AI Medicine
https://www.sciltp.com/journals/aim



Article

Check for updates

R-PADE: Ranking-Guided Evolutionary Neural Architecture Search for Medical Image Segmentation

Tian Zhang [1], Jianlun Ma [2], Aohan Mei [2], Nan Li [3], Dong Ge [2,4], Huiyan Jiang [2], Dong Li [4] and Jianhui Lv [5,*]

[1] The Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 518052, China
[2] The College of Software, Northeastern University, Shenyang 110169, China
[3] The School of Computer and Information Technology (School of Big Data), Shanxi University, Taiyuan 030006, China
[4] Neusoft Corporation, Shenyang 110000, China
[5] Multi-modal Data Fusion and Precision Medicine Laboratory, The First Affiliated Hospital of Jinzhou Medical University, Jinzhou 121000, China
* Correspondence: lvjh@jzmu.edu.cn

How To Cite: Zhang, T.; Ma, J.; Mei, A.; et al. R-PADE: Ranking-Guided Evolutionary Neural Architecture Search for Medical Image Segmentation. *AI Medicine* **2026**, *3*(2), 13. https://doi.org/10.53941/aim.2026.100013

Received: 2 July 2026
Revised: 24 October 2026
Accepted: 29 October 2026
Published: 9 November 2026

Abstract: Neural architecture search (NAS) can automate medical image segmentation network design, but costly candidate evaluation limits the supervision available for evolutionary selection. Since replacement decisions depend on relative candidate quality, learning architecture preferences directly can provide guidance under constrained training budgets. In this paper, we propose Ranking-Aware Predictor-Assisted Differential Evolution (R-PADE) for U-Net-style architecture search. Specifically, a predictor trained with pairwise ranking supervision and contrastive regularization directly determines trial–target replacement in discrete differential evolution. Aggregated pairwise preferences prioritize candidates for proxy training, whose observations subsequently update the predictor. This connects evolutionary selection with selective candidate evaluation, reducing reliance on direct training for replacement decisions. Structured encoding and feasibility repair maintain valid architectures during the search. Experiments on CHAOS, BUSI, and ISIC demonstrate competitive segmentation performance with compact architectures. Compared with EU-Net, R-PADE reduces search GPU time by 35.6% while achieving higher Dice scores on CHAOS and BUSI. The ablation experiments compare the implemented predictor formulations and evaluate the contribution of contrastive regularization using matched search settings and proxy-evaluation budgets.

Keywords: neural architecture search; medical image segmentation; differential evolution; ranking predictor

1. Introduction

Recent advances in deep learning have profoundly influenced medical and healthcare artificial intelligence, enabling increasingly capable models for a broad range of clinical tasks, including medical decision support and medical imaging analysis [1,2]. Within medical imaging, segmentation is a particularly important task because it provides pixel-level delineations of anatomical structures and pathological regions across modalities such as MRI, CT, and ultrasound. These delineations support a variety of clinical workflows, including treatment planning, quantitative measurement, and longitudinal assessment. U-Net and its successors have become widely adopted segmentation architectures because of their strong performance on diverse organ- and lesion-oriented tasks [3]. In parallel, transformer-based, graph-based, and diffusion-inspired architectures have been investigated to improve representation capacity and robustness in challenging medical imaging scenarios [4].

Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).
Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Despite this progress, constructing a high-performing segmentation model for a new clinical dataset or imaging modality remains far from "plug-and-play". Segmentation performance is sensitive to architecture-level choices, including stage depth, receptive field, skip connectivity, and feature-fusion strategy, which commonly require expert knowledge and iterative trial-and-error [5, 6]. Such manual optimization is costly, difficult to reproduce, and poorly suited to the substantial heterogeneity of medical images in acquisition protocols, noise, contrast, and anatomical appearance. These challenges motivate automated and task-adaptive architecture design for medical image segmentation [7].

Neural architecture search (NAS) has therefore attracted increasing attention as a means to automate model design for medical imaging [8, 9]. Broadly, a NAS pipeline consists of three components: (i) a search space that specifies allowable architectural building blocks, (ii) a search strategy that explores the space efficiently, and (iii) a performance evaluation procedure that provides feedback for selecting better candidates. In medical image segmentation, NAS has been used to automatically construct U-shaped networks that can rival or surpass manually engineered baselines, thereby reducing the burden of architecture handcrafting. However, applying NAS effectively to segmentation remains difficult, and two key obstacles are commonly observed.

First, segmentation-oriented NAS is computationally intensive because candidate evaluation typically requires non-trivial training on dense prediction tasks. Although surrogate predictors have been introduced to estimate candidate performance without full training, they must be trained from a limited number of evaluated architectures, an issue that is amplified in medical settings where pixel-level annotations are expensive and proxy evaluations can be noisy [10]. As a consequence, prediction errors often translate into unreliable candidate ordering, which directly harms the selection process and leads to sub-optimal architectures [11]. Second, existing medical segmentation NAS methods differ substantially in their search strategies and evaluation mechanisms. Evolutionary approaches, such as EU-Net [12], naturally handle discrete U-Net design choices but may still require many costly candidate evaluations. In contrast, training-free approaches, such as AEG-cTFNAS [13], reduce search-time cost by avoiding conventional proxy training. However, the quality of search guidance can remain unstable when candidate comparisons rely on limited or noisy evaluation signals. Therefore, a central challenge is to preserve reliable relative ordering among candidate architectures under constrained evaluation budgets, since evolutionary replacement depends fundamentally on pairwise comparison rather than perfectly calibrated absolute performance prediction [14].

To address these challenges, we propose Ranking-Aware Predictor-Assisted Differential Evolution (R-PADE), a ranking-guided evolutionary NAS framework for U-Net-style medical image segmentation. Building on ranking-based prediction [11, 14, 15], R-PADE uses learned architecture preferences to guide population replacement and allocate limited proxy evaluations within discrete differential evolution (DE). Specifically, direct pairwise scores determine trial–target replacement, while predicted wins are aggregated to prioritize candidates for proxy training. The resulting validation observations are then used to update the predictor for subsequent search decisions. The contributions of this paper are summarized as follows:

- We develop a ranking-guided DE search procedure in which direct pairwise comparisons determine trial–target replacement, while aggregated predicted wins prioritize proxy evaluations. A shared learned comparison function thus guides both population updates and the allocation of limited resources for candidate training.
- We implement this procedure using the U-Net-style architecture encoding adopted in this work. Discrete mapping and feasibility repair translate arithmetic DE updates into valid choices of operators, skip connections, and network depths, enabling comparison-guided search over executable segmentation architectures.
- We evaluate R-PADE on CHAOS, BUSI, and ISIC through segmentation performance, search-cost, and convergence comparisons. Ablation studies distinguish the effects of pointwise, listwise, and pairwise supervision and examine the contribution of contrastive regularization. Repeated searches on CHAOS and BUSI further assess variability in segmentation performance and the size of the discovered architectures.

2. Related Work

2.1. U-Net and Its Variants for Medical Image Segmentation

As shown in Figure 1, U-Net-style encoder-decoder architectures with skip connections remain the dominant paradigm for medical image segmentation because they fuse high-level semantics with fine-grained spatial details, enabling precise boundary delineation across diverse organs and lesions. However, vanilla convolutional U-Nets are inherently limited in modeling long-range dependencies and global context, which can degrade performance on targets with large scale variation, weak contrast, or complex morphology [16].

To address these limitations, recent U-Net variants mainly evolve along three directions. First, skip-connection redesign and feature fusion are revisited to reduce semantic gaps between encoder and decoder features, including transformer-enhanced skip modeling such as UCTransNet [17]. Second, Transformer/U-shaped hybrids introduce

global self-attention while preserving multi-scale decoding, e.g., TransUNet [18] and UNETR [19], as well as hierarchical Swin-based U-shaped designs (e.g., Swin-Unet [20], Swin UNETR [21]). Third, for deployment-sensitive scenarios, lightweight U-Net variants reduce parameters/compute via efficient convolutions (e.g., depthwise separable designs) while maintaining competitive accuracy [22]. Encoder-decoder designs are also used for image retrieval: AHIR combines ResNet50 feature extraction with autoencoder hashing [23], while ODH-IR integrates object detection with autoencoder-based feature compression [24]. These methods emphasize compact retrieval representations, whereas our U-Net-aligned search space retains multi-scale spatial decoding and explores operator choices and skip connectivity to support pixel-wise medical image segmentation.

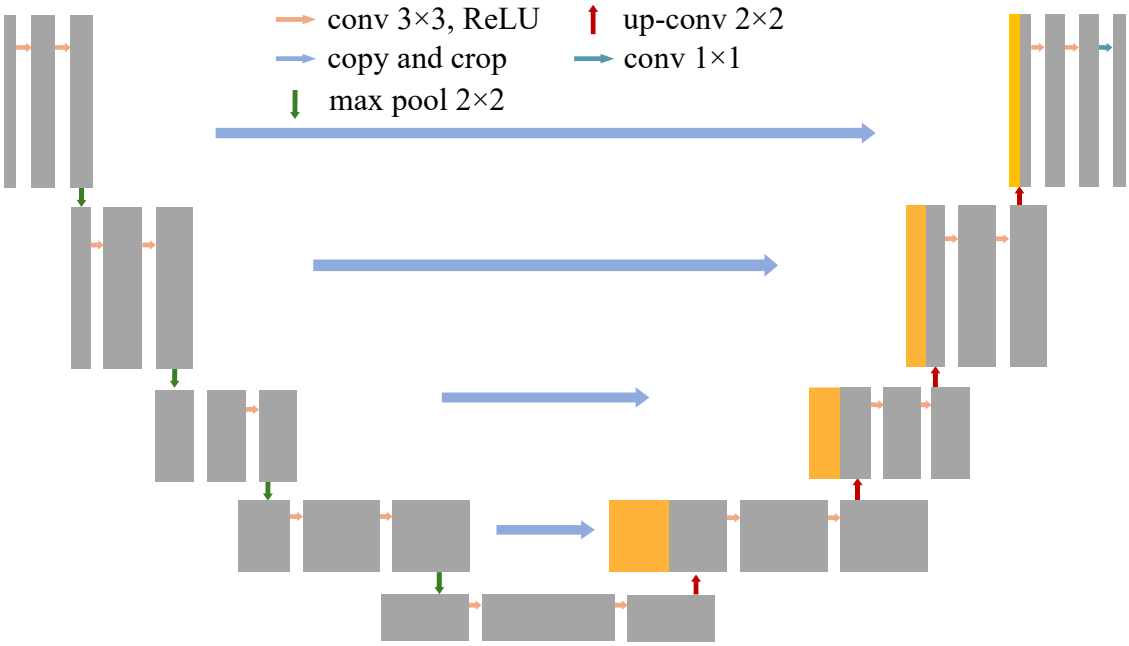


Figure 1. General U-Net Architecture.

2.2. NAS for U-Net-Based Medical Image Segmentation

NAS has been increasingly explored to automate the design of U-Net-based segmentation networks, reducing manual trial-and-error in choosing stage depth, operator sets, and feature-fusion patterns. Most studies retain the encoder-decoder scaffold and search either micro-level blocks/cells (e.g., operator selection within stages) or macro-level topology (e.g., depth/width and skip connectivity), so that the discovered models remain compatible with dense prediction [12].

A key bottleneck is search efficiency, since training and validating segmentation candidates is substantially more expensive than classification. To reduce cost, differentiable and one-shot approaches share weights across candidates and optimize relaxed architecture parameters, enabling scalable exploration under limited budgets [25,26]. In parallel, evolutionary NAS remains appealing for mixed discrete design choices, where constraints and discrete operators are easier to incorporate [27]. For instance, EU-Net leverages differential evolution to search U-Net-like architectures for medical segmentation [12]. More recently, AEG-cTFNAS adopts a compact training-free NAS formulation with an alternating evolution game to reduce search-time evaluation costs [13]. Although these methods improve the practicality of evolutionary NAS, reliable comparison among candidate architectures remains challenging when proxy signals are limited or noisy. Resource-aware formulations are also studied to obtain compact architectures under deployment constraints (e.g., Thrifty NAS), and recent work further investigates reduced-training or training-free signals to accelerate evaluation [28].

Predictor-assisted NAS has explored different forms of ranking supervision. Listwise predictors model relative ordering across candidate sets [11], while transferable relativistic predictors address cross-task cold-start through transferable pairwise supervision [14]. Contrastive ranking prediction has also been investigated for biomedical architecture design [15]. R-PADE focuses on the use of learned comparisons within discrete DE: pairwise scores determine trial–target replacement, and aggregated pairwise wins prioritize proxy evaluations.

3. Our Method

3.1. Overall Framework

An overview of the proposed R-PADE framework is illustrated in Figure 2. R-PADE combines a discrete DE backbone with a ranking predictor trained on proxy-evaluated architectures. The predictor guides trial–target replacement and candidate selection for further proxy training. The resulting validation observations are then used to update the predictor as the search progresses.

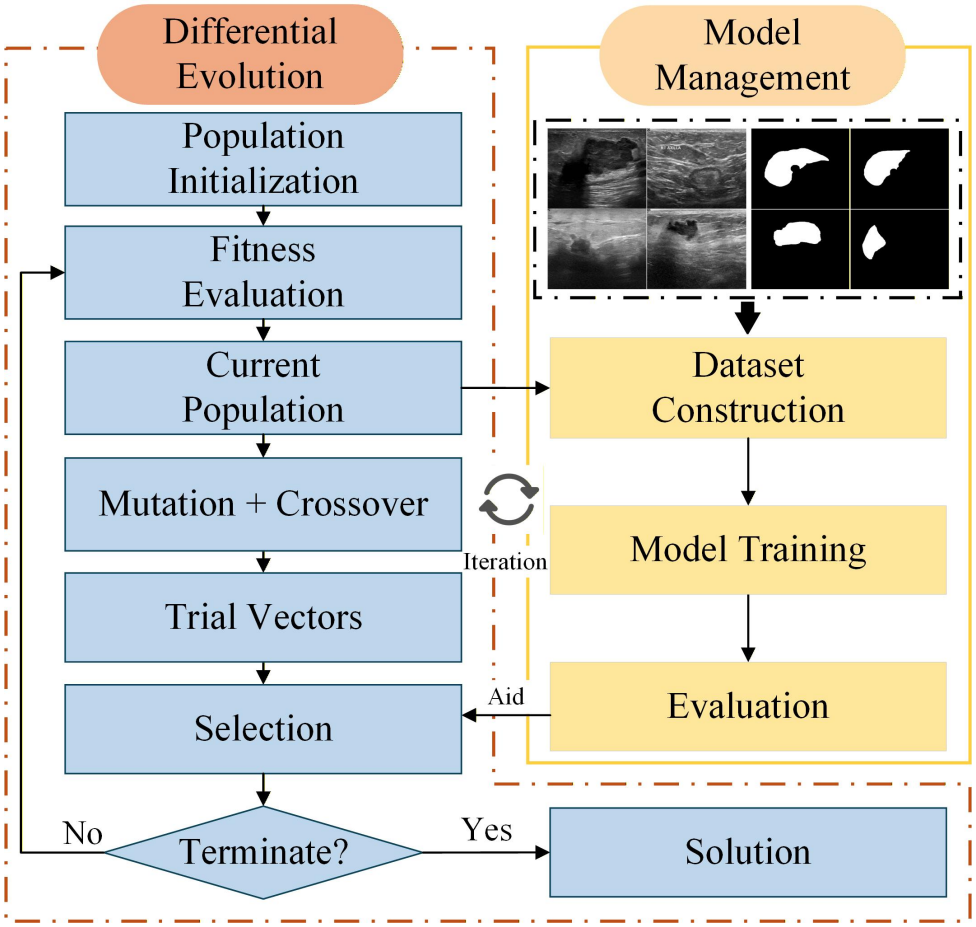


Figure 2. R-PADE framework.

Discrete DE backbone. R-PADE adopts Discrete Differential Evolution with the classic DE/rand/1/bin template as the search backbone. Let x_i^t be the encoded vector of the i-th architecture at generation t. We randomly sample three mutually distinct individuals $x_{r1}^t, x_{r2}^t, x_{r3}^t$ and perform mutation:

$$v_i^t = x_{r1}^t + F \cdot \left(x_{r2}^t - x_{r3}^t\right) \tag{1}$$

This is followed by binomial crossover to form a trial vector u_i^t:

$$u_{i,j}^t = \begin{cases} v_{i,j}^t, & \text{if } \mathrm{rand}(0,1) < CR \text{ or } j = j^*, \\ x_{i,j}^t, & \text{otherwise.} \end{cases} \tag{2}$$

where F and CR are the mutation factor and crossover rate, and j^* enforces at least one mutated dimension.

Discrete mapping & feasibility repair. Since the architecture encoding is discrete (e.g., one-hot operator choices, binary skip indicators, and an integer depth), we convert u_i^t into a valid architecture via: (i) One-hot blocks: apply $\arg\max$ within each operation-choice block to obtain a single active entry. (ii) Binary skip bits: threshold at 0.5 to map to $0, 1$. (iii) Depth d: round to nearest integer and clamp into the allowed range.

If any structural constraint is violated (e.g., inconsistent depth-dependent fields), we apply a minimal feasibility repair by re-clamping and re-normalizing the affected discrete blocks. This step keeps the DE arithmetic update while guaranteeing that each trial corresponds to a legal U-Net-style architecture.

Selection with ranking guidance. For each target architecture x_i^t and its trial architecture u_i^t, the ranking predictor estimates their relative quality. The trial architecture replaces the target when the predicted pairwise score is positive, as defined below:

$$x_i^{t+1} = \begin{cases} u_i^t, & \text{if } f(\mathrm{Enc}(u_i^t) - \mathrm{Enc}(x_i^t)) > 0, \\ x_i^t, & \text{otherwise.} \end{cases} \tag{3}$$

Otherwise, we keep x_i^t. For proxy evaluation, aggregated pairwise wins prioritize a subset of architectures for training. Their validation intersection-over-union (IoU) scores are added to the evaluated archive and used for periodic predictor updates. The learned comparison function therefore guides population replacement without requiring a new proxy-training run for every trial–target pair, while selected evaluations provide observations for subsequent search decisions. This procedure builds on standard DE variation and the adopted ranking-predictor formulation [15].

3.2. Search Space

We adopt a lightweight yet expressive U-Net-aligned search space that promotes architectural diversity while remaining computationally feasible. Following the original design [29,30], the search space includes: convolution operators $C = \{$standard convolution (Conv), depthwise separable convolution (DSC), grouped/depthwise convolution (GDC)$\}$, upsampling operators $U = \{$transposed convolution (TransConv), upsampling followed by convolution (Up + Conv)$\}$, pooling strategies, including adaptive average pooling (AAP) [31] and spatial pyramid pooling (SPP) [32], and a key topological hyperparameter, downsampling depth d (#DS), which controls multi-scale abstraction and receptive field. Receptive-field design is also explored in temporal action segmentation, where MSLID-TCN organizes dilated convolutions to capture short- and long-term dependencies [33]. This provides a cross-task reference for contextual modeling, with temporal receptive fields in MSLID-TCN and two-dimensional spatial receptive fields in the present segmentation search space.

3.3. Encoding

To enable efficient evolution and surrogate modeling, each candidate architecture is encoded into a fixed-length vector aligned with the symmetric encoder–decoder topology. The encoding consists of:

- **Operation selection:** one-hot vectors for each encoder-decoder block pair; for depth d, the operation encoding length is $d \times (|C| + |U|)$.
- **Skip connections:** a d-dimensional binary vector indicating whether the skip connection of each level is enabled.
- **Depth:** the integer d appended to the encoding.

This structured encoding supports direct application of DE arithmetic updates while allowing deterministic discretization back to valid architectures. To ensure a common encoding dimension for architectures with different downsampling depths, operation and skip-connection slots are allocated according to the maximum depth $d_{\max}$, and the inactive levels of shallower architectures are zero-padded. Consequently, $\mathrm{Enc}(A_i) - \mathrm{Enc}(A_j)$ is well defined for any pair of candidate architectures.

3.4. Ranking-Aware Performance Predictor

Following the ranking-predictor formulation in [15], we learn relative preferences from proxy-evaluated architectures. The design is motivated by the decision required during DE selection: given a trial architecture and its corresponding target, the search only needs to determine which member of the pair is preferable, rather than accurately estimating their absolute segmentation performance. Let $\mathcal{D}_e = \{(A_k, q_k)\}$ denote the evaluated architecture archive, where q_k is the proxy-validation IoU of architecture A_k. For a pair (A_i, A_j) sampled from $\mathcal{D}_e$, the preference label is defined as:

$$y_{ij} = \begin{cases} 1, & q_i > q_j, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The predictor $f(\cdot)$ outputs a signed comparison score: a positive score indicates that the first architecture is predicted to outperform the second one. Instead of processing the two architectures independently, the predictor takes their differential encoding as input $\Delta_{ij} = \mathrm{Enc}(A_i) - \mathrm{Enc}(A_j)$.

This representation emphasizes the directed architectural changes from A_j to A_i while cancelling encoding components shared by the two candidates. It also satisfies $\Delta_{ji} = -\Delta_{ij}$, which is consistent with the directional

nature of pairwise preference. For each training pair (A_i, A_j), we additionally include the reversed pair (A_j, A_i) with label $1 - y_{ij}$ to encourage consistent comparison under reversed input order.

Let $\mathcal{P}$ denote the set of ordered training pairs. The predictor is first optimized using the hinge ranking loss:

$$\mathcal{L}_{\text{hinge}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \max(0,\, 1 - (2y_{ij} - 1) f(\Delta_{ij})) \tag{5}$$

The term $2y_{ij} - 1 \in \{-1, 1\}$ specifies the preference direction. Consequently, minimizing $\mathcal{L}_{\text{hinge}}$ encourages the comparison score to have the correct sign and a unit ranking margin. The loss therefore supervises the ordering required by evolutionary replacement without requiring the predictor to reproduce the exact performance difference between two architectures.

To further distinguish confident comparisons from pairs with similar proxy performance, we employ contrastive regularization. Let $b(q_k)$ denote the performance bin assigned to A_k, and define $y'_{ij} = \mathbb{I}[b(q_i) = b(q_j)]$. The contrastive term is:

$$\begin{aligned}\mathcal{L}_{\text{contrast}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \Big[& \quad y'_{ij} \| f(\Delta_{ij}) - f_0 \|_2^2 \\ & + (1 - y'_{ij}) \max\big(0,\, m - \| f(\Delta_{ij}) - f_0 \|_2\big)^2 \Big]\end{aligned} \tag{6}$$

where $f_0 = f(\mathbf{0})$ is the comparison output for two identical encodings and m is the contrastive margin. For architectures in the same performance bin, this term pulls the comparison output toward the neutral reference f_0. For architectures in different bins, it encourages the output to remain at least m away from that reference. Thus, the hinge loss determines the direction of preference, whereas the contrastive term regularizes the magnitude of the comparison relative to proxy-performance similarity. The total training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{hinge}} + \lambda \mathcal{L}_{\text{contrast}} \tag{7}$$

where λ controls the contribution of contrastive regularization.

The trained predictor supports two related but distinct decisions during the search. First, following Equation 3, the trial replaces its target if the comparison score below is positive:

$$f\big(\text{Enc}(A_i^{\text{trial}}) - \text{Enc}(A_i^{\text{target}})\big) \tag{8}$$

Second, we adopt the pairwise win count [15] as a priority score for allocating proxy evaluations:

$$s_k = \sum_{l \neq k} \mathbb{I}[f(\text{Enc}(A_k) - \text{Enc}(A_l)) > 0] \tag{9}$$

Candidates with larger s_k are prioritized for proxy evaluation. Their observed proxy IoUs are subsequently added to $\mathcal{D}_e$ and used to update the predictor. In this manner, the direct comparison score governs local evolutionary replacement, whereas the aggregated win count allocates proxy-evaluation effort, forming a closed loop between pairwise learning, DE selection, and candidate evaluation.

4. Experimental Setup

4.1. Medical Image Datasets and Data Preparation

We evaluate R-PADE in three 2D medical image segmentation settings: liver segmentation from abdominal CT, breast-lesion segmentation from ultrasound, and skin-lesion segmentation from dermoscopic images. A separate architecture search is conducted for each dataset; therefore, the experiments evaluate the applicability of the search procedure across tasks rather than the transfer of one architecture between datasets.

(1) CHAOS: The CHAOS dataset [34] contains 40 CT subjects in total, of which 20 subjects belong to the annotated CT training set. We use these 20 annotated CT training subjects, comprising 2874 axial slices, and exclude the MRI portion. The segmentation target is the liver, and the CT volumes are processed as 2D axial slices to match the search space considered in this work. Data partitioning is performed at the subject level to prevent slices from the same subject from appearing in different subsets. The available subjects are randomly partitioned into training, validation, and test subsets using a 70%/10%/20% ratio. Subject assignments are determined before constructing the 2D slice datasets, and all slices from the same subject belong exclusively to one subset. Consequently, the proportions of slices may differ from the subject-level partitioning ratio. The validation and test subsets are internal holdouts constructed from the annotated training data, rather than the official challenge test set. This setting mainly

evaluates organ segmentation under variations in liver shape, size, and CT appearance.

(2) BUSI: We use the complete BUSI dataset [35], comprising 780 2D grayscale breast ultrasound images from 600 subjects in normal, benign, and malignant categories. The corresponding masks are used for breast-lesion segmentation, with normal cases represented by empty lesion masks. Images are grouped by subject and assigned to training, validation, and test subsets with target image proportions of 70%, 10%, and 20%, respectively. Group-stratified partitioning approximately preserves the proportions of the three diagnostic categories while keeping each subject entirely within one subset. Subject separation takes precedence over exact image counts and category proportions. For images with multiple lesion masks, the masks are combined by their pixel-wise union to obtain a binary lesion mask. The dataset is relevant to our evaluation because ultrasound images exhibit speckle noise, low tissue contrast, and irregular lesion boundaries.

(3) ISIC: We use the 2594 annotated dermoscopic images from the training portion of the ISIC 2018 lesion-segmentation task [36]. The images and their pixel-level masks are used to construct fixed training, validation, and test subsets. The target proportions are 70%, 10%, and 20%, respectively. Images associated with the same subject or lesion, including repeated acquisitions, are grouped before partitioning and assigned to the same subset. The resulting image counts may therefore deviate slightly from the target proportions. Our validation and test subsets are constructed from the annotated training portion and do not correspond to the official challenge validation or test sets. This dataset complements the CT and ultrasound experiments with substantial variations in lesion scale, pigmentation, boundary ambiguity, and background artifacts.

Partition reproducibility. All partitions are generated using a fixed random seed of 42. Group identifiers and their subset assignments are recorded and reused across search runs, ablation experiments, and locally reproduced baselines. Duplicate copies of an image are assigned to the same group. Data augmentation is performed only after partitioning, and augmented samples inherit the subset assignment of their source image. The partition seed is independent of the master seeds controlling the repeated search-and-retraining runs.

Preprocessing and augmentation. CHAOS volumes are processed as single-channel 2D axial slices, and BUSI images retain their grayscale representation. Intensities are normalized independently for each image using min–max normalization:

$$\widetilde{I} = \frac{I - \min(I)}{\max(I) - \min(I)}.$$

Constant-intensity images are mapped to zero. ISIC images retain their three RGB channels, with 8-bit intensities divided by 255. These operations do not estimate normalization statistics from other images or dataset partitions.

After intensity normalization, all images are resized to 256×256 pixels using bilinear interpolation. Ground-truth masks are converted to binary foreground–background maps and resized using nearest-neighbor interpolation to preserve discrete labels. The foreground represents the liver for CHAOS and lesions for BUSI and ISIC. Normal BUSI cases retain empty foreground masks.

During both proxy training and final retraining, augmentation consists of random horizontal flipping with probability 0.5 and random rotation with an angle sampled uniformly from $[-10°, 10°]$. The same spatial transformation is applied to each image and its corresponding mask. Rotations use bilinear interpolation for images, nearest-neighbor interpolation for masks, and zero padding outside the image boundaries. Validation and test data undergo only deterministic normalization and resizing, without random augmentation.

Data usage and model selection. For every dataset, candidate architectures are optimized on the training subset, and their validation IoU values provide the pairwise labels used to update the ranking predictor. The architecture selected at the end of each independent search is retrained from scratch, selected according to validation mean intersection over union (mIoU), and evaluated on the held-out test subset. Final retraining uses only the training subset; the validation subset is not merged into it. The checkpoint with the highest validation mIoU is retained for final test evaluation. Test data are not used for predictor training, architecture selection, or hyperparameter adjustment.

4.2. Search, Training, and Repeated-Run Protocol

Each experiment consists of an architecture-search stage followed by final retraining of the selected architecture. During search, we employ DE/rand/1/bin with a population size of 20 and a maximum of 25 generations. The mutation factor and crossover rate are set to $F = 0.4$ and $CR = 0.7$, respectively. Each candidate is represented using the structured encoding. In each generation, a selected subset of candidates is proxy-trained for 50 epochs under the default setting, and their validation IoU values are used to construct pairwise samples and update the ranking predictor. These proxy observations provide search guidance only and are not reported as final segmentation results. The cumulative candidate-training epochs and GPU time incurred during this stage are reported separately

from the final retraining cost.

After the search terminates, the architecture selected according to validation performance is retrained from scratch for up to 200 epochs. All experiments were implemented in PyTorch version 2.1.0 with Python 3.10. The experiments were conducted on a workstation equipped with an NVIDIA RTX 4090 GPU with 24GB of memory. We use SGD with a momentum of 0.95, an initial learning rate of 0.05, and a batch size of 16. The learning rate is multiplied by 0.1 every 60 epochs. Early stopping is triggered when validation mIoU does not improve for 20 consecutive epochs, and the checkpoint with the highest validation mIoU is retained for testing. The held-out test subset is used only for the final evaluation. For baselines reproduced in our environment, the same dataset split, preprocessing pipeline, and final-retraining protocol are applied.

In the experiments, R-PADE, EU-Net, and AEG-cTFNAS are evaluated through ten independent runs on each of the CHAOS and BUSI datasets. Specifically, each run uses a different random seed and consists of a complete architecture search followed by retraining the selected architecture from scratch. For a fair comparison, the dataset partitions, preprocessing procedures, and final retraining settings are kept the same across all methods and runs, while each baseline follows its own search procedure. The architectures and model checkpoints are selected according to the validation performance, and the test set is used only for the final evaluation. All runs are independent both within each method and across methods. The results are reported as the mean $\pm$ sample standard deviation over the ten complete search and retraining runs. The number of runs is determined according to the computational budget and fixed before evaluation.

The predictor ablation compares Full R-PADE, hinge-only, listwise, and pointwise variants on CHAOS and BUSI using the ten-run protocol above. The search settings, number of proxy-trained candidates, and their per-generation evaluation allocation are matched across variants within each dataset. The proxy-epoch sensitivity experiments reporting mean $\pm$ standard deviation followed a three-run protocol on CHAOS. The ISIC experiment used an architecture obtained through a separate search on ISIC, rather than transferring an architecture from CHAOS or BUSI.

For our evaluations, mIoU is computed from the confusion matrix accumulated over all test images and averaged over the foreground and background classes. Dice is computed for the foreground of each image and then averaged across test images. These different aggregation procedures mean that the identity $\mathrm{IoU} = \mathrm{Dice}/(2 - \mathrm{Dice})$, which holds for metrics derived from the same confusion matrix, does not directly apply to the reported aggregate values.

4.3. Statistical Analysis

In the statistical analysis, each complete architecture search and retraining run is treated as an independent statistical unit. Test-set Dice is used as the primary endpoint and is calculated using the same aggregation procedure as in the main evaluation. R-PADE is compared with EU-Net and AEG-cTFNAS on each dataset. Specifically, an exact two-sided permutation test for independent samples is performed for each comparison, with the absolute difference in mean Dice as the test statistic. Method labels are permuted under the null hypothesis that the run-level score distributions are identical between the two methods.

The mean Dice differences are reported in percentage points, together with pointwise 95% basic-bootstrap confidence intervals calculated from 20,000 resamples. For bootstrap estimation, the run-level scores are resampled independently within each method. These intervals quantify the uncertainty in individual effect estimates, while statistical decisions are based on the Holm-adjusted p values. All statistical inference is conditional on the fixed dataset partitions. The four main-method comparisons form one Holm correction family. The predictor ablation evaluates Full–Hinge, Hinge–Listwise, and Hinge–Pointwise on each dataset using the same statistical procedures. These six contrasts form a separate Holm correction family, with $\alpha = 0.05$ applied within each family.

5. Experimental Results

5.1. Main Results

Table 1 compares R-PADE with representative manually designed segmentation networks and NAS-based methods on CHAOS and BUSI, using the published baseline results [15]. R-PADE improves mIoU, Dice, and mean pixel accuracy (mPA) over both EU-Net and AEG-cTFNAS on the two datasets, while reducing parameter counts by 18.8–21.7% and 4.9–8.3%, respectively. The discovered architectures also have the lowest parameter counts among the compared NAS methods, indicating that the accuracy improvements accompany a reduction in model size.

On CHAOS, R-PADE achieves 97.63% mIoU, 95.76% Dice, and 99.10% mPA with 8.22 M parameters, obtaining the highest scores across all three metrics among the NAS-based methods. Relative to EU-Net and AEG-cTFNAS, the Dice improvements are 0.64 and 0.45 percentage points, respectively. The comparison with manually designed networks reveals a different balance between segmentation accuracy and model size. U-Net++

with a ResNet backbone achieves a slightly higher mIoU of 97.80%, whereas R-PADE improves Dice from 94.90% to 95.76% and reduces the parameter count from 67.98 M to 8.22 M. Thus, the 0.17-percentage-point difference in mIoU accompanies a 0.86-percentage-point Dice improvement and an 87.9% parameter reduction.

On BUSI, R-PADE obtains 87.79% mIoU, 69.63% Dice, and 93.52% mPA with 7.92 M parameters. Its Dice gains over EU-Net and AEG-cTFNAS increase to 1.82 and 0.89 percentage points, respectively, and its mPA is the highest among all compared methods. U-Net++ with an Inception backbone achieves a higher mIoU of 88.33%, but its Dice score is lower at 67.50% and its parameter count is substantially larger at 59.36 M. Relative to this model, R-PADE improves Dice by 2.13 percentage points with 86.7% fewer parameters. LAEDNet-L achieves the highest Dice of 75.90%, although it requires 63.25 M parameters compared with 7.92 M for R-PADE. These comparisons identify model compactness as a consistent advantage of R-PADE, alongside improved segmentation performance over EU-Net and AEG-cTFNAS.

Table 1. Comparison of R-PADE with representative manually designed and NAS-based methods on the CHAOS and BUSI datasets. Baseline results are taken from the published comparison in [15].

Category	Algorithm	Backbone	CHAOS			BUSI			#Param (M)
			mIoU (%)	Dice (%)	mPA (%)	mIoU (%)	Dice (%)	mPA (%)	
Manually	U-Net	–	91.90	96.60	95.42	82.10	66.50	87.79	34.53
		ResNet	96.97	93.50	98.92	82.88	65.10	88.70	51.51
		DenseNet	96.29	95.30	98.92	75.91	66.80	91.46	21.20
		Inception	97.76	95.00	99.02	83.77	68.60	92.91	48.79
	R2U-Net	–	82.40	42.40	88.20	69.61	27.40	80.22	39.09
	AttU-Net	–	91.30	96.70	94.00	82.05	65.60	87.59	34.88
	U-Net++	ResNet	97.80	94.90	99.01	86.80	64.10	93.30	67.98
		DenseNet	97.75	95.00	99.07	84.85	69.50	93.40	38.85
		Inception	97.67	94.60	99.07	88.33	67.50	92.70	59.36
	PSPNet	ResNet	96.14	93.80	98.08	86.76	57.90	90.48	2.42
		DenseNet	97.05	94.90	98.41	84.02	63.50	90.67	2.23
		Inception	97.56	92.90	98.77	85.14	63.90	92.17	2.11
	MANet	ResNet	96.63	94.10	98.15	81.05	64.50	91.34	166.43
		DenseNet	95.01	95.60	98.13	84.55	59.90	90.24	99.24
		Inception	95.97	95.70	98.34	83.33	65.40	88.90	114.19
	DeepLabV3+	ResNet	97.31	95.00	98.95	84.61	66.10	92.04	45.67
NAS-based	MixSearch (k=2) [37]	–	97.02	94.96	98.88	85.92	67.08	92.64	12.79
	LAEDNet-L [38]	–	96.48	94.41	98.32	71.00	75.90	92.40	63.25
	EU-Net [12]	–	97.18	95.12	98.93	86.34	67.81	92.86	10.12
	AEG-cTFNAS [13]	–	97.34	95.31	98.99	87.08	68.74	93.18	8.64
Ours	R-PADE (CHAOS)	–	97.63	95.76	99.10	–	–	–	8.22
	R-PADE (BUSI)	–	–	–	–	87.79	69.63	93.52	7.92

Table 2 presents stage-wise architecture specifications for CHAOS and BUSI, including convolution operators, downsampling and upsampling operations, channel widths, and skip connections. Each stage pairs an encoder block E_ℓ with a decoder block D_ℓ at the same spatial resolution. The paired blocks use the same convolution operator family with independent weights. Stages are ordered from the highest to the lowest resolution, and d denotes the number of downsampling operations.

Table 2. Final architectures obtained by R-PADE on CHAOS and BUSI. Skip: 1 = enabled; 0 = disabled.

Stage	Resolution	Channels	Convolution	Downsampling	Upsampling	Skip
CHAOS: $d = 4$, $\boldsymbol{s} = (1, 1, 1, 0)$						
E_1/D_1	256×256	32	Conv	AAP	TransConv	1
E_2/D_2	128×128	64	DSC	AAP	Up + Conv	1
E_3/D_3	64×64	128	Conv	AAP	Up + Conv	1
E_4/D_4	32×32	256	DSC	AAP	TransConv	0
Bottleneck	16×16	512	Conv	–	–	–
BUSI: $d = 3$, $\boldsymbol{s} = (1, 1, 1)$						
E_1/D_1	256×256	32	DSC	AAP	Up + Conv	1
E_2/D_2	128×128	64	Conv	AAP	TransConv	1
E_3/D_3	64×64	128	DSC	AAP	Up + Conv	1
Bottleneck	32×32	256	Conv	–	–	–

The CHAOS configuration uses four downsampling operations and reaches a 16×16 bottleneck, retaining

skip connections at the three higher-resolution stages. The BUSI configuration uses three downsampling operations, reaches a 32×32 bottleneck, and retains all three skip connections. In both configurations, AAP uses a 2×2 window with stride 2. TransConv uses a 2×2 transposed convolution with stride 2 and padding 0, whereas Up + Conv combines bilinear upsampling by a factor of two with a 1×1 convolution. Both upsampling operators produce the channel width of the receiving decoder stage before skip fusion. The bottleneck uses a conventional convolution block, and the output of D_1 is mapped to a single foreground probability channel through a 1×1 convolution and sigmoid activation.

Figure 3 presents a visual comparison of segmentation results for four representative test cases from the CHAOS CT dataset. The first row shows the input CT images, the second row presents the corresponding ground-truth masks, and the third row displays the predictions generated by R-PADE. As observed, the predicted masks generally agree with the ground truth in liver shape and spatial extent. In particular, R-PADE captures the main liver contours, although localized discrepancies remain along some boundary segments, especially near concave regions. These visual results complement the quantitative evaluation by illustrating both the agreement of the predicted masks with the annotations and the locations of residual boundary errors in the selected cases.

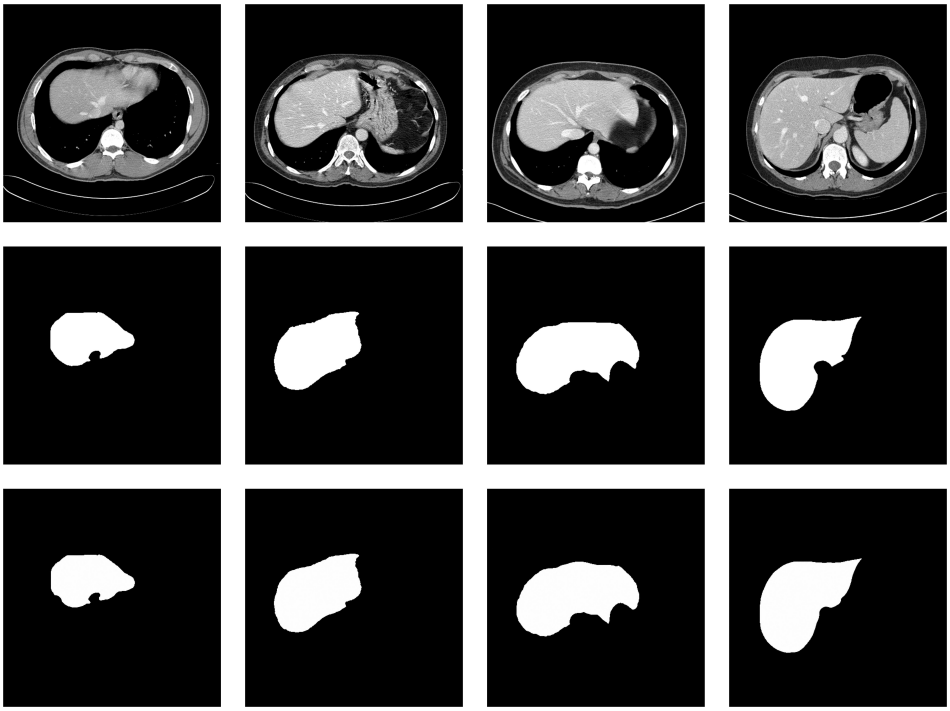


Figure 3. Qualitative liver segmentation results of R-PADE on four representative test cases from the CHAOS CT dataset.

5.2. Generalization Analysis on ISIC

To further evaluate the generalization capability of R-PADE across different medical image segmentation scenarios, we conduct an additional experiment on the ISIC 2018 skin lesion segmentation dataset. In contrast to the abdominal CT images in CHAOS and the breast ultrasound images in BUSI, ISIC contains dermoscopic images with substantial variations in lesion appearance, color distribution, boundary ambiguity, and surrounding artifacts. The architecture for ISIC is independently searched using the same R-PADE procedure rather than transferred from the other datasets.

As shown in Table 3, R-PADE achieves the best performance among the compared methods, obtaining 83.47% mIoU, 90.58% Dice, and 94.61% mPA with only 8.05 M parameters. Compared with EU-Net, R-PADE improves mIoU, Dice, and mPA by 1.36%, 1.02%, and 0.58%, respectively. It also outperforms AEG-cTFNAS by 0.53% in mIoU, 0.44% in Dice, and 0.30% in mPA, while using fewer parameters. These results indicate that the proposed ranking-aware predictor-assisted search strategy can remain effective beyond the CT and ultrasound settings, enabling the discovery of compact and accurate architectures for dermoscopic skin lesion segmentation.

Table 3. Generalization comparison on the ISIC 2018 skin lesion segmentation dataset.

Method	mIoU (%)	Dice (%)	mPA (%)	#Param (M)
U-Net++	81.26	89.01	93.72	67.98
EU-Net [12]	82.11	89.56	94.03	10.12
AEG-cTFNAS [13]	82.94	90.14	94.31	8.64
R-PADE	83.47	90.58	94.61	8.05

5.3. Statistical Stability Analysis

The Dice results of R-PADE, EU-Net, and AEG-cTFNAS over ten independent runs are shown in Table 4. For each baseline, ΔDice denotes the mean Dice improvement of R-PADE, calculated as the mean Dice of R-PADE minus that of the baseline. The corresponding 95% confidence intervals and Holm-adjusted p values are also reported.

As can be seen in Table 4, R-PADE achieves the highest mean Dice among the three methods on both datasets. Specifically, on CHAOS, R-PADE obtains a mean Dice of 95.708%, outperforming EU-Net and AEG-cTFNAS by 0.496 and 0.435 percentage points, respectively. On BUSI, R-PADE achieves a mean Dice of 69.493%, exceeding the two baselines by 2.039 and 0.992 percentage points, respectively. These improvements are larger than those on CHAOS, showing a greater performance advantage on BUSI. Moreover, all four comparisons remain statistically significant after Holm adjustment ($p < 0.001$), and all pointwise 95% confidence intervals for the mean differences lie above zero.

Table 4. Dice results over ten independent runs and comparisons with R-PADE.

Dataset	Method	Dice (%)	ΔDice	95% CI	Raw p	Holm-Adjusted p
CHAOS	R-PADE	95.708 ± 0.147	–	–	–	–
	EU-Net [12]	95.212 ± 0.149	0.496	[0.371, 0.616]	1.083×10^{-5}	4.330×10^{-5}
	AEG-cTFNAS [13]	95.273 ± 0.227	0.435	[0.273, 0.586]	3.248×10^{-5}	6.495×10^{-5}
BUSI	R-PADE	69.493 ± 0.361	–	–	–	–
	EU-Net [12]	67.454 ± 0.321	2.039	[1.763, 2.331]	1.083×10^{-5}	4.330×10^{-5}
	AEG-cTFNAS [13]	68.501 ± 0.585	0.992	[0.589, 1.389]	3.248×10^{-4}	3.248×10^{-4}

Dice is reported as mean $\pm$ sample SD ($n = 10$). ΔDice denotes R-PADE minus the corresponding baseline; differences and confidence interval limits are expressed in percentage points. ”–” denote entries that are not applicable to the reference method, R-PADE.

In terms of run-to-run variation, R-PADE has the lowest observed sample standard deviation on CHAOS, with a value of 0.147, compared with 0.149 for EU-Net and 0.227 for AEG-cTFNAS. On BUSI, the standard deviation of R-PADE is 0.361, which is lower than that of AEG-cTFNAS (0.585) but higher than that of EU-Net (0.321). Thus, R-PADE exhibits lower observed variability than AEG-cTFNAS on both datasets, while its comparison with EU-Net depends on the dataset. The results show that the higher mean Dice of R-PADE does not imply uniformly lower run-to-run variability. These findings concern repeated architecture search and retraining under the fixed dataset partitions.

5.4. Search Efficiency Analysis

Table 5 compares cumulative search-stage training epochs, GPU hours, and final segmentation performance. Training epochs are accumulated across all candidate-network evaluations within a complete search, and final retraining is excluded from the reported search costs. Epoch counts complement GPU hours by making the candidate-training workload explicit.

Table 5. Search-stage computational cost and final segmentation performance. GPU hours exclude the final retraining of the selected architectures.

Method	NAS Type	Proxy Evaluation	Total Search-Stage Training Epochs	GPU Hours	CHAOS Dice (%)	BUSI Dice (%)
EU-Net [12]	Evolutionary	Reduced training	5175	41.8	95.12	67.81
AEG-cTFNAS [13]	Training-free	None	0	12.6	95.31	68.74
R-PADE	Predictor-assisted evolutionary	50-epoch proxy	3150	26.9	95.76	69.63

AEG-cTFNAS adopts training-free evaluation and therefore requires zero candidate-training epochs, although computing its evaluation scores still incurs computational cost. R-PADE uses 50 epochs per proxy evaluation. The cumulative training budgets are 5175 epochs for EU-Net and 3150 epochs for R-PADE, with the latter

corresponding to 63 proxy-training evaluations. These budgets should be interpreted alongside GPU hours because the computational cost per epoch varies across evaluation settings.

R-PADE requires 26.9 GPU hours, approximately 35.6% less than EU-Net's 41.8 GPU hours, while achieving higher Dice scores of 95.76% on CHAOS and 69.63% on BUSI. AEG-cTFNAS has the lowest search cost at 12.6 GPU hours. Relative to this training-free baseline, R-PADE incurs an additional 14.3 GPU hours and improves Dice by 0.45 and 0.89 percentage points on CHAOS and BUSI, respectively.

5.5. Evolutionary Process Analysis

Figures 4 and 5 illustrate the evolutionary trajectories of R-PADE on the CHAOS and BUSI datasets, respectively. On CHAOS, the segmentation performance improves rapidly during the early generations, and the best individuals converge to high-quality architectures after approximately 10 generations. Meanwhile, the average population performance increases steadily and stabilizes in the later search stage, indicating that R-PADE can progressively improve the overall quality of the candidate population.

On BUSI, the search process exhibits a more gradual improvement because breast ultrasound lesion segmentation involves weaker boundaries, higher appearance variability, and stronger image noise. Nevertheless, the best, average, and worst population performances all improve consistently over generations, demonstrating that R-PADE remains effective in the more challenging ultrasound setting.

To further examine the contribution of ranking-aware predictor assistance, Figure 6 compares the convergence behavior of R-PADE and vanilla DE on CHAOS (a) and ISIC (b), using the same population size and number of generations within each dataset. The curves track best-so-far validation Dice throughout the search. On CHAOS, R-PADE discovers higher-performing candidate architectures earlier and maintains higher validation Dice during most of the search, supporting the effectiveness of ranking-aware guidance in candidate selection. The additional ISIC comparison evaluates whether this convergence advantage extends to another dataset.

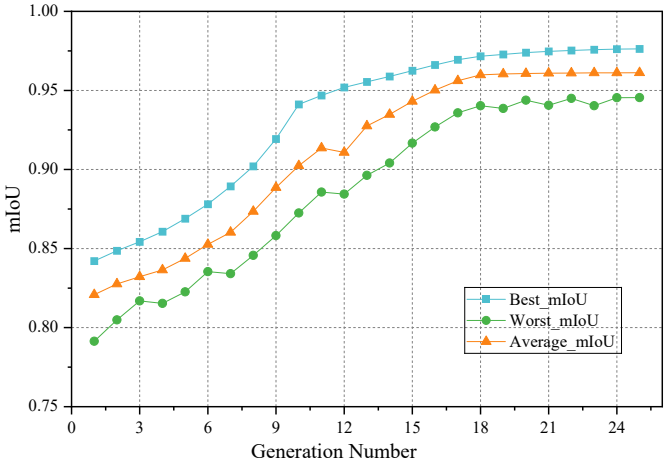


Figure 4. Evolutionary trajectory of segmentation performance (mIoU) across 25 generations on CHAOS.

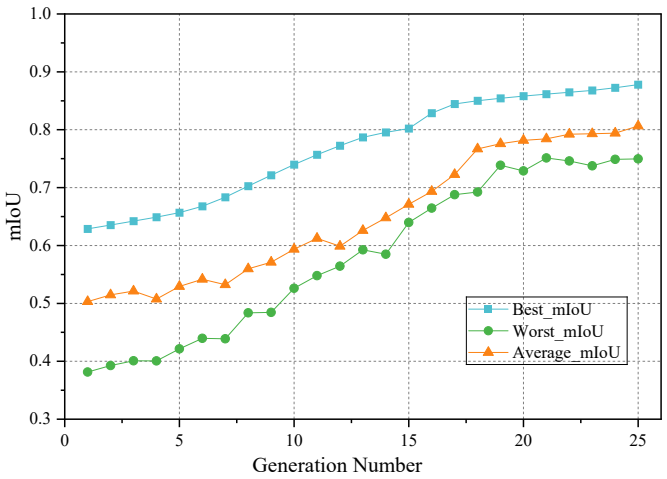
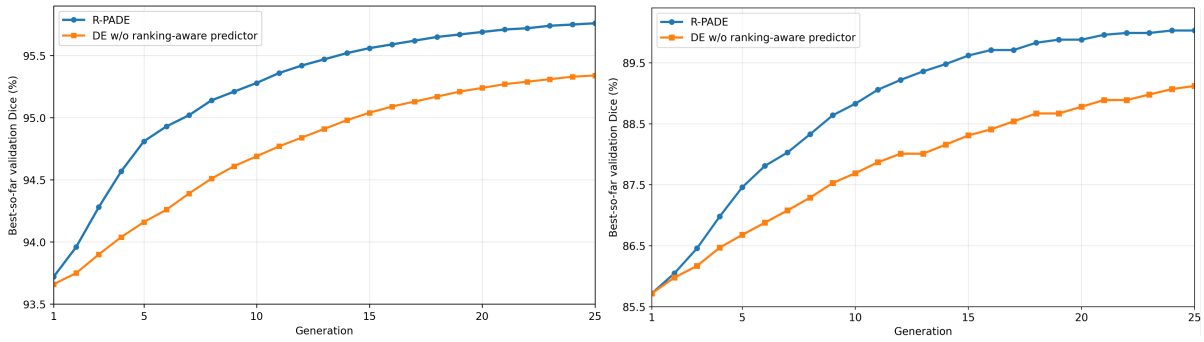


Figure 5. Evolutionary trajectory of segmentation performance (mIoU) across 25 generations on BUSI.



(a) CHAOS (b) ISIC

Figure 6. Convergence comparison between R-PADE and vanilla DE on (**a**) CHAOS and (**b**) ISIC under the same search space, population size, and number of generations within each dataset.

5.6. Ablation Study

In the predictor ablation study, four configurations are compared on CHAOS and BUSI, namely Full R-PADE, Hinge, Listwise, and Pointwise. Specifically, Full R-PADE combines pairwise hinge supervision with contrastive regularization. The Hinge variant sets $\lambda = 0$ to remove the contrastive term while keeping the remaining pairwise predictor settings unchanged. The Listwise and Pointwise variants adopt listwise ranking and pointwise performance regression, respectively, without contrastive regularization. For each variant, the predictor implementation and training settings are kept the same across the two datasets, while the predictor is trained separately on each dataset.

To examine the contribution of contrastive regularization within the pairwise predictor, Full R-PADE is compared with Hinge. In addition, Hinge is compared with Listwise and Pointwise to evaluate the implemented predictor formulations in the absence of contrastive regularization. All four configurations follow the ten-run architecture search and retraining protocol, with test-set Dice used as the primary evaluation metric.

As can be seen in Table 6, the predictor variants exhibit modest absolute differences in mean Dice. Full R-PADE achieves higher mean Dice than Hinge on both datasets, with improvements of 0.300 and 0.137 percentage points on CHAOS and BUSI, respectively. Among the variants without contrastive regularization, Hinge obtains a mean Dice that is 0.166 percentage points lower than that of Listwise on CHAOS but 0.004 percentage points higher on BUSI. Hinge also achieves higher mean Dice than Pointwise, with improvements of 0.260 percentage points on CHAOS and 0.363 percentage points on BUSI.

The corresponding uncertainty estimates and statistical comparisons are presented in Table 7. After Holm correction, the improvement of Full R-PADE over Hinge is statistically significant on CHAOS ($p = 0.016432$), but not on BUSI ($p = 0.864362$). Nevertheless, the gain on CHAOS is modest in absolute Dice, and its practical importance cannot be inferred from statistical significance alone. The four comparisons of Hinge with Listwise and Pointwise all have Holm-adjusted p values above 0.05. These results provide evidence for a modest benefit from contrastive regularization on CHAOS, but do not establish a consistent advantage of pairwise hinge supervision over the implemented alternatives.

We further examine the sensitivity of R-PADE to the proxy-training budget on CHAOS by comparing 30, 50, and 70 epochs per candidate, while keeping the remaining search settings and final retraining protocol unchanged. As shown in Table 8, increasing the budget from 30 to 50 epochs improves mean Dice from 95.40% to 95.71%. Extending training to 70 epochs provides a further gain of 0.04 percentage points, while increasing search cost from 26.9 to 35.6 GPU hours. This limited additional performance gain relative to the increased computational cost supports 50 epochs as a practical accuracy–cost trade-off among the evaluated budgets on CHAOS.

Table 6. Predictor ablation results on CHAOS and BUSI.

Variant	Predictor Objective	CHAOS Dice (%)	BUSI Dice (%)
Full R-PADE	Pairwise + contrastive	95.708 ± 0.147	69.493 ± 0.361
Hinge	Pairwise hinge	95.408 ± 0.218	69.356 ± 0.408
Listwise	Listwise ranking	95.574 ± 0.182	69.352 ± 0.383
Pointwise	Performance regression	95.148 ± 0.254	68.993 ± 0.580

Table 7. Statistical comparisons for the predictor ablation.

Dataset	Comparison ($A - B$)	ΔDice	95% CI	Raw p	Holm-Adjusted p
CHAOS	Full – Hinge	0.300	[0.1491, 0.4576]	0.002739	0.016432
	Hinge – Listwise	−0.166	[−0.3292, 0.0004]	0.081123	0.324493
	Hinge – Pointwise	0.260	[0.0574, 0.4540]	0.023945	0.119725
BUSI	Full – Hinge	0.137	[−0.1790, 0.4610]	0.432181	0.864362
	Hinge – Listwise	0.004	[−0.3289, 0.3284]	0.981933	0.981933
	Hinge – Pointwise	0.363	[−0.0644, 0.7657]	0.122811	0.368432

Table 8. Sensitivity to proxy-training epochs on CHAOS.

Proxy Epochs	mIoU (%)	Dice (%)	Total Proxy Epochs	GPU Hours
30	97.35 ± 0.16	95.40 ± 0.20	1890	18.4 ± 0.4
50	97.59 ± 0.11	95.71 ± 0.13	3150	26.9 ± 0.5
70	97.62 ± 0.10	95.75 ± 0.12	4410	35.6 ± 0.7

6. Discussion

The methodological contribution of R-PADE lies in explicitly coordinating population replacement and proxy-evaluation allocation through a shared learned comparison function within discrete architecture search. These decisions serve distinct purposes: population replacement maintains the evolving population, while proxy evaluation supplies observations for predictor learning. The procedure builds on standard evolutionary operators, ranking supervision, and feasibility repair.

The ablation study on CHAOS and BUSI examines the contribution of contrastive regularization separately from the comparison of predictor formulations. Pairwise supervision aligns with the trial-target replacement decisions in DE, providing a rationale for incorporating learned comparisons into evolutionary replacement and selective proxy evaluation. This alignment does not, however, establish empirical superiority over listwise ranking or pointwise regression. The modest Dice differences and their uncertainty limit the conclusions that can be drawn about individual components.

The experiments on CHAOS, BUSI, and ISIC cover CT, ultrasound, and dermoscopic imaging, respectively, with architecture search and predictor training performed separately on each dataset. These experiments evaluate dataset-specific architecture search and do not assess the transfer of a trained architecture or predictor to an unseen dataset. The component comparisons are conducted on CHAOS and BUSI using the same 2D U-Net-style search space and DE backbone. Consequently, the results do not establish applicability to other search optimizers, substantially different network topologies, or volumetric segmentation. R-PADE is also designed to optimize the balance between search cost and final segmentation quality, rather than minimizing computational cost alone. Training-free NAS methods may incur lower search overhead, while R-PADE relies on proxy training to collect validation observations for predictor learning. The reported segmentation performance and search cost reflect the overall search procedure and do not separately quantify the contribution of an individual predictor objective.

The present study is limited to 2D U-Net-style architectures. Extending R-PADE to 3D would require volumetric operators and revised encoding and feasibility-repair rules to accommodate anisotropic downsampling and 3D feature compatibility. Increased memory and training demands would also require reassessing patch size, batch size, and the 50-epoch proxy budget. Smaller patches or reduced resolution may change candidate rankings and bias evolutionary selection. The predictor would therefore require 3D training observations and validation of ranking agreement between proxy evaluations and fully retrained architectures.

These challenges are reflected in DiNTS and C2FNAS, which address memory-constrained search and search-deployment inconsistencies, respectively [39,40]. The current results do not establish R-PADE's performance relative to 3D-NAS methods; such comparisons would require consistent volumetric evaluation protocols and joint reporting of segmentation accuracy, search time, and peak memory. Other limitations include restricted operator families, task-specific predictor training, and limited cohort scale. Such dynamical models serve a different computational purpose from the spatial feature-extraction operators considered in our current segmentation search space. Broader search spaces, transferable predictors, and multi-center validation remain directions for future work.

7. Conclusions

This paper presented R-PADE, a ranking-aware predictor-assisted differential evolution framework for efficient neural architecture search in U-Net-style medical image segmentation. Unlike conventional predictor-assisted NAS methods that estimate the absolute performance of individual architectures, R-PADE formulates surrogate guidance as a pairwise ranking problem and learns the relative quality between candidate architectures. The pairwise formulation aligns with the trial–target comparisons in differential evolution, allowing learned preferences to directly inform evolutionary replacement. R-PADE combines contrastive ranking prediction, structured discrete architecture encoding, feasibility-aware repair, and comparison-driven DE updates to guide replacement decisions and prioritize proxy evaluations within a discrete U-Net search space.

Experiments on CHAOS, BUSI, and ISIC demonstrate that R-PADE can discover compact architectures with competitive or superior segmentation performance across CT, ultrasound, and dermoscopic imaging scenarios. The repeated-run experiments and the convergence, efficiency, and ablation analyses evaluate search performance, computational cost, and run-to-run variability within the examined 2D U-Net-style search space and DE framework. Future work will extend R-PADE to 3D and multi-organ segmentation settings and investigate transferable ranking predictors for more robust cross-dataset architecture search.

Author Contributions

T.Z. and J.M.: conceptualization, methodology; A.M. and N.L.: writing—original draft preparation, visualization; H.J. and D.G.: methodology; D.L. and J.L.: supervision. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

This study involved secondary analysis of the publicly available, de-identified CHAOS, BUSI, and ISIC 2018 datasets [34–36]. Ethical review and approval were waived for this study, as no new participants were recruited, no clinical interventions were performed, and no directly identifying personal information was collected. The original data sources and data-use conditions are documented in the corresponding dataset publications and repositories. The study was conducted in accordance with the applicable principles of the Declaration of Helsinki.

Informed Consent Statement

Not applicable.

This study used publicly available, de-identified data from the CHAOS, BUSI, and ISIC 2018 datasets and involved no direct contact with participants. The abdominal CT images and associated segmentation masks presented in this article contain no directly identifying personal information.

Data Availability Statement

Datasets in this study are all public.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used GPT-5.5 to refine and polish the language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published paper.

References

1. Zhang, T.; Cheng, J.; Miao, L.; et al. Multi-Hop Reasoning with Relation Based Node Quality Evaluation for Sparse Medical Knowledge Graph. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *9*, 1805–1816. https://doi.org/10.1109/tetci.2024.3452748.
2. Wang, M.; Jiang, H.; Shi, T.; et al. SCL-Net: Structured Collaborative Learning for PET/CT Based Tumor Segmentation. *IEEE J. Biomed. Health Informatics* **2022**, *27*, 1048–1059. https://doi.org/10.1109/jbhi.2022.3226475.

3. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*; Springer: Cham, Switzerland, 2015; pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.
4. Wang, M.; Jiang, H.; Shi, T.; et al. HD-RDS-UNet: Leveraging Spatial-Temporal Correlation Between the Decoder Feature Maps for Lymphoma Segmentation. *IEEE J. Biomed. Health Informatics* **2021**, *26*, 1116–1127. https://doi.org/10.1109/jbhi.2021.3102612.
5. Li, N.; Ma, L.; Xing, T.; et al. Automatic design of machine learning via evolutionary computation: A survey. *Appl. Soft Comput.* **2023**, *143*, 110412. https://doi.org/10.1016/j.asoc.2023.110412.
6. Guo, L.; Li, N.; Zhang, T. EEG-based emotion recognition via improved evolutionary convolutional neural network. *Int. J. Bio-Inspired Comput.* **2024**, *23*, 203–213. https://doi.org/10.1504/ijbic.2024.139268.
7. Zhang, T.; Ma, L.; Cheng, S.; et al. Automatic prompt design via particle swarm optimization driven LLM for efficient medical information extraction. *Swarm Evol. Comput.* **2025**, *95*, 101922. https://doi.org/10.1016/j.swevo.2025.101922.
8. Ma, L.; Li, N.; Yu, G.; et al. Pareto-Wise Ranking Classifier for Multiobjective Evolutionary Neural Architecture Search. *IEEE Trans. Evol. Comput.* **2023**, *28*, 570–581. https://doi.org/10.1109/tevc.2023.3314766.
9. Ma, L.; Li, N.; Zhu, P.; et al. A Novel Fuzzy Neural Network Architecture Search Framework for Defect Recognition with Uncertainties. *IEEE Trans. Fuzzy Syst.* **2024**, *32*, 3274–3285. https://doi.org/10.1109/tfuzz.2024.3373792.
10. Ma, L.; Kang, H.; Yu, G.; et al. Single-Domain Generalized Predictor for Neural Architecture Search System. *IEEE Trans. Comput.* **2024**, *73*, 1400–1413. https://doi.org/10.1109/tc.2024.3365949.
11. Li, N.; Ma, L.; Wang, R.; et al. Listwise ranking predictor for evolutionary neural architecture search. *Swarm Evol. Comput.* **2025**, *96*, 101956. https://doi.org/10.1016/j.swevo.2025.101956.
12. Yu, C.; Wang, Y.; Tang, C.; et al. EU-Net: Automatic U-Net neural architecture search with differential evolutionary algorithm for medical image segmentation. *Comput. Biol. Med.* **2023**, *167*, 107579. https://doi.org/10.1016/j.compbiomed.2023.107579.
13. Sun, X.X.; Wang, H.; Song, P.C. Compact Training-Free NAS with Alternating Evolution Game for Medical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Daejeon, Republic of Korea, 23–27 September 2025; pp. 108–118. https://doi.org/10.1007/978-3-032-05325-1_11.
14. Li, N.; Xue, B.; Ma, L.; et al. Transferable Relativistic Predictor: Mitigating Cross-Task Cold-Start Issue in NAS. In Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 16–22 August 2025; pp. 5625–5633. https://doi.org/10.24963/ijcai.2025/626.
15. Li, N.; He, M.; Mei, A.; et al. Ranking-Aware Predictor-Assisted Automatic Architecture Design for Biomedical Image Segmentation. In Proceedings of the 2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Wuhan, China, 15–18 December 2025; pp. 1–8. https://doi.org/10.1109/bibm66473.2025.11389832.
16. Punn, N.S.; Agarwal, S. Modality specific U-Net variants for biomedical image segmentation: a survey. *Artif. Intell. Rev.* **2022**, *55*, 5845–5889. https://doi.org/10.1007/s10462-022-10152-1.
17. Wang, H.; Cao, P.; Wang, J.; et al. UCTransNet: rethinking the skip connections in u-net from a channel-wise perspective with transformer. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 2441–2449.
18. Chen, J.; Lu, Y.; Yu, Q.; et al. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.04306.
19. Hatamizadeh, A.; Tang, Y.; Nath, V.; et al. UNETR: Transformers for 3D Medical Image Segmentation. In Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022; pp. 1748–1758. https://doi.org/10.1109/wacv51458.2022.00181.
20. Cao, H.; Wang, Y.; Chen, J.; et al. Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation. In *Computer Vision—ECCV 2022 Workshops*; Springer: Cham, Switzerland, 2022; pp. 205–218. https://doi.org/10.1007/978-3-031-25066-8_9.
21. Hatamizadeh, A.; Nath, V.; Tang, Y.; et al. Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*; Springer: Cham, Switzerland, 2022; pp. 272–284. https://doi.org/10.1007/978-3-031-08999-2_22.
22. Chen, J.; Mei, J.; Li, X.; et al. TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers. *Med. Image Anal.* **2024**, *97*, 103280. https://doi.org/10.1016/j.media.2024.103280.
23. Yılmaz, A.; Erkan, U.; Toktaş, A.; et al. AHIR: Deep learning-based autoencoder hashing image retrieval. *Neurocomputing* **2026**, *671*, 132639. https://doi.org/10.1016/j.neucom.2026.132639.
24. Erkan, U.; Yilmaz, A.; Toktas, A.; et al. Object detection-based deep autoencoder hashing image retrieval. *Signal Process. Image Commun.* **2025**, *138*, 117384. https://doi.org/10.1016/j.image.2025.117384.
25. Peng, C.; Myronenko, A.; Hatamizadeh, A.; et al. HyperSegNAS: Bridging One-Shot Neural Architecture Search with 3D Medical Image Segmentation using HyperNet. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 20709–20719. https://doi.org/10.1109/cvpr52688.2022.02008.
26. Huang, Z.; Wang, Z.; Yang, Z.; et al. AdwU-Net: Adaptive Depth and Width U-Net for Medical Image Segmentation by Differentiable Neural Architecture Search. In Proceedings of the International Conference on Medical Imaging with Deep

Learning, Zürich, Switzerland, 6–8 July 2022; pp. 576–589.

27. Ma, L.; Li, N.; Guo, Y.; et al. Learning to Optimize: Reference Vector Reinforcement Learning Adaption to Constrained Many-Objective Optimization of Industrial Copper Burdening System. *IEEE Trans. Cybern.* **2021**, *52*, 12698–12711. https://doi.org/10.1109/tcyb.2021.3086501.
28. Chen, R.; Zhang, M.; Zheng, X.; et al. Thrifty Neural Architecture Search for Medical Image Segmentation (Student Abstract). *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 12925–12926.
29. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 21–26 July 2017; pp. 1800–1807. https://doi.org/10.1109/cvpr.2017.195.
30. Gao, R. Rethinking Dilated Convolution for Real-time Semantic Segmentation. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 17–24 June 2023; pp. 4675–4684. https://doi.org/10.1109/cvprw59228.2023.00493.
31. Stergiou, A.; Poppe, R.W. AdaPool: Exponential Adaptive Pooling for Information-Retaining Downsampling. *IEEE Trans. Image Process.* **2022**, *32*, 251–266. https://doi.org/10.1109/tip.2022.3227503.
32. Paisitkriangkrai, S.; Shen, C.; van den Hengel, A. Pedestrian Detection with Spatially Pooled Features and Structured Ensemble Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1243–1257. https://doi.org/10.1109/tpami.2015.2474388.
33. Gao, S.; Wu, R.; Liu, S.; et al. MSLID-TCN: multi-stage linear-index dilated temporal convolutional network for temporal action segmentation. *Int. J. Mach. Learn. Cybern.* **2025**, *16*, 567–581. https://doi.org/10.1007/s13042-024-02251-y.
34. Kavur, A.E.; Gezer, N.S.; Barış, M.; et al. CHAOS Challenge—combined (CT-MR) healthy abdominal organ segmentation. *Med. Image Anal.* **2021**, *69*, 101950. https://doi.org/10.1016/j.media.2020.101950.
35. Al-Dhabyani, W.; Gomaa, M.; Khaled, H.; et al. Dataset of breast ultrasound images. *Data Brief* **2020**, *28*, 104863. https://doi.org/10.1016/j.dib.2019.104863.
36. Codella, N.; Rotemberg, V.M.; Tschandl, P.; et al. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv* **2019**, arXiv:1902.03368.
37. Liu, L.; Wen, Z.; Liu, S.; et al. MixSearch: Searching for Domain Generalized Medical Image Segmentation Architectures. *arXiv* **2021**, arXiv:2102.13280.
38. Zhou, Q.; Wang, Q.; Bao, Y.; et al. LAEDNet: A Lightweight Attention Encoder–Decoder Network for ultrasound medical image segmentation. *Comput. Electr. Eng.* **2022**, *99*, 107777. https://doi.org/10.1016/j.compeleceng.2022.107777.
39. He, Y.; Yang, D.; Roth, H.; et al. DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021 ; pp. 5837–5846. https://doi.org/10.1109/cvpr46437.2021.00578.
40. Yu, Q.; Yang, D.; Roth, H.; et al. C2FNAS: Coarse-to-Fine Neural Architecture Search for 3D Medical Image Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4125–4134. https://doi.org/10.1109/cvpr42600.2020.00418.