	15	2.3	88.89	80.71	16.23	9.00

Results are averaged over 5 runs. Best results for each metric within a group are typically achieved by EP. The Dice and IoU of *EP* consistently outperform the counterpart of *GAP* and *NSP* with very high significance ($p < 0.001$).
EP Expert Prompt (Ours), *GAP* Generic Assistant Prompt, *NSP* No System Prompt.

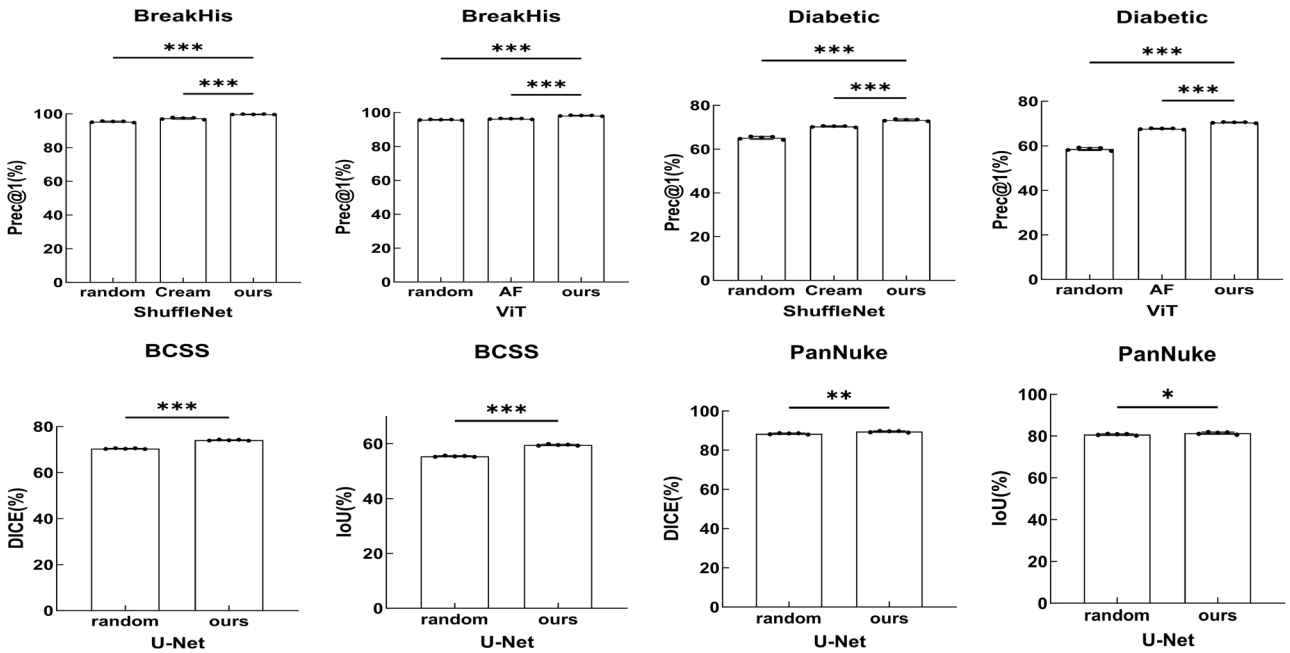


Fig. 5 | Significance testing of Top-1 accuracy (Prec@1(%)) for different NAS methods on the BreakHis, Diabetic, BCSS and PanNuKe datasets, utilizing ShuffleNet, ViT and U-Net backbones. The x-axis displays the compared search methods, including random (i.e., Random Search), Cream, AF (i.e., AutoFormer), and "ours" (i.e., Pathology-NAS), grouped by dataset and backbone. The y-axis represents the Prec@1(%) performance. Asterisks indicate statistical significance levels assessed by independent two-sample Welch's *t* tests: * denotes $p < 0.05$ (significant), ** denotes $p < 0.01$ (highly significant), and *** denotes $p < 0.001$ (very highly significant). Prec@1: Top-1 accuracy. Prec@5: Top-5 accuracy.

improvement over the Random Search-derived architecture (39.21% ± 0.57%). Similarly for segmentation, a U-Net architecture found by Pathology-NAS (trained on PanNuke) achieves a Dice score of 62.07% ± 0.45% on Polyp[28], representing an improvement of 22.89% over the Random Search baseline (39.18% ± 0.27%). These results on unseen external datasets underscore the robustness and generalization of architectures discovered by Pathology-NAS.

Significance analysis of search method comparison

To enhance statistical robustness, we conducted comprehensive statistical analyses across all primary experiments (including newly added datasets). For statistical assessment, we conducted 5 independent runs for each method and dataset combination, then performed independent two-sample Welch's *t* tests (which do not assume equal variances) with a one-sided alternative hypothesis to test if Pathology-NAS performance is significantly superior. Significance levels are reported using a tiered system: * for $p < 0.05$ (significant), ** for $p < 0.01$ (highly significant), and *** for $p < 0.001$ (very highly significant).

As shown in Figs. 5 and 6, Pathology-NAS consistently achieves statistically significant improvements over all baseline methods across different datasets, tasks, and backbone architectures. These improvements reach $p < 0.001$ (very highly significant) level in most cases. For example, on BreakHis with ShuffleNet backbone, Pathology-NAS achieves 99.98% ± 0.27% top-1 accuracy compared to Random Search's 95.21% ± 0.34% (Table 3). For segmentation on PanNuke with U-Net backbone, our method achieves 89.31% ± 0.44% Dice, versus Random Search's 88.24% ± 0.38% (Table 5), while using significantly fewer computational resources.

As illustrated in Fig. 6, these significant improvements extend to our newly added datasets as well. For the Gastric Cancer dataset (Table 4), Pathology-NAS achieves 63.15% ± 0.25% top-1 accuracy with ShuffleNet backbone (versus Random Search's 62.47% ± 0.21%) and 43.04% ± 0.23% with ViT backbone (versus Random Search's 40.62% ± 0.30%). Similarly, for the Zenodo Lung Cancer dataset (Table 5), our method demonstrates superior performance with strong statistical significance. Detailed statistical results for all experiments are provided in the corresponding tables throughout the manuscript and supplementary materials.

Discussion

We introduce Pathology-NAS, a LLM-driven medical image analysis framework empowered by neural architecutre search, which is developed for

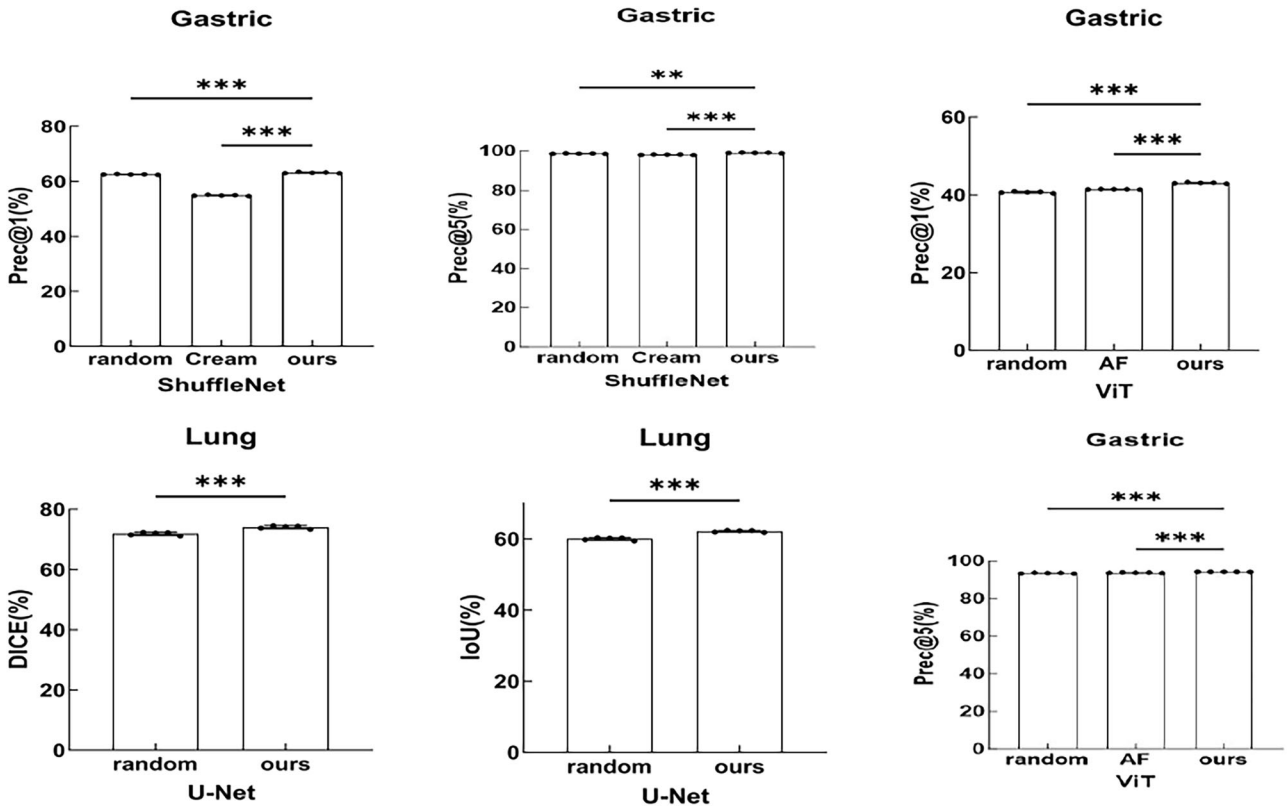


Fig. 6 | Significance testing of Top-1 accuracy (Prec@1(%)) for different NAS methods on the Gastric (i.e., Gastric Cancer) and Lung (i.e., Zenodo Lung Cancer) datasets, utilizing ShuffleNet and ViT backbones. The x-axis displays the compared search methods, including random (i.e., Random Search), Cream, AutoFormer, and "ours", grouped by dataset (Gastric, Lung) and backbone (ShuffleNet, ViT). The y-axis represents the Prec@1(%) performance. Asterisks indicate statistical significance levels assessed by independent two-sample Welch's *t* tests: ** denotes $p < 0.01$ (highly significant), and *** denotes $p < 0.001$ (very highly significant). Prec@1: Top-1 accuracy. Prec@5: Top-5 accuracy.

universal and lightweight malignant tissue classification and segmentation across a diverse spectrum of anatomical regions. Pathology-NAS integrates task-aware domain knowledge possessed by LLMs with search-and-fine-tuning on pathology images, thereby obtaining the ability to search the optimal architecture for target tasks within a short period. Pathology enables automated and customized design of deep learning models, rendering itself a versatile AI agent across a large amount of intelligent medical imaging analysis tasks.

We conduct extensive experiments to evaluate the performance of Pathology-NAS on a variety of pathological slides, covering disease type classification and malignant tissue segmentation across different anatomical regions. Through comprehensive evaluations, Pathology-NAS have showcased extraordinary capabilities in accurate tissue recognition and segmentation with much less model FLOPs constraint. Specifically, its performance not only significantly outperforms that of existing the state-of-the-art task-specific models, the visualization results show that our solution presents more distinct segmentation boundaries compared to other methods. The key characteristic of Pathology-NAS lies in automated model design via LLM-driven architecture search. Tremendous domain knowledge about medical analysis could be exploited to recommend a suite of promising architectures. After fine-tuning on downstream medical image datasets, these architectures are prone to demonstrate superior performances, accordingly alleviate the time and hardware burden for architecture search. Consequently, our method achieves a favorable performance efficiency trade-off by conducting the LLM-driven architecture search, which holds the potential to facilitate the cost-effective clinical prediction service.

While Pathology-NAS has achieved remarkable performance, there still remain a few limitations. One of the limitations is the supernet training from single path one-shot used in this study. Although one-shot weight sharing has substantially reduced the search cost, each sub-network still suffers from the insufficient optimization during supernet training with uniform sampling. Given the limited number of some medical datasets, it is non-trivial to train a high-performing model solely based on pathology slides. This also underscores the necessity of large-scale generic visual datasets during the pre-training phase. Another limitation lies in the backbone model and search space for target tasks. We only search over a number of fundamental operations and blocks in this study. Although neural architecture search is conducted on a few representative classification and segmentation models, developing search strategies for visual foundation models presents a challenging yet promising direction. However, these limitations do not conflict with the generality of our method. Since the supernet is pretrained on a vast amount of generic images, Pathology-NAS enables swift adaptation to various downstream tasks.

In conclusion, this study investigates the feasibility of cost-effective and versatile pathology analysis framework, which can be rapidly adapted to downstream medical tasks via LLM-driven neural architecture search. Pathology-NAS, as an intelligent and efficient fundamental solution, offers tremendous potential to accelerate the advancement of automatic diagnostic tools and the personalization of treatment strategies.

Algorithm 1. Pathology-NAS for Diverse Pathology Analysis

1: **Input**:
- Set of supernet types $\{\mathcal{S}_c\}$, where $c \in$ {U-Net, ViT, ShuffleNet}
- Large generic training datasets D_{gen}
- Medical tasks $\mathcal{T}$ and corresponding datasets $\mathcal{D}_m$
- Budgets $\mathcal{B}$ for allowable FLOPs
- LLM for generating architecture configurations
- Universal Task NAS Prompt (UNP) $\mathcal{P}$

2: **Output**: Optimal architecture configuration $a^*_{t_i,d_i}$ for each task t_i and dataset d_i

3: Define a task-to-supernet mapping $\mu : \mathcal{T} \rightarrow \{\mathcal{S}_c\}$
4: Pretrain each supernet $\mathcal{S}_c$ on D_{gen} to generate model configurations $\mathcal{M}_c$ using Eqs. (1)–(3).
5: Initialize UNP prompt $\mathcal{P}_{\text{init}}$ for each supernet based on the $\mathcal{M}_c$
6: **for** each task $t_i \in \mathcal{T}$ and dataset d_i in $\mathcal{D}_m$ **do**
7: Select the appropriate supernet $\mathcal{S}_{\mu(t_i)}$ and its config $\mathcal{M}_{\mu(t_i)}$ based on the mapping $\mu(t_i)$
8: Initialize performance memory $\beta_{t_i,d_i} = \{\}$
9: **for** $t = 0$ to T **do**
10: **if** $t = 0$ **then**
11: Set prompt $\mathcal{P}_{t_i,d_i} = \mathcal{P}_{\text{init}}(\mathcal{S}_{\mu(t_i)}, \mathcal{M}_{\mu(t_i)})$
12: **else**
13: Update prompt $\mathcal{P}_{t_i,d_i} = \mathcal{P}_{\text{init}}(\mathcal{S}_{\mu(t_i)}, \mathcal{M}_{\mu(t_i)}, \beta_{t_i,d_i})$
14: **end if**
15: $a_{t,d_i} \leftarrow \text{LLM}(\mathcal{P}_{t_i,d_i})$ {Generate preferred model architecture}
16: $(\text{acc}_{t,d_i}, \text{FLOPs}_{t,d_i}) \leftarrow$ Finetune and evaluate a_{t,d_i} on $D_{tr}(t_i, d_i)$, $D_{val}(t_i, d_i)$
17: **if** $\text{FLOPs}_{t,d_i} \leq \mathcal{B}$ and $\text{acc}_{t,d_i} > \max(\beta_{t_i,d_i}[acc])$ **then**
18: $\beta_{t_i,d_i}[acc] \leftarrow \text{acc}_{t,d_i}$
19: $\beta_{t_i,d_i}[FLOPs] \leftarrow \text{FLOPs}_{t,d_i}$
20: **end if**
21: **end for**
22: $a^*_{t_i,d_i} = \beta_{t_i,d_i}[a_{t,d_i}]$
23: **end for**
24: **return** $a^*_{t_i,d_i}$ for each t_i and d_i

Methods

Datasets curation and processing

We collected extensive tissue slide images of cancer diagnosis and pathology analysis from various sources from the Internet, including Kaggle, GrandChallenge, and scientific data. For histopathology classification task, we use datasets from BreakHis challenge[29] and Diabetic retinopathy challenge[30], SkinTumor dataset[27] and Gastric Cancer Histopathology Tissue Image Dataset[26]. The Breast Cancer Histopathology Image Classification (BreakHis) consists of 9109 microscopic images of breast tumor tissue obtained from 82 patients. Diabetic retinopathy dataset is composed of 3662 retina images taken using fundus photography under a variety of conditions. BreakHis contains 2480 benign and 5429 malignant samples with different magnifying factors (40X, 100X, 200X, and 400X). Each image is with 700 × 460 pixels, 3-channel RGB, 8-bit in each channel and PNG format. The diabetic retinopathy detection dataset is a subset of data from APTOS 2019 Blindness Detection, where the original file consists of 20GB of data among 13,000 images. The SkinTumor dataset is a refined version of ISIC 2019 challenge dataset, which includes 25,331 dermoscopic images in 2 categories, 8 subtypes. All melanoma diagnoses in the dataset were confirmed by pathological annotations. The Gastric Cancer Histopathology Tissue Image Dataset (Gastric Cancer) provides a large database of nearly 31,000 histological images from 300 whole slide images, annotated for 8 distinct tissue categories, making it a suitable benchmark for evaluating multi-class classification performance. ImageNet-1k dataset is utilized to pretrain the supernet of vision transformer and shuffle net for neural architecture search.

For image segmentation task, we use datasets from Breast Cancer Semantic Segmentation[31] (BCSS) and Cancer Instance Segmentation and Classification[32] (PanNuke), which were derived from The Cancer Genome Atlas (TCGA) project. We use additional segmentation datasets such as ConSep[33], Zenodo Lung Cancer[34] and Polyp[28]. The BCSS dataset holds more than 20,000 segmentation annotations of breast cancer tissue regions. The number of samples in the training set, validation set, and test set are 30,760, 5429, and 4021, respectively. The PanNuke dataset includes histopathology images that were semi automatically generated nuclei instance segmentation and classification, covering tremendous nuclei labels across 19 different tissue types. It is composed of 2661 samples and 205,343 labeled nuclei, each with a ground truth mask. The ConSep dataset consists of 41 H&E stained image tiles, each of size 1000 × 1000 pixels at 40× objective magnification. Images were extracted from 16 colorectal adenocarcinoma (CRA) WSIs, each belonging to an individual patient. The Zenodo Lung Cancer dataset is a dataset of 85 tiles of size 1024 × 1024 pixels with cell level annotations extracted from 9 lung WSIs. The annotations define the cells' nuclei shape and classify each cell as either cancerous or non-cancerous. The Polyp dataset includes 1000 frames taken from colonoscopy videos, which feature numerous instances of polyp. The ground truth is represented by a mask that corresponds to the area of the image occupied by the polyp. To ensure consistency and compatibility with deep learning models, all whole slide images (WSI) have been cropped into small patches using a sliding window method. Each image is in the PNG format of 224 × 224 and 512 × 512 pixels. All images have been resized into the size of 224 × 224 pixels to ensure uniformity for histology classification task. We utilize z-score normalization with default mean and standard deviation to rescale the pixel values. The implementation of these standardization protocols guaranteed a consistent and harmonized approach throughout all imagery, thereby streamlining their incorporation into the successive phases of the model's learning and assessment procedure.

LLM-driven neural architecture search

In this paper, we propose a LLM-driven neural architecture search (NAS) pipeline to apply a universal and efficient pathology segmentation and recognition framework, as illustrated in Fig. 1. This approach is inspired by one-shot NAS methods that adopt a weight-sharing strategy to avoid training each subnet independently[19,20]. By decoupling the supernet training and architecture search, one-shot NAS can alleviate the problematic coupling of joint optimization. The architecture search space, $\mathcal{A}$, is represented as a set of supernets $\mathcal{S}_c(\mathcal{A}, W)$, where c represents a different supernet types (details see Section "Network architecture"), W is the weight of the supernet. W is shared among all possible architecture candidates, i.e., subnet $\alpha \in \mathcal{A}$ in $\mathcal{N}$. Searching for the optimal architecture a^* is formulated as a dual-stage optimization problem:

$$W_{\mathcal{A}} = \text{argmin}_W \mathcal{L}_{train}(\mathcal{S}_c(\mathcal{A}, W)), \tag{1}$$

which indicates optimizing W based on loss function on training dataset by sampling subnets. The second stage is to search the optimal architecture of subnet $\alpha \in \mathcal{A}$ via the validation performance of the tuned weights of $W_{\mathcal{A}}$.

$$\alpha^* = \arg\max_{\alpha \in \mathcal{A}} Acc_{val}(\mathcal{S}_c(\alpha, w)), \tag{2}$$

where the sampled subnet inherits a weight w from $W_{\mathcal{A}}$. PrevIoUs works resort to different search algorithms to find the fittest candidate architecture, such as random search[35,36], reinforcement learning[37,38] and evolution search[20,39]. However, It remains uncertain why the pre-trained weights $W_{\mathcal{A}}(\alpha)$ remain effective for any arbitrary architecture α. According to the principle that the supernet weights $W_{\mathcal{A}}$ should be optimized in a way that all architectures in the search space are optimized simultaneously, a subnet α is randomly sampled and optimized for each iteration. This is expressed as:

$$W_{\mathcal{A}} = \text{argmin}_W \mathbb{E}_{\alpha \sim \Gamma(\mathcal{A})}\left[\mathcal{L}_{train}\left(\mathcal{S}_c(\alpha, W(\alpha)\right)\right], \tag{3}$$

where $\Gamma(\mathcal{A})$ is a prior distribution of $\alpha \in \mathcal{A}$.

During the pretraining stage, we design a supernet structure that each architecture is a single path, which means that no overlap blocks exists among subnet architectures. More specifically, for a subnet $\alpha \in \mathcal{A}$ with a stack of l layers, the architecture is represented as follows:

$$\begin{cases} \alpha & = (\alpha^{(1)}, \cdots, \alpha^{(i)}, \cdots, \alpha^{(l)}) \\ w & = (w^{(1)}, \cdots, w^{(i)}, \cdots, w^{(l)}), \end{cases} \tag{4}$$

where the $\alpha^{(i)}$ is the sampled block in the i-th layer and $w^{(i)}$ is the corresponding block weight. Therefore, $\alpha^{(i)}$ is actually sampled from a set of

n choice blocks, which can be denoted as:

$$\begin{cases} \alpha^{(i)} & \in (b_1^{(i)}, \cdots, b_j^{(i)}, \cdots, b_n^{(i)}) \\ w^{(i)} & \in (w_1^{(i)}, \cdots, w_j^{(i)}, \cdots, w_n^{(i)}), \end{cases} \tag{5}$$

where the $b_j^{(i)}$ is a choice of candidate blocks in the i-th layer and $w_j^{(i)}$ is the corresponding weight.

In our setting it is forced that one random path drop any operations in a block, eliminating the occurrence of shot cut connection. Moreover, it also helps to reduce the entanglement of different operation weights. Note that the supernet's architecture is uniformly sampled from a fixed prior distribution, following the principle in existing works that purely random search from a supernet is competitive enough.

During the architecture search stage, we leverage GPT-4 as a sophisticated black-box optimizer to guide our neural architecture search (NAS). Although prevIoUs research has explored numerous algorithms to identify optimal architectures efficiently and accurately, they still suffer from substantial computational burdens. Recent studies suggest that GPT-4, endowed with vast inherent knowledge, is suitably equipped to tackle NAS tasks, substantiating its capability beyond traditional models[40].

The GPT-4-driven architecture search works in an iterative improvement process. In the first round, the NAS problem statement is provided to the GPT-4 model in a natural language format, along with an initial candidate model architecture configuration $\mathcal{M}_{\mu(t_i)}$ derived from the Eqs. (1)–(3). Then, we initialize the UNP prompt $\mathcal{P}_{\text{init}}$ for GPT-4 with the configuration and the supernet types $\mathcal{S}_{\mu(t_i)}$, setting a robust foundation for subsequent optimizations:

$$\mathcal{P}_{t_i,d_i} = \begin{cases} \mathcal{P}_{\text{init}}(\mathcal{S}_{\mu(t_i)}, \mathcal{M}_{\mu(t_i)}) & \text{if } t = 0, \\ \mathcal{P}_{\text{init}}(\mathcal{S}_{\mu(t_i)}, \mathcal{M}_{\mu(t_i)}, \beta_{t_i,d_i}) & \text{otherwise}, \end{cases} \tag{6}$$

where β_{t_i,d_i} includes accumulated performance metrics such as accuracy and computational efficiency (FLOPs), providing a feedback loop to refine the search.

After initialization, the current state of the network, represented by $\mathcal{P}_{t_i,d_i}$, is fed into the LLM model to recommend a new preferred architecture configuration:

$$a_{t,d_i} \leftarrow \text{LLM}(\mathcal{P}_{t_i,d_i}). \tag{7}$$

The proposed architecture a_{t,d_i} is then fine-tuned on the respective training dataset $D_{tr}(t_i, d_i)$ and evaluated on the validation dataset $D_{val}(t_i, d_i)$ to obtain empirical accuracy and FLOPs:

$$(\text{acc}_{t,d_i}, \text{FLOPs}_{t,d_i}) \leftarrow \text{Fine-tune and evaluate } a_{t,d_i}. \tag{8}$$

The performance memory β_{t_i,d_i} is updated based on the new metrics if they meet the defined criteria of computational budget $\mathcal{B}$ and improved accuracy:

$$\text{if}(\text{FLOPs}_{t,d_i} \le \mathcal{B}) \wedge (\text{acc}_{t,d_i} > \max(\beta_{t_i,d_i}[\text{acc}])) \text{ then update } \beta_{t_i,d_i}. \tag{9}$$

The iterative process continues until a predetermined number of iterations are completed. The optimal architecture for each task and dataset is selected based on the best performance metrics recorded in the performance memory:

$$a^*_{t_i,d_i} = \beta_{t_i,d_i}[a_{t,d_i}]. \tag{10}$$

This LLM-driven Pathology-NAS not only ensures that the architectures are optimized for performance but also adheres to computational constraints, making it a practical approach for medical image analysis tasks, as shown in Algorithm 1.

Network architecture

The networks of the supernet utilized in this study represent a new paradigm that dynamically adapts to specific tasks, incorporating both CNN-based models and vision transformer models, which have achieved remarkable performance in existing image recognition and segmentation tasks[41],as shown in Fig. 7a. The architecture includes three main model types:

- ShuffleNet v2: Adapted for classification, this model comprises 20 choice blocks, each offering 4 operation candidates: 3×3, 5×5, 7×7 convolutions, and an identity block[18].
- U-net: Employed for segmentation tasks, it includes 8 choice blocks corresponding to 4 down-sampling and 4 up-sampling layers. Each choice block is enhanced with a squeeze-and-excitation (SE) network to adaptively calibrate channel-wise feature responses, thereby improving performance[23].
- Vision Transformer (ViT): Utilized as another backbone model for medical image recognition, it features a modular design with a patch embedding module, a classifier head, and a series of stacked transformer blocks, each comprising a multi-head self-attention layer and a feed forward network layer with layer normalization. The search space for ViT includes options for depth of transformer layers (12, 13, 14), number of heads for attention layers (3, 4, 6, 8), and scale ratios for MLP layers (3, 4, 5).

These models are pretrained on the ImageNet-1k dataset to develop robust initial capabilities, which are then specialized through architecture search and fine-tuning on pathology images. This dynamic selection of network architectures allows Pathology-NAS to tailor its approach to effectively address the diverse requirements of medical image analysis tasks.

Training protocols

The training protocols include datasets preparation, model parameters, loss functions, evaluation metrics, and baseline methods.

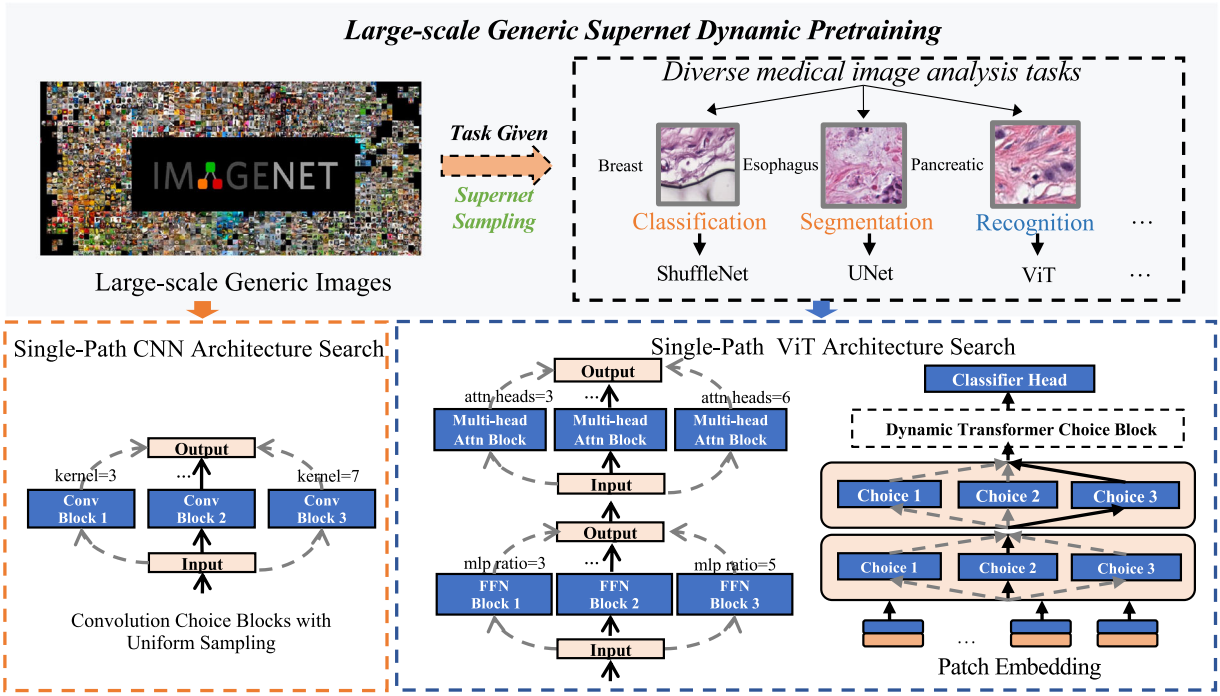
Dataset. To thoroughly exploit rich visual features in large-scale images, the supernet is pretrained using the ImageNet-1k dataset, with 10% of the training set reserved for validation to ensure fair model evaluation. For histopathology images used in the architecture search, we follow the official setting in terms of data partition. The 65% samples of BreakHis constitute the training set while other images are included as the test dataset. For BCSS, there are totally 30,760 train images, 5429 validation images, and 4021 test images. The diabetic retinopathy dataset contains 3662 images, which is divided into 80%, 10%, 10% as training, validation, and test, respectively.

Network initialization and loss functions. Following the settings in prevIoUs works[18,20,42], the weights of convolutional models are simply initialized with normal and uniform distribution, while the weights of vision transformer are initialized with truncated normal distribution. The loss functions used are cross entropy loss for image classification and dice loss for image segmentation.

Training configuration. The networks are optimized by an SGD optimizer with an initialized learning rate of 0.1 and a weight decay of 5e−2. We deploy distributed data parallel for model training, where the global batch size is 256. The CNN and ViT models were trained on 4 NVIDIA V100(32G) GPUs for 500 epochs, selecting the checkpoint with the best validation accuracy as the final model. During the GPT-4 assisted search phase, we manually set 10 iterations and 20 finetuning epochs for each iteration.

Baseline and comparative methods. We conducted comparative experiments against state-of-the-art image classification models (ResNet[13]/EfficientNet[14]/Swin-transformer[15]), semantic segmentation

a. Illustration of supernet model



b. Illustration of prompt for U-Net search

Your task is to assist me in selecting the best for a given {Task} architecture(e.g. U-Net). The model will be trained and tested on pathology images, and your objective will be to maximize the model 's performance on pathology images.

The model architecture will be defined as follows:
layer0: VGGBlock(input_channels=224, nb_filter[0], nb_filter[0], 3),
layer1: VGGBlock(input_channels=nb_filter[0], nb_filter[1], nb_filter[1], kernels[1]),
.....
The nb_filter is defined as follows:
*nb_filter = [32, 32 * 2, 32 * 4, 32 * 8, 32 * 16]*
For the `kernel` variable, the available kernel size for each index would be:
{
kernels[1]: [3, 5, 7, 'id'],
.....
}

The implementation of the VGGBlock and SELayer is as follows:

```
class VGGBlock(nn.Module):
    def __init__(self, in_channels, middle_channels, out_channels, kernel, supernet):
        super(VGGBlock, self).__init__()
        .....
    def forward(self, x):
class SELayer(nn.Module):
    def __init__(self, channel, reduction=16):
        super(SELayer, self).__init__()
        .....
    def forward(self, x):
```

Your objective is to define the optimal choice block for each layer based on the given options above to maximize the model's performance on pathology images. Your response should be the a channel list consisting of N numbers (e.g. [3, 5, ..., 1]).

Fig. 7 | Demonstration of supernet dynamic pretraining and LLM-driven neural architecture search. a Illustration of supernet model for single path one-shot architecture search. During pretraining, numerous subnetworks with independent choice block path are trained via uniform sampling. We search the kernel size of each convolution block in ShuffleNet search for classification and U-Net search for segmentation. We search the depth of transformer layers, number of attention heads, the hidden scale ratio of FFN layer in ViT search for classification. **b** Illustration of prompt template for searching U-Net architectures via LLM recommendation on pathological tasks. The search prompt template include task formulation, network architecture implementation, search space of different variables and LLM response format.

models (U-Net[16]/FPN[17]) and NAS methods (Single path one-shot[19]/AutoFormer[20]/Cream[21]). For CNN-based classification models, we employed ResNet-50 and EfficientNet-b0 implemented by the timm library (https://timm.fast.ai/) as the backbone models. For ViT classification models, we directly finetuned the pretrained weights of Swin Transformer. For segmentation models, we utilized VGGNet[43] as the backbone encoders. For one-shot NAS methods, we referred to three representative methods, contrasting our LLM-driven search strategy against SPOS's vanilla evolutionary search strategy, AutoFormer's weight entanglement strategy, and Cream's architecture distillation strategy.

Loss functions

For classification task, we use the soft target cross entropy loss for training and cross entropy loss for validation. Soft target cross entropy is a softened variant of traditional cross entropy loss function. It is commonly applied in scenarios where targets are soft distributions, corresponding to mixup strategy. Specifically, each class $c \in C$ is assigned with a soft target probability t_c, the modified loss can be formulated as

$$\mathcal{L}_{soft} = -\sum_{c=1}^{C} t_c log(p_c), \tag{11}$$

where p_c denotes the predicted probability of of the sample belonging to class c.

For segmentation tasks, we adopt dice loss that has proved to be effective in numerous literatures. Dice loss measures the overlaps between predicted segmentation results and ground truth. Specifically, given S,G denote the predicted segmentation and ground truth, s_i, g_i denotes the pixel-

level predicted result and ground truth, respectively. N is the number of pixels in image I, dice loss is defined as

$$\mathcal{L}_{soft} = 1 - \frac{2\sum_{i=1}^{N} g_i s_i}{\sum_{i=1}^{N} (g_i)^2 + \sum_{i=1}^{N} (s_i)^2}. \quad (12)$$

Evaluation metrics

We follow the recommended metrics in Metrics Reloaded[44]. Accuracy is the primary metric used in the validation for classification results, while the Dice Coefficient and Intersection over Union (IoU) are adopted in the validation for segmentation results. The Dice Coefficient is calculated by taking twice the intersection of the predicted and ground truth masks divided by the sum of their areas, which is defined as follows

$$Dice = \frac{2 \times |X \cap Y|}{|X| + |Y|}, \quad (13)$$

where X is the set of predicted mask and Y is the set of ground truth mask. IoU, also known as the Jaccard Index, measures the overlap between the predicted mask and the ground truth mask. It is defined as the size of the intersection divided by the size of the union of the masks

$$IoU = \frac{|X \cap Y|}{|X \cup Y|}. \quad (14)$$

A dice score of 1 indicates perfect match, while a score of 0 indicates no overlap. Similarly, an IoU of 1 indicates a perfect match, and an IoU of 0 indicates no overlap.

Algorithm 2. LLM Response Parsing Algorithm (Core Logic)

Input: LLM response text (string)
Output: Instantiated architecture model or core configuration
response_text ← LLM_response
json_str ← ExtractJSONFromText(response_text) {Extract JSON data block}
parsed_json ← AttemptParseJSON(json_str) {Convert JSON string to dictionary}
config_dict ← parsed_json["configuration"] {Access the architecture configuration}
validated_config ← ValidateConfigurationValues(config_dict, expected_search_space) {Validate values against search space definitions}
architecture_model ← InstantiateArchitecture(validated_config) {Create model instance from validated_config}
return architecture_model

Implementation details of LLM-driven NAS

Prompt engineering has been widely leveraged to narrow the gap between pre-training and downstream tasks[45]. In essence, language prompt is constructed to reformulate downstream tasks into the format of pre-training, thereby boosting the zero-shot generation capabilities of LLMs. The prompt template in our experiments is composed of three parts. System prompt tell GPT-4 that he is now an expert in the field of neural architecture search. Role Assignment allows LLM to better understand the background and nuances of the question, leading to more accurate and targeted responses. Content prompt includes task description, implementation details of model architecture, operation candidates, as well as output format. Additionally, experiment prompt with evaluation results will be attached to the content prompt after the initial iteration as supplement materials for LLM decision-making. A detailed prompt example is presented in Fig. 7b.

The complete prompt template used for our LLM-driven Neural Architecture Search is composed of system prompt template shown in Supplementary Fig. 2 and user prompt template shown in Supplementary Fig. 3. This template is used for all architecture search tasks, with task-specific details (i.e., dataset description, search space) modified accordingly. The Algorithm 2 describes how we process the LLM's responses to extract and validate the architecture configurations. An example of a JSON response from GPT-4 for a ShuffleNet architecture search task is presented in Supplementary Fig. 4. For ViT architecture search, the configuration would include different parameters such as embedding dimension, number of layers, number of attention heads, and MLP ratio. An example of a JSON response from GPT-4 for a ShuffleNet architecture search task is presented in Supplementary Fig. 5. For U-Net segmentation architecture search, parameters would include encoder/decoder depth, channel scaling factors, and kernel sizes.

Table 11 | List of abbreviations

Abbreviation	Full form
AI	Artificial Intelligence
LLM	Large Language Model
NAS	Neural Architecture Search
FLOPs	Floating Point Operations
IoU	Intersection over Union
SAM	Segment Anything Model
BCSS	Breast Cancer Semantic Segmentation
PanNuke	Pan-cancer Nuclear Segmentation
TCGA	The Cancer Genome Atlas
WSI	Whole Slide Images
UNP	Universal Task NAS Prompt
RLHF	Reinforcement Learning with Human Feedback
DPO	Direct Preference Optimization
SGD	Stochastic Gradient Descent
CNN	Convolutional Neural Network
ViT	Vision Transformer
FPN	Feature Pyramid Network
SPOS	Single Path One-Shot
MBConv	Mobile inverted Bottleneck Convolution
SE	Squeeze-and-Excitation
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act

Table of abbreviations

For the convenience of the reader, the Table 11 lists major abbreviations used throughout this manuscript, along with their corresponding full forms. This includes frequently used acronyms such as NAS (Neural Architecture Search), LLM (Large Language Model), and FLOPs (Floating Point Operations).

Ethics statement

This study utilized publicly available medical image datasets (e.g., histopathological slides, retina images), detailed with access links in Section "Datasets curation and processing" for transparency and reproducibility. We relied on the ethical approvals (including IRB/ethics committee review and informed consent) and data usage permissions established by the original dataset creators. These datasets were anonymized or de-identified by the original providers in compliance with relevant regulations (e.g., GDPR, HIPAA) prior to public release. Our research involved the use of these pre-existing, de-identified public datasets for computational modeling, did not involve access to personally identifiable information (PII), and no attempts were made to re-identify individuals. This study did not involve new experiments directly on human subjects or animals.

Data availability

The training and validation medical image datasets used in this study, in the form of fundus and histopathology, are publicly available and can be downloaded via the web links. The BreakHis dataset can be obtained from https://www.kaggle.com/datasets/ambarish/breakhis. The Diabetic retinopathy dataset can be obtained from https://www.kaggle.com/datasets/sovitrath/diabetic-retinopathy-224x224-gaussian-filtered. The SkinTumor dataset can be obtained from https://challenge.isic-archive.com/landing/2019/. The Gastric Cancer dataset can be obtained from https://www.kaggle.com/datasets/orvile/gastric-cancer-histopathologytissue-image-dataset. The BCSS dataset can be obtained from https://www.kaggle.com/datasets/whats2000/breast-cancersemantic-segmentation-bcss. The PanNuke dataset can be obtained from https://www.kaggle.com/datasets/andrewmvd/cancerinst-segmentation-and-classification. The ConSep dataset can be obtained from https://paperswithcode.com/dataset/consep. The Zenodo Lung Cancer dataset can be obtained from (https://zenodo.org/records/8368163). The Polyp dataset can be obtained from https://paperswithcode.com/dataset/polypgen.

Code availability

The training script, search script, validation script as well as SPOS model implementation has already been publicly available at our Github (https://github.com/maopopovich/Pathology-NAS).

Received: 2 April 2025; Accepted: 26 September 2025;
Published online: 18 November 2025

References

1. Mousavi, S. E., Ilaghi, M., Elahi Vahed, I. & Nejadghaderi, S. A. Epidemiology and socioeconomic correlates of gastric cancer in Asia: results from the GLOBOCAN 2020 data and projections from 2020 to 2040. *Sci. Rep.* **15**, 6529 (2025).
2. Claudio Quiros, A. et al. Mapping the landscape of histomorphological cancer phenotypes using self-supervised learning on unannotated pathology slides. *Nat. Commun.* **15**, 4596 (2024).
3. Wang, S. et al. Deep learning of cell spatial organizations identifies clinically relevant insights in tissue images. *Nat. Commun.* **14**, 7872 (2023).
4. Ma, J. et al. Segment anything in medical images. *Nat. Commun.* **15**, 654 (2024).
5. Fu, X. et al. BIDCELL: biologically-informed self-supervised learning for segmentation of subcellular spatial transcriptomics data. *Nat. Commun.* **15**, 509 (2024).
6. Achiam, J. et al. Gpt-4 technical report. Preprint at https://arxiv.org/abs/2303.08774 (2023).
7. Li, J., Li, D., Savarese, S. & Hoi, S. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742 (PMLR, 2023).
8. Kirillov, A. et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026 (IEEE, 2023).
9. Huang, Z., Bianchi, F., Yuksekgonul, M., Montine, T. J. & Zou, J. A visual–language foundation model for pathology image analysis using medical Twitter. *Nat. Med.* **29**, 2307–2316 (2023).
10. Lu, M. Y. et al. A visual-language foundation model for computational pathology. *Nat. Med.* **30**, 863–874 (2024).
11. Chen, Z., Yang, H. H., Tay, Y., Chong, K. F. E. & Quek, T. Q. The role of federated learning in a wireless world with foundation models. *IEEE Wirel. Commun.* **31**, 42–49 (2024).
12. Hu, E. J. et al. LoRA: Low-rank adaptation of large language models. International Conference on Learning Representations. https://openreview.net/forum?id=nZeVKeeFYf9 (2022).
13. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778 (IEEE, 2016).
14. Tan, M. & Le, Q. Efficientnet: rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 6105–6114 (PMLR, 2019).
15. Liu, Z. et al. Swin transformer: hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10012–10022 (2021).
16. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-assisted Intervention–MICCAI 2015: 18th International Conference*, 234–241 (Springer, 2015).
17. Lin, T.-Y. et al. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125 (IEEE, 2017).
18. Ci, Y. et al. Evolving search space for neural architecture search. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6659–6669 (IEEE, 2021).
19. Guo, Z. et al. Single path one-shot neural architecture search with uniform sampling. In *Computer Vision–ECCV 2020: 16th European Conference*, 544–560 (Springer, 2020).
20. Chen, M., Peng, H., Fu, J. & Ling, H. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12270–12280 (IEEE, 2021).
21. Peng, H. et al. Cream of the crop: distilling prioritized paths for one-shot neural architecture search. *Adv. Neural Inf. Process. Syst.* **33**, 17955–17964 (2020).
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520 (IEEE, 2018).
23. Hu, J., Shen, L. & Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7132–7141 (IEEE, 2018).
24. Ouyang, L. et al. Training language models to follow instructions with human feedback. *Adv. Neural Inf. Process. Syst.* **35**, 27730–27744 (2022).
25. Rafailov, R. et al. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 36 (NIPS, 2024).
26. Lou, S. et al. A large histological images dataset of gastric cancer with tumour microenvironment annotation for ai. *Sci. Data* **12**, 138 (2025).
27. Pham, T. C. et al. Improving binary skin cancer classification based on best model selection method combined with optimizing full connected layers of Deep CNN. In *2020 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, 1–6 (IEEE, 2020).
28. Ali, S. et al. A multi-centre polyp detection and segmentation dataset for generalisability assessment. *Sci. Data* **10**, 19 (2023).
29. Spanhol, F. A., Oliveira, L. S., Petitjean, C. & Heutte, L. A dataset for breast cancer histopathological image classification. *IEEE Trans. Biomed. Eng.* **63**, 1455–1462 (2016).
30. Wang, Z. & Yang, J. Diabetic retinopathy detection via deep convolutional networks for discriminative localization and visual explanation. In *AAAI Workshops*, 514–521 (AAAI, 2018).
31. Wang, H., Ahn, E. & Kim, J. A dual-branch self-supervised representation learning framework for tumour segmentation in whole slide images. Preprint at https://arxiv.org/abs/2303.11019 (2023).
32. Gamper, J. et al. Pannuke dataset extension, insights and baselines. Preprint at https://arxiv.org/abs/2003.10778 (2020).
33. Graham, S. et al. Hover-Net: simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Med. Image Anal.* **58**, 101563 (2019).
34. Pérez-Cano, J. et al. Combining graph neural networks and computer vision methods for cell nuclei classification in lung tissue. *Heliyon* **10**, e28463 (2024).

35. Li, L. & Talwalkar, A. Random search and reproducibility for neural architecture search. In *Uncertainty in Artificial Intelligence*, 367–377 (PMLR, 2020).
36. Bender, G., Kindermans, P.-J., Zoph, B., Vasudevan, V. & Le, Q. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, 550–559 (PMLR, 2018).
37. Pham, H., Guan, M., Zoph, B., Le, Q. & Dean, J. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*, 4095–4104 (PMLR, 2018).
38. Tan, M. et al. MnasNet: platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2820–2828 (IEEE, 2019).
39. Real, E., Aggarwal, A., Huang, Y. & Le, Q. V. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33, 4780–4789 (AAAI, 2019).
40. Zheng, M. et al. Can gpt-4 perform neural architecture search? Preprint at https://arxiv.org/abs/2304.10970 (2023).
41. Dosovitskiy, A. et al. An image is worth 16x16 words: transformers for image recognition at scale. International Conference on Learning Representations. https://openreview.net/forum?id=YicbFdNTTy (2021).
42. Su, X. et al. ViTAS: vision transformer architecture search. In *European Conference on Computer Vision*, 139–157 (Springer, 2022).
43. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations* (eds, Bengio, Y. & LeCun, Y.) (ICLR, 2015).
44. Maier-Hein, L. et al. Metrics reloaded: recommendations for image analysis validation. *Nat. Methods* **21**, 195–212 (2024).
45. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).

Acknowledgements

This research is funded by National Natural Science Foundation of China (No. 62406347 and No. 62202302 and No. 62572311).

Author contributions

Xiu Su, Qinghua Mao, and Xi Lin wrote the main manuscript text. Zhongze Wu and Xi Lin prepared all the figures. Xiu Su, Shan You, Yue Liao, and Chang Xu were responsible for the study conception (formulation of the research question) and algorithm design. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41746-025-02042-x.

Correspondence and requests for materials should be addressed to Xi Lin.