# Compact Training-free NAS with Alternating Evolution Game for Medical Image Segmentation

Xiaoxue Sun[1[0000−0002−3986−6368]], Hongpeng Wang[2,3[0000−0003−4488−1375]], and Pei-Cheng Song[2,4,*[0000−0002−9355−1797]]

[1] College of Computer Science, Nankai University, Tianjin 300350, China
xxsun@mail.nankai.edu.cn
[2] College of Artificial Intelligence, Nankai University, Tianjin 300350, China
hpwang@nankai.edu.cn, spc@mail.nankai.edu.cn
[3] Institute of Intelligence Technology and Robotic Systems, Shenzhen, Research Institute of Nankai University, China
[4] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University

**Abstract.** Neural Architecture Search (NAS) has shown significant potential in designing deep neural networks for medical image segmentation. However, even emerging training-free NAS frameworks often incur substantial computational costs and lengthy search times. To address the critical challenges of computational efficiency and architecture interpretability, the paper proposes a compact training-free NAS framework based on an Alternating Evolution Game (AEG-cTFNAS). The proposed method alternates the search and contribution evaluation of the encoder and decoder within the UNet architecture via alternating games. It employs a truncated normal distribution for compact encoding, sampling, and updating to minimize computational overhead, while Bayesian inference is utilized to estimate the contribution of each block, adaptively adjusting the search strategy and facilitating process visualization. Experimental results on two benchmark datasets reveal that AEG-cTFNAS outperforms both manually designed architectures and NAS-based algorithms, underscoring its efficacy and potential on medical image segmentation. Code is available at https://github.com/spcity/AEG-cTFNAS.

**Keywords:** Alternating game · Training-free Neural Architecture Search · Evolutionary optimization · Medical Image Segmentation

## 1 Introduction

Accurate medical image segmentation is a cornerstone of modern clinical workflows, providing critical quantitative information for diagnosis, treatment planning, and longitudinal disease monitoring [10]. Designing deep neural networks for medical image segmentation necessitates a delicate balance between efficiency and performance, a process traditionally reliant on expert experience [19, 24].

Neural architecture search (NAS) has emerged as a paradigm for automated network design. While NAS can autonomously search for optimal architectures,

traditional methods require validating each candidate, leading to high computational and time costs [5, 21]. Previous studies have focused on designing different encoder or decoder architectures or enhancing segmentation accuracy through special modules [21, 28]. Some studies also explore the impact of backbone network topology and multi-scale fusion capabilities [13, 25]. However, few studies have applied NAS to medical image tasks [20]. The training-free NAS (TFNAS) estimates architecture performance via metrics, avoiding training and reducing verification time [1]. Nonetheless, most methods still employ population-based evolutionary algorithms (EAs), which are computationally demanding. The complex impact of various modules on overall performance is not well integrated due to high interpretability costs [23, 6]. The TFNAS primarily focuses on designing better performance metrics as proxies for model accuracy or greatly utilizing network structure information [8, 26]. The TFNAS is still not agile enough in the clinical environment for pursuing rapid model development [22]. EAs with probability distributions can achieve similar performance with lower costs [12]. Most UNet-based medical image segmentation studies focus on task efficiency, neglecting the impact of modules. Research on module influence in architecture search is limited. Consequently, reducing search costs while analyzing module impacts to enhance performance and visualization remains a critical challenge.

The paper aims to reduce the search cost of EAs and better utilize neural network components to improve efficiency and achieve process visualization. It proposes using probability distributions for sampling and updating, reducing the cost of evaluating numerous candidate architectures. The concept of alternating game theory is introduced to progressively update parts of the network, lowering costs and improving efficiency. Bayesian inference (BI) estimates module impacts without extra evaluations, visualizing module contribution changes. To address the challenge of varying target sizes inherent in medical images, the search space for the encoder and decoder integrates multi-scale feature aggregation. This capability is vital for robustly segmenting diverse structures, from small tumors to entire organs, within a single framework.

Based on the above ideas, the paper proposes an Alternating Evolution Game-based compact Training-Free NAS (AEG-cTFNAS) method for medical image segmentation, with the following key contributions: 1) Propose a truncated normal distribution-based architecture sampling method using evolutionary algorithms, updating the probability distribution with only two evaluations per iteration to reduce the search cost. 2) Propose an alternating game control strategy that regulates the encoder-decoder search cadence based on their contributions to enhance performance. 3) Propose a Bayesian inference approach to estimate the impact of block modifications, facilitating dynamic strategy adjustments and real-time visualization of the search process.

## 2 Methodology

Fig. 1 shows the overall framework of AEG-cTFNAS. It employs an alternating game framework with an evolutionary algorithm, encoding network architectures

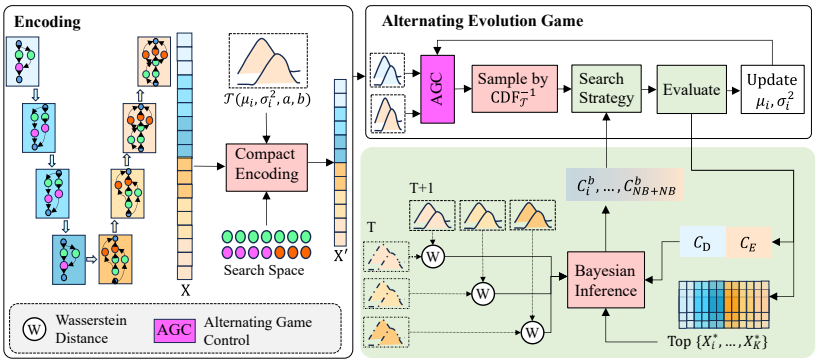


**Fig. 1. Overview of our proposed AEG-cTFNAS.** Compact encoding replaces the original encoding with a truncated normal distribution $\mathcal{T}$ and uses $CDF_{\mathcal{T}}^{-1}$ for sampling. AGC controls the encoder-decoder pace and evaluates contributions. BI estimates block contributions to enhance the search strategy.

via probability distributions. BI then evaluates the impact of different modules, enhancing search effectiveness and enabling visualization.

### 2.1 Alternating Evolution Game

**Compact encoding and sampling** The UNet consists of an encoder and a decoder, each with $N_B$ blocks. The encoder block includes Normal Cell and Reduction Cell, while the decoder block contains Normal Cell and Upsample Cell. In the search space $\mathcal{A}$, the available operations are $S_N$ for Normal Cell, $S_R$ for Reduction Cell, and $S_U$ for Upsample Cell.

The encoder block consists of $N_N$ operations from the Normal Cell space and $N_R$ operations from the Reduction Cell space. The decoder block contains $N_N$ operations from the Normal Cell space and $N_U$ operations from the Upsample Cell space. The genotype is $X^{b'} = [x_1^o, x_1^s, \ldots, x_{N_N}^o, x_{N_N}^s, x_{N_N+1}^o, \ldots, x_{N_N+N_R}^o]$, where $x_i^o$ ranges from 1 to $S_N$ and $x_i^s$ indicates the connection of operation $x_i^o$, with its range from 0 to $i$ ($i < N_N$). $N_R$ operations are performed in parallel. The design for multi-scale feature processing is crucial for capturing both fine-grained local details and broader global context. The first $N_N \times 2$ dimensions and the last $N_R$ dimensions of $X^{b'}$ are each modeled using the truncated normal distribution $x_i^c \sim \mathcal{T}(\mu_i, \sigma_i^2)$. In the first $N_N \times 2$ dimensions, pairs of dimensions are combined as $x_i^c$, while each dimension in the last $N_R$ dimensions is represented by $x_i^c$. For the $i$-th dimension of the compact genotype $x_i^c$, sampling is based on the inverse of the cumulative distribution function $CDF_{\mathcal{T}}^{-1}$, and the equation is

$$x_i^c = \sqrt{2\sigma_i}\,\mathrm{erf}^{-1}\left[-\mathrm{erf}\left(\frac{\mu_i+1}{\sqrt{2\sigma_i}}\right) - y\,\mathrm{erf}\left(\frac{\mu_i-1}{\sqrt{2\sigma_i}}\right)\right] + y\,\mathrm{erf}\left(\frac{\mu_i+1}{\sqrt{2\sigma_i}}\right) + \mu_i, \quad (1)$$

where $y$ is a random number uniformly distributed in the range $[0, 1]$, erf is the error function, $\mathrm{erf}^{-1}$ is the inverse of the error function. The range of values for

$x_i^c$ generated by the Eq. 1 is $[-1, 1]$, which needs to be mapped to the range of $x_i$. After mapping, for $x_i$ ($i < N_N \times 2$), the value range is $[0, S_N \times i]$, for $x_i$ ($i \geq N_N \times 2$), the value range is $[0, S_R]$. Thus, the encoding for a block is $X^b = [x_1, x_2, \ldots, x_{N_N}, x_{N_N+1}, \ldots, x_{N_N+N_R}]$. Finally, the compact encoding for the entire network architecture is obtained by concatenating the encodings of all blocks in the encoder and decoder, represented as $X$.

**Alternating Game Control** In Fig. 1, AGC controls the alternating updates of the decoder and encoder. The network is encoded as a compact genotype $X = [X^E, X^D] = [X_1^{b,E}, \ldots, X_{N_B}^{b,E}, X_{N_B+1}^{b,D}, \ldots, X_{N_B+N_B}^{b,D}]$, where $X_1^{b,E}$, the first encoder block, has the same shape as $X^b$, and $X_{N_B+N_B}^{b,D}$ denotes the $N_B$-th decoder block. The alternating interval $M$ is set, and the initial compact genotype is $X$. If the encoder is searched and updated first, the decoder $X^D$ is not sampled, and $X$ is updated to $X^{+E}$. Once the iteration count exceeds $M$, the search and update switch to the decoder, with $X^E$ sampling stopping and $X$ being updated to $X^{+D}$. To assess the contributions, a joint update $X^{+E+D}$ is generated. The individual contributions of the encoder and decoder updates are

$$\begin{aligned} C_E &= \frac{\text{sign}[f(X)-f(X^{+E+D})]\times[f(X)-f(X^{+E})]}{f(X)-f(X^{+E+D})}, \\ C_D &= \frac{\text{sign}[f(X)-f(X^{+E+D})]\times[f(X)-f(X^{+D})]}{f(X)-f(X^{+E+D})}. \end{aligned} \tag{2}$$

Let $f(\cdot)$ be the objective function for the minimization problem, which can be either single-objective or multi-objective. To allow for more flexible alternating control, if $C_E$ or $C_D$ contributes positively for two consecutive iterations, $M$ will increase by 1, up to a maximum of $1.5M$; if negative contributions occur consecutively, $M$ will decrease by 1, down to a minimum of $0.5M$.

### 2.2 Contribution inference for search strategy

**Bayesian Inference** The paper examines the encoder and decoder's influence on overall optimization and provides a detailed analysis of block-level contribution changes. Given that traditional methods like Shapley value calculations are inefficient for many blocks, the paper leverages BI to estimate each block's contribution by combining encoder and decoder effects.

As shown in Fig. 1, for the $i$-th block in the encoder, the truncated normal distributions of the entire block are treated as an independent joint distribution. The Wasserstein distance $d_i$ between the probability distributions at iterations $T$ and $T+1$ can be calculated. Then, a BI dataset $Data_E = [d^{T-W+1}, ..., d^T, d^{T+1}]$ is constructed, where $W$ controls the amount of data retained, and $d^{T+1} = [d_1, ..., d_i, ..., d_{N_B}, C_E^{T+1}]$. Similarly, the dataset for the decoder is $Data_D = [p^{T-W+1}, ..., p^T, p^{T+1}]$, and $p^{T+1} = [p_1, ..., p_i, ..., p_{N_B}, C_D^{T+1}]$. Based on the constructed data, a logistic regression model can be used to establish the relationship between the first $N_B$ dimensions of input and the last dimension of the response. The model maps the input to the response probability using $\log p(\mathbf{y}|X, \boldsymbol{\beta}) = \sum_{i=1}^{W} [y_i \log \sigma(z_i) + (1 - y_i) \log(1 - \sigma(z_i))]$, where $z_i$

represents the linear part of the logistic regression model, and $\boldsymbol{\beta}$ denotes the set of coefficients for the input dimensions in $z_i$.

A normal prior is imposed on $\boldsymbol{\beta}$, and Bayes theorem combines likelihood and before forming the posterior. Since the posterior is often intractable, and assuming it is Gaussian, the paper applies the Laplace approximation at the maximum posterior (MAP) point $\hat{\boldsymbol{\beta}}$, approximating the posterior as a Gaussian: $p(\boldsymbol{\beta}|Data_E) \approx \mathcal{N}\left(\hat{\boldsymbol{\beta}}, \mathbf{H}^{-1}\right)$. This method requires solving the MAP estimate once, which is efficient when the posterior is approximately Gaussian. Ultimately, the contribution of each block in the encoder can be obtained as $C_E^b = [C_{E,1}^b, ..., C_{E,i}^b, ..., C_{E,N_B}^b]$. Similarly, for decoder, $C_D^b = [C_{D,1}^b, ..., C_{D,i}^b, ..., C_{D,N_B}^b]$. Finally, the contributions of all blocks are represented as $C^b = [C_E^b, C_D^b]$.

**Contribution inference-based search** To better leverage the superior architectures during the search process, the paper constructs a historical optimal list $Best_{pool} = [X_1^*, ..., X_K^*, X_{eq}^*]$ under the compact encoding, which includes $K$ historical optimal encodings and an average encoding $X_{eq}^* = \frac{1}{K}\sum_i^K X_i^*$. Based on the inferred contributions, the following search strategy is constructed.

$$X^{t+1} = X_r^{Best} + V_{cr} C_v^b (X^t - X_\mu^t) + C_{ED}(X_r^{Best} - X^t) \tag{3}$$

$X_r^{Best}$ is a historical optimal architecture randomly selected from $Best_{pool}$ with equal probability, and $V_{cr}$ is a 0-1 vector based on the crossover probability $cr$, matching the shape of $X_r^{Best}$. The $C^b$ is extended to the corresponding compact encoded vector $C_v^b$. $X_\mu^t$ is the compact architecture encoding obtained by mapping the mean $\mu$, and $C_{ED}$ extends the contributions of the encoder and decoder ($C_E$ and $C_D$) to the compact encoded vectors. $X^t$ is generated by Eq. 1.

The objective function compares $f(X^{t+1})$ and $f(X_1^*)$ to select a *winner* and a *loser*, which update each dimension of $\mathcal{T}$ as: $\mu^{t+1} = \mu^t + \frac{\text{winner}-\text{loser}}{Np}$, $\sigma^{t+1} = \sqrt{(\sigma^t)^2 + (\mu^t)^2 - (\mu^{t+1})^2 + \frac{\text{winner}^2-\text{loser}^2}{Np}}$, where $Np$ is the size of the virtual population. Unlike traditional evolutionary algorithms with a population size $N_P$ require $N_G \times N_P$ evaluations over $N_G$ iterations, AEG-cTFNAS performs only two evaluations per iteration (i.e., $f(X^{t+1})$ and either $f(X^{+E})$ or $f(X^{+D})$), totaling $2N_G$ evaluations.

## 3 Experiments and Results

### 3.1 Experimental Settings and Details

The performance of AEG-cTFNAS was evaluated on two medical image segmentation benchmarks: the Automatic Cardiac Diagnosis Challenge (ACDC) dataset, from the MICCAI 2017 challenge, the partition follows TransUNet [3, 4], and the Endovis 2018 dataset from the MICCAI Robot Scene Segmentation Challenge, consisting of 15 training and 4 inference videos [7]. Experiments were conducted using Pytorch 1.13.0 on a single NVIDIA RTX 4090 GPU with

24GB of memory for architecture searching, training, and testing. AEG-cTFNAS searched for 50 iterations on each dataset, with the block ($N_B$) being 4. For ACDC, the input size was $224 \times 224$, trained with the AdamW optimizer (learning rate $1e-4$) for up to 400 epochs (batch size 12). For Endovis 2018, training was done for 200 epochs (batch size 4) at a resolution of $640 \times 640$ and learning rate $1e-4$. FLOPs, $jacob\_cov$, and $synflow$ were used as the search optimization functions [20]. The loss function combines DICE and cross-entropy loss.

### 3.2 Comparison with State-of-the-art Methods

**Evaluation on ACDC** The section compares 10 manually designed networks (MN) and 3 NAS-based automatically designed networks (AN), with the results summarized in Table 1. The organs include the left ventricle (LV), right ventricle (RV), and myocardium (Myo). Our proposed AEG-CTFNAS achieves a new state-of-the-art (SOTA) mean Dice score of 91.72% on the ACDC dataset, with only $3.34M$ parameters and $7.79G$ FLOPs. This represents a 0.43% improvement over the previous SOTA EMCAD (91.29% Dice), especially on LV and Myo. Among NAS-based methods, AEG-cTFNAS outperforms NASUNet, MedNAS(a) (using MOEAD), and MedNAS(b) (using NSGA-II) by about 9.16%, 1.82%, 2.74%. The AEG-cTFNAS requires fewer search iterations than MedNAS. So, AEG-cTFNAS surpasses MNs and outperforms other NAS-based methods with a lower search cost, achieving SOTA performance. In terms of runtime, NASUNet takes 18441 seconds. With training-free metrics, MedNAS (using NSGA-II) finishes in 3183.96 seconds, while AEG-cTFNAS requires only 369.76 seconds.

**Table 1.** Results of cardiac organ segmentation on ACDC dataset. The ↑ / ↓ denotes the higher (lower) the better. The best scores are shown in **bold**.

| Architectures | | FLOPs (G) | Params (M) | DICE(%)↑ | | | |
|---|---|---|---|---|---|---|---|
| | | | | LV | RV | Myo | Ave |
| MN | UNet [16] | 1.96 | 1.08 | 93.16 | 82.36 | 81.73 | 85.75 |
| | R50 UNet [15] | 4.06 | 25.55 | 94.92 | 87.10 | 80.63 | 87.55 |
| | DenseNet [9] | 116.19 | 7.55 | 95.99 | 87.69 | 89.59 | 91.09 |
| | Att-UNet [11] | 50.97 | 34.88 | 93.47 | 87.58 | 79.20 | 86.75 |
| | TransUNet [3] | 19.70 | 90.44 | 95.18 | 86.67 | 87.27 | 91.05 |
| | TransCASCADE [14] | - | 123.48 | 95.48 | 89.30 | 89.07 | 91.28 |
| | SwinUNet [2] | 8.71 | 41.34 | 94.85 | 88.60 | 87.28 | 90.24 |
| | MT-UNet [18] | 44.79 | 75.07 | 95.62 | 86.64 | 89.04 | 90.43 |
| | EMCAD [15] | 4.29 | 26.77 | 95.22 | **89.49** | 89.15 | 91.29 |
| | VM-UNet [17] | 3.54 | 15.48 | 95.65 | 89.01 | 88.80 | 91.15 |
| AN | NASUNet [21] | **0.82** | **0.16** | 90.92 | 77.40 | 79.37 | 82.56 |
| | MedNAS(a) [20] | 5.18 | 1.77 | 95.05 | 86.63 | 88.01 | 89.90 |
| | MedNAS(b) [20] | 5.48 | 2.11 | 94.83 | 85.26 | 86.85 | 88.98 |
| | AEG-cTFNAS(**Ours**) | 7.79 | 3.34 | **96.04** | 89.33 | **89.80** | **91.72** |

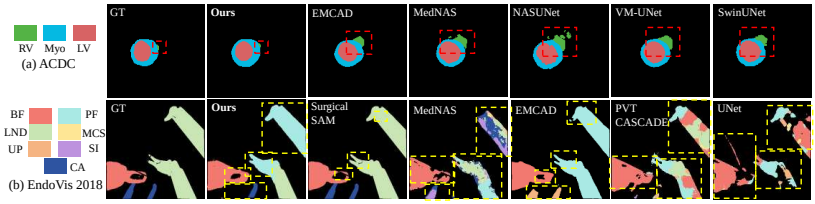


**Fig. 2.** Qualitative results of segmentation on ACDC and Endovis 2018.

**Evaluation on Endovis 2018** In medical image segmentation, the ACDC dataset contains only three tissues in balanced proportions, while the EndoVis 2018 dataset includes multiple similar instruments in a complex background that can lead to confusion and imbalance. We compare 9 MNs and 2 ANs on the EndoVis 2018 dataset. In Table 2, AEG-cTFNAS demonstrates excellent IoU performance on some MNs, outperforming other NAS-based methods. However, compared with the networks designed specifically for surgical tool segmentation (ISINet and SurgicalSAM), the effect is inferior, not reaching SOTA. In Fig. 2, AEG-cTFNAS excels in capturing fine structures on pure tissue segmentation but is less effective in distinguishing categories on surgical tool segmentation with class-imbalance. In contrast, SurgicalSAM achieves accurate class prompts using a prototype-based class prompt encoder and contrastive prototype learning.

**Table 2.** Comparison results for surgical tool segmentation on the Endovis 2018 dataset. The best scores are shown in **bold**. The metrics include intersection over union (IoU) and mean class IoU (mc IoU) for Bipolar Forceps (BF), Prograsp Forceps (PF), Large Needle Driver (LND), Monopolar Curved Scissors (MCS), Ultrasound Probe (UP), Suction Instrument (SI), and Clip Applier (CA).

| Architectures | | IoU↑ (%) | Instrument Categories | | | | | | | mc IoU↑ (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | BF | PF | LND | MCS | UP | SI | CA | |
| MN | UNet [16] | 26.81 | 55.96 | 5.51 | 1.01 | 56.44 | 0.07 | 0.00 | 0.00 | 17.00 |
| | DenseNet [9] | 28.44 | 57.06 | 4.92 | 0.02 | 62.50 | 0.26 | 5.41 | 0.00 | 18.60 |
| | PVTCASCADE [14] | 29.86 | 59.71 | 5.91 | 4.18 | 65.17 | 0.52 | 5.96 | 0.00 | 19.48 |
| | EMCAD [15] | 31.95 | 72.26 | 9.00 | 0.74 | 67.81 | 0.00 | 7.41 | 0.14 | 22.48 |
| | ISINet [7] | 70.94 | 73.83 | 48.61 | 30.98 | 88.16 | 2.16 | 37.68 | 0.00 | 40.21 |
| | SurgicalSAM [27] | **80.33** | **83.66** | **65.63** | **58.75** | **88.56** | **21.23** | **54.48** | **39.78** | **58.87** |
| AN | NASUNet [21] | 29.90 | 60.05 | 4.63 | 3.51 | 62.74 | 0.09 | 7.38 | 0.20 | 19.80 |
| | MedNAS [20] | 30.64 | 65.38 | 5.70 | 3.75 | 3.67 | 0.18 | 8.99 | 0.18 | 21.12 |
| | **Ours** | 32.70 | 71.87 | 8.60 | 0.89 | 67.21 | 0.37 | 7.76 | 0.91 | 22.52 |

### 3.3 Ablation Study

The section presents the ablation study of AEG-cTFNAS on the ACDC dataset (Table 3), evaluating the contribution of each module, where ASD denotes Av-

erage surface distance. Baseline 1 uses a standard evolutionary algorithm for TFNAS-based segmentation. Baseline 2 replaces the population with a probability distribution. The score is poor, but it reduces the search time. Baseline 3 further integrates the AGC module for an alternating evolutionary game, while Baseline 4 incorporates BI into compact encoding. Results indicate that the AGC module improves segmentation performance, with BI delivering similar benefits. Every component demonstrates its capabilities in AEG-cTFNAS. Overall, AEG-cTFNAS achieves the best segmentation performance.

**Table 3.** Ablation study based on different components on ACDC dataset.

| Methods | Components | | | mIoU ↑ | ASD ↓ | DICE(%) ↑ | | | | HD95 |
|---|---|---|---|---|---|---|---|---|---|---|
| | Compact | AGC | BI | (%) | (mm) | LV | RV | Myo | Ave | (mm) |
| Baseline 1 | No | No | No | 82.35 | 3.83 | 95.05 | 86.63 | 88.01 | 89.90 | 1.54 |
| Baseline 2 | Yes | No | No | 74.29 | 5.16 | 91.90 | 78.13 | 81.27 | 83.77 | 2.05 |
| Baseline 3 | Yes | Yes | No | 77.74 | 4.27 | 92.97 | 84.74 | 82.98 | 86.89 | 1.79 |
| Baseline 4 | Yes | No | Yes | 79.94 | 5.37 | 93.71 | 85.52 | 85.51 | 88.25 | 2.2 |
| AEG-cTFNAS | Yes | Yes | Yes | 85.16 | 3.50 | **96.04** | **89.33** | **89.80** | **91.72** | **1.52** |

### 3.4 Visualization of the Search Process

Fig. 3 visualizes the search process, illustrating the actual contributions of the encoder and decoder and the variations in the estimated contributions of different blocks. The estimated block contributions in the decoder align with the

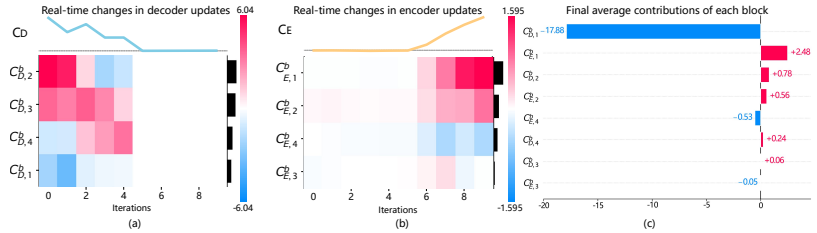


**Fig. 3.** Visualization of dynamic module contributions during search ($M = 5$).

actual evaluations, indicating that block updates negatively affect the decoder and reduce its overall contribution. In contrast, after the 5th iteration, encoder blocks contribute positively, leading to a positive overall contribution.

## 4 Conclusion

The paper proposes AEG-cTFNAS, a novel TFNAS for efficiently generating network architectures tailored to medical image segmentation, addressing the criti-

cal barriers of high search cost and low transparency that hinder the widespread adoption in clinical practice. For the first time, a truncated normal distribution is used to encode neural architectures, replacing population-based evolutionary algorithms to reduce search costs. The alternating evolutionary game framework is proposed to adjust the search process based on the contributions of different modules. BI-based module contribution estimation is crucial for precise segmentation, incurs no extra cost, and enables real-time visualization of the search. Experiment results demonstrate that AEG-cTFNAS achieves SOTA performance on the ACDC dataset, although further improvements are needed for challenging surgical tool segmentation tasks. Future work will extend AEG-cTFNAS by incorporating inter-class contrastive learning to enhance generalizability across diverse datasets.

**Acknowledgments.** This work was supported by the National Key R&D Program of China under Grant 2022YFB4702800; the Science and Technology Program of Tianjin under Grants 22KPXMRC00040 and 22JCYBJC01240; the Sustainable Development Science and Technology Special Project of Shenzhen under Grant KCXFZ20230731100900002; and the Guangdong Basic and Applied Basic Research Foundation under Grant 2414050001625.

**Disclosure of Interests.** The authors have no competing interests to declare.

## References

1. Bhardwaj, K., Li, G., Marculescu, R.: How does topology influence gradient propagation and model performance of deep networks with densenet-type skip connections? In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13493–13502 (2021)
2. Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-unet: Unet-like pure transformer for medical image segmentation. In: European Conference on Computer Vision (ECCV). pp. 205–218. Springer (2022)
3. Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L., Zhou, Y.: Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306 (2021)
4. Chen, J., Mei, J., Li, X., Lu, Y., Yu, Q., Wei, Q., Luo, X., Xie, Y., Adeli, E., Wang, Y., et al.: TransUNet: Rethinking the U-Net architecture design for medical image segmentation through the lens of transformers. Medical Image Analysis **97**, 103280 (2024)
5. Elsken, T., Metzen, J.H., Hutter, F.: Neural Architecture Search: A Survey. Journal of Machine Learning Research **20**(55), 1–21 (2019)
6. Gao, S., Zhou, H., Gao, Y., Zhuang, X.: BayeSeg: Bayesian Modeling for Medical Image Segmentation with Interpretable Generalizability. Medical image analysis **89**, 102889 (2023)
7. González, C., Bravo-Sánchez, L., Arbelaez, P.: Isinet: an instance-based approach for surgical instrument segmentation. In: International conference on medical image computing and computer-assisted intervention (MICCAI). pp. 595–605. Springer (2020)
8. Li, G., Hoang, D., Bhardwaj, K., Lin, M., Wang, Z., Marculescu, R.: Zero-Shot Neural Architecture Search: Challenges, Solutions, and Opportunities. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(12), 7618–7635 (2024)

9. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.W., Heng, P.A.: H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes. IEEE Transactions on Medical Imaging **37**(12), 2663–2674 (2018)
10. Ma, J., He, Y., Li, F., Han, L., You, C., Wang, B.: Segment anything in medical images. Nature Communications **15**(1), 654 (2024)
11. Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al.: Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999 (2018)
12. Pan, J.S., Song, P.C., Chou, J.H., Watada, J., Chu, S.C.: FPGA-Based Compact Differential Evolution for General-Purpose Optimization in Resource-Constrained Devices. IEEE Transactions on Industrial Informatics **20**(4), 6781–6790 (2024)
13. Qin, S., Zhang, Z., Jiang, Y., Cui, S., Cheng, S., Li, Z.: NG-NAS: Node growth neural architecture search for 3D medical image segmentation. Computerized Medical Imaging and Graphics **108**, 102268 (2023)
14. Rahman, M.M., Marculescu, R.: Medical image segmentation via cascaded attention decoding. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 6222–6231 (2023)
15. Rahman, M.M., Munir, M., Marculescu, R.: Emcad: Efficient multi-scale convolutional attention decoding for medical image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11769–11779 (2024)
16. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 234–241. Springer (2015)
17. Ruan, J., Li, J., Xiang, S.: Vm-unet: Vision mamba unet for medical image segmentation. arXiv preprint arXiv:2402.02491 (2024)
18. Wang, H., Xie, S., Lin, L., Iwamoto, Y., Han, X.H., Chen, Y.W., Tong, R.: Mixed transformer u-net for medical image segmentation. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 2390–2394. IEEE (2022)
19. Wang, Y., Wang, Z., Feng, Y., Zhang, L.: WDCCNet: Weighted Double-Classifier Constraint Neural Network for Mammographic Image Classification. IEEE Transactions on Medical Imaging **41**(3), 559–570 (2022)
20. Wang, Y., Zhen, L., Zhang, J., Li, M., Zhang, L., Wang, Z., Feng, Y., Xue, Y., Wang, X., Chen, Z., Luo, T., Goh, R.S.M., Liu, Y.: MedNAS: Multiscale Training-Free Neural Architecture Search for Medical Image Analysis. IEEE Transactions on Evolutionary Computation **28**(3), 668–681 (2024)
21. Weng, Y., Zhou, T., Li, Y., Qiu, X.: NAS-Unet: Neural Architecture Search for Medical Image Segmentation. IEEE Access **7**, 44247–44257 (2019)
22. Wu, M.T., Lin, H.I., Tsai, C.W.: A Training-Free Neural Architecture Search Algorithm Based on Search Economics. IEEE Transactions on Evolutionary Computation **28**(2), 445–459 (2024)
23. Xiao, H., Wang, Z., Zhu, Z., Zhou, J., Lu, J.: Shapley-NAS: Discovering Operation Contribution for Neural Architecture Search. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 11882–11891 (2022)
24. Xing, Z., Ye, T., Yang, Y., Liu, G., Zhu, L.: SegMamba: Long-range Sequential Modeling Mamba For 3D Medical Image Segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2024)

25. Yan, X., Jiang, W., Shi, Y., Zhuo, C.: MS-NAS: Multi-Scale Neural Architecture Search for Medical Image Segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). pp. 388–397 (2020)
26. Yang, J., Liu, Y., Wang, W., Wu, H., Chen, Z., Ma, X.: PATNAS: A Path-Based Training-Free Neural Architecture Search. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(3), 1484–1500 (2025)
27. Yue, W., Zhang, J., Hu, K., Xia, Y., Luo, J., Wang, Z.: Surgicalsam: Efficient class promptable surgical instrument segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI). vol. 38, pp. 6890–6898 (2024)
28. Zhou, J., Chen, J.: SIDN-NAS: Scalable Iterative Dense Network with Neural Architecture Search Optimization for Medical Image Segmentation. In: 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 2297–2304 (2024)