Medical Image Analysis (2021)

Contents lists available at ScienceDirect

Medical Image Analysis

journal homepage: www.elsevier.com/locate/media

Towards Bi-directional Skip Connections in Encoder-Decoder Architectures and Beyond

Tiange Xiang[a], Chaoyi Zhang[a], Xinyi Wang[a], Yang Song[b], Dongnan Liu[a], Heng Huang[c,d], Weidong Cai[a,*]

[a] *School of Computer Science, University of Sydney, Australia*
[b] *School of Computer Science and Engineering, University of New South Wales, Australia*
[c] *Electrical and Computer Engineering, University of Pittsburg, USA*
[d] *JD Finance America Corporation, Mountain View, CA, USA*

ARTICLE INFO

Article history:
Received ***
Received in final form ***
Accepted ***
Available online ***

Communicated by ***

Keywords: Semantic segmentation, Bi-direction connections, Recursive networks, Multi-scale, Neural architecture search

ABSTRACT

U-Net, as an encoder-decoder architecture with forward skip connections, has achieved promising results in various medical image analysis tasks. Many recent approaches have also extended U-Net with more complex building blocks, which typically increase the number of network parameters considerably. Such complexity makes the inference stage highly inefficient for clinical applications. Towards an effective yet economic segmentation network design, in this work, we propose backward skip connections that bring decoded features back to the encoder. Our design can be jointly adopted with forward skip connections in any encoder-decoder architecture forming a recurrence structure without introducing extra parameters. With the backward skip connections, we propose a U-Net based network family, namely Bi-directional O-shape networks, which set new benchmarks on multiple public medical imaging segmentation datasets. On the other hand, with the most plain architecture (BiO-Net), network computations inevitably increase along with the pre-set recurrence time. We have thus studied the deficiency bottleneck of such recurrent design and propose a novel two-phase Neural Architecture Search (NAS) algorithm, namely BiX-NAS, to search for the best multi-scale bi-directional skip connections. The ineffective skip connections are then discarded to reduce computational costs and speed up network inference. The finally searched BiX-Net yields the least network complexity and outperforms other state-of-the-art counterparts by large margins. We evaluate our methods on both 2D and 3D segmentation tasks in a total of six datasets. Extensive ablation studies have also been conducted to provide a comprehensive analysis for our proposed methods.

© 2021 Elsevier B. V. All rights reserved.

1. Introduction

Accurate and efficient analysis of medical images is of great interest to the computer vision and medical communities. The diagnosis of potential disease from medical images relies on a wide range of features implied by visual clues. Such decision making process requires time-consuming efforts from physicians, slowing down the process of diagnosis. Therefore, effective and efficient computer-aided models that provide automated analysis of medical images are highly desirable.

As one of the essential medical image analysis, semantic image segmentation requires dense predictions to indicate the class of each pixel: part of a nucleus, a certain organ, or an anomaly region. Benefiting from forward feature skips, U-Net has demonstrated its wide success in segmenting images

*Corresponding author.
e-mail: tom.cai@sydney.edu.au (Weidong Cai)

of all kinds of modalities. However, the level-to-level forward connections are limited in feature aggregation ability and better encoder-decoder aggregation strategies are needed for more advanced feature refinement. In this work, we study the skip mechanism in encoder-decoder architectures and design effective yet efficient segmentation networks.

1.1. Related Work

Image segmentation. Computer-aided image segmentation has a long history (Haralick and Shapiro, 1985) in the computer vision field. With the recent success of deep learning, neural network based methods have demonstrated their ability in fast and accurate segmentation of digital images.

As a pioneer segmentation model, U-Net (Ronneberger et al., 2015) adopts an encoder-decoder structure that first encodes the visual signals provided in an image into high-dimensional features and then decodes the abstract semantics to learn the pixel-to-pixel mapping between the input image and the segmentation mask. In contrast to conventional autoencoders (Hinton and Zemel, 1994), skip connections are added between the encoder and decoder, so that encoded features are forwarded to the decoder. Such forward skip connections enable fine-grained aggregations of features and preserve better gradient flows during network optimization. Although other advanced approaches exist in different forms (Chen et al., 2017; Long et al., 2015), U-Net often demonstrates its superior performances and serves as the most important backbone in medical image segmentation tasks.

U-Net variants. Recent studies have developed different building blocks to extend U-Net. For instance, V-Net (Milletari et al., 2016) applies U-Net to process 3D voxel data for 3D vision tasks. W-Net (Xia and Kulis, 2017) concatenates two U-Nets head-to-tail to approach image segmentation tasks in an unsupervised style. M-Net (Mehta and Sivaswamy, 2017) studies the impact of using multi-scale features through paired downsampling and upsampling layers. U-Net++ (Zhou et al., 2018), as a direct extension of U-Net, designs nested building blocks with dense skip connections to better propagate encoded features. Apart from the modifications on network structure, attention U-Net (Oktay et al., 2018) incorporates attention mechanism into U-Net by learning self-attentions to be applied on the skipped features. However, the above U-Net variants require additional functional modules, leading to an escalation of network complexity. In our work, we improve the performance of U-Net via inserting novel backward feature skip connections, where building blocks are reused without introducing any extra network parameters.

Recurrent convolutional networks. Iterative refinement of features has been proved effective in many computer vision tasks (Han et al., 2018; Guo et al., 2019; Wang et al., 2019; Alom et al., 2018). Guo et al. (2019) reuses ResNet residual blocks (He et al., 2016) to fully utilize the limited parameters. With the similar recurrent strategy, Wang et al. (2019) proposed R-U-Net, which connects multiple U-Net architectures head-to-tail with shared parameters to enhance segmentation performances. R2U-Net (Alom et al., 2018) adopts a similar approach that only recurses the last building block at each level of refinement. By contrast, our method learns recurrent bi-directional connections between encoders and decoders.

Neural architecture search. Compared to manually crafted networks, Neural Architecture Search (NAS) algorithms automatically search for the optimal architecture in a defined search space. Reinforcement Learning (RL) based methods (Zoph and Le, 2016) utilize a stand-alone agent to monitor the search process under certain proxies. Evolution based methods (Real et al., 2017, 2019) start with a set of randomly sampled ancestor networks and then progressively evolve the population to new generations with more powerful offspring networks. Differentiable methods (Liu et al., 2019b; Guo et al., 2020) optimize "architecture parameters" through back propagation by relaxing the discrete search space, and define the network topology based on the optimized architecture parameters.

NAS searched architectures also obtained success in segmentation tasks. Auto-DeepLab (Liu et al., 2019a) designed a differentiable search space to determine the best operators and topologies for each building block. Similarly, NAS-Unet (Weng et al., 2019) also adopted a gradient-based search that automatically discovers basic cell structures to construct a U-Net like architecture. As a multi-scale counterpart, the search space in (Yan et al., 2020) covers multiple encoding and decoding levels. Their proposed MS-NAS has the ability of aggregating multi-level features, which leads to better segmentation performance. However, the search process of the above methods is time consuming and computational inefficient. In this work, we design an efficient two-phase NAS method to find a subset of optimal bi-directional skip connections that yield the least network parameters and computations.

1.2. Contributions

The preliminary versions of this work were published as conference papers in MICCAI 2020 (Xiang et al., 2020) and MICCAI2021 (Wang et al., 2021). In this work, our overall contributions include: **(1)** We propose bi-directional skip connections in encoder-decoder architectures for an iterative aggregation of encoded and decoded features. **(2)** We incorporate bi-directional skip connections into a simple U-Net architecture, namely BiO-Net, achieving better segmentation performance without using extra parameters. **(3)** The deficiency of BiO-Net is analyzed and an upgraded network, BiO-Net++, is designed with significantly less network complexity and multi-scale skip connections. **(4)** To further reduce the redundancies in BiO-Net++, we introduce an efficient two-phase BiX-NAS method to automatically search for resource-aware and optimal skip connections from the BiO-Net++. The finally searched BiX-Net model achieves on par performance to the BiO-Net but with significantly less complexity. **(5)** Our proposed bi-directional skip connections along with the novel networks were evaluated on a variety of medical image segmentation tasks including: 2D nuclei segmentation, 2D multi-organ segmentation, 3D Covid-19 infection segmentation, and two 3D segmentation tasks from the Medical Segmentation Decathlon (MSD). Our methods establish new benchmarks on these datasets. **(6)** Extensive ablation studies were conducted to fully study the usefulness of

each of the proposed components. Our project page with source codes has been made publicly available to foster any future research [1].

1.3. Symbols and Notations

For notion simplicity, here we define several symbols and notations before presenting our methods: **T**: The pre-set recurrence time. Note that $\mathbf{T}-1$ represents the total time of backward feature skipping in our bi-directional networks; **N**: The channel expansion multiplier to all network layers; **W**: The number of backward skip connections used from the deepest encoding level; **L**: The greatest encoding depth; **P**: The population of candidate networks of a generation; **Extraction stage**: A sequential encoding or decoding process. There are two extraction stages (i.e., one encoder and one decoder) in U-Net; **Searching block**: Any building block in an extraction stage; **C**: The ready-to-be-searched candidate skips between a pair of extraction stages; **E**: The evolved skips from a set of candidate skips between a pair of extraction stages.

2. Bi-directional Skip Connections

Let's first consider a simple encoder-decoder architecture similar to U-Net. Skip connections are built at different levels, facilitating the information exchange between encoder and decoder.

Forward Skip Connections. Forward skip connections carry forward the encoded low-level features to the decoder. There are two incoming feature streams to each decoder block: the sequential feature stream from a lower level decoder block $\hat{\mathbf{x}}_{in}$ and the forward skipped feature stream $\mathbf{f}_{enc}$ from the corresponding encoder block at the same level. The decoder block then fuses the two streams through a FUSE operation, and the features are subsequently propagated through the decoding convolutions DEC to generate semantic features $\mathbf{f}_{dec}$. This process can be defined as:

$$\mathbf{f}_{dec} = \texttt{DEC}(\texttt{FUSE}(\mathbf{f}_{enc},\ \hat{\mathbf{x}}_{in})). \tag{1}$$

Backward Skip Connections. Paired to forward skip connections, we define the backward skip connections that pass the decoded high-level semantic features $\mathbf{f}_{dec}$ back to the encoder. An encoder block now combines $\mathbf{f}_{dec}$ with its original sequential input $\mathbf{x}_{in}$ from an upper level encoder block to create additional aggregations between low-level and high-level features. Similar to 1, our encoding process can be formulated with its encoding convolutions ENC as:

$$\mathbf{f}_{enc} = \texttt{ENC}(\texttt{FUSE}(\mathbf{f}_{dec},\ \mathbf{x}_{in})). \tag{2}$$

[1] https://bionets.github.io/

2.1. Recurrent Inference

Unlike forward skips, our proposed backward skip connections cannot work directly in the classic one-pass encoder-decoder architecture, since encoder layers will only be forwarded once. To enable backward feature skipping, we design an "O"-shape recurrent inference routine for bi-directional networks (Sec. 3) that propagate features iteratively between encoder and decoder through the bi-directional skip connections. Noticeably, such recurrent inference reuses existing network weights, and no extra parameters are introduced during the inference. Finally, the building block outputs using our bi-directional skip connections can be formulated as follows, at the iteration i:

$$\begin{aligned}\mathbf{x}^i_{out} &= \texttt{DOWN}(\texttt{ENC}(\texttt{FUSE}(\texttt{DEC}(\texttt{FUSE}(\mathbf{f}^{i-1}_{enc}, \hat{\mathbf{x}}^{i-1}_{in})), \mathbf{x}^i_{in}))),\\ \hat{\mathbf{x}}^i_{out} &= \texttt{UP}(\texttt{DEC}(\texttt{FUSE}(\texttt{ENC}(\texttt{FUSE}(\mathbf{f}^i_{dec},\ \mathbf{x}^i_{in})), \hat{\mathbf{x}}^i_{in}))),\end{aligned} \tag{3}$$

where DOWN represents the downsampling process, and UP represents the upsampling process.

3. Recurrent Bi-directional Networks

3.1. BiO-Net: A Bi-directional U-Net

Our bi-directional skip connections are believed to be universally applicable. In this study, we insert them into a basic encoder-decoder architecture, namely BiO-Net, for method developments and evaluations. BiO-Net involves only the plain convolutional layers, batch normalization (Ioffe and Szegedy, 2015) layers, and ReLU (Nair and Hinton, 2010) activation layers. Note that no batch normalization layer is reused through network recursion. To align with U-Net, our BiO-Net adopts max pooling, convolution transpose and concatenation as DOWN, UP and FUSE, respectively. An overview of our BiO-Net is shown in Fig. 1.

Giving an input image, we first apply three convolution-normalization-activation combinations to extract low-level features. There is no skip connection attached to such pre-processing blocks and the parameters will not be reused. The initially extracted features are then sent to a cascade of encoder blocks that are recursed through the bi-directional skip connections. Note that no features are backward skipped at the first recurrence. To make the feature map size consistent along different iterations, we duplicate the encoded features at the first encoding iteration.

After the encoding stage, a bridge block with additional convolutions is employed to transfer the encoded features. Subsequently, a series of decoder blocks consume the features and recover encoded details using convolution transpose. During the decoding stage, our backward skip connections preserve retrieved features by concatenating them with the ones skipped from the same level encoder block. The recursion begins at the end of the last decoder block. After recursing for **T** iterations, the decoded output will be fed into the post-processing block constructed similarly to the pre-processing block. The post-processing blocks will not be involved in the recurrence.

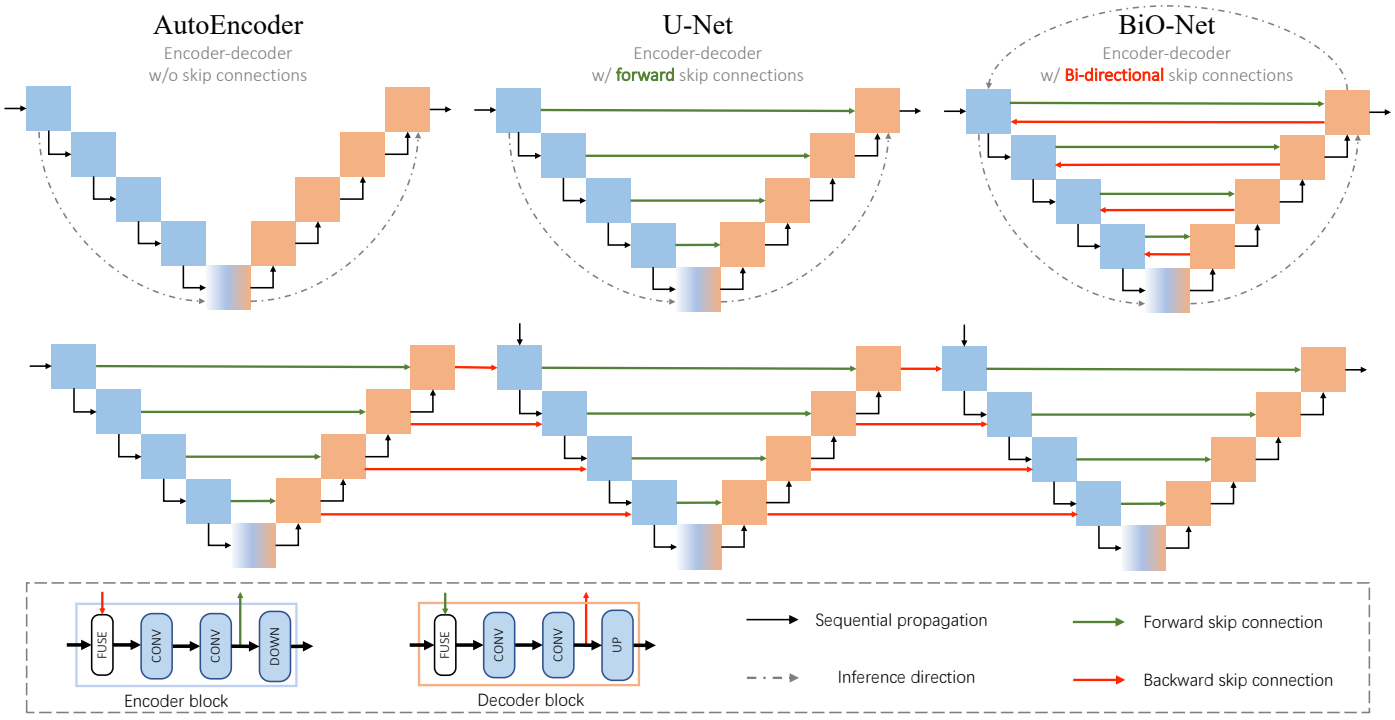

Fig. 1: Top: AutoEncoder (Hinton and Zemel, 1994), U-Net (Ronneberger et al., 2015), and our BiO-Net. With the same architecture structure, the major differences are the proposed bi-directional skip connections. Bottom: unrolled overview of our BiO-Net when $\mathbf{T} = 3$ and $\mathbf{L} = 4$. Note that in BiO-Net, same level encoder/decoder blocks share the same weights, such that no extra parameters are required.

Optimizing a recurrent network can be tricky, since multiple computation graphs are possibly involved in one network structure. In classic Recurrent Neural Networks (RNN) (Hochreiter and Schmidhuber, 1997), BackPropagation Through Time (BPTT) strategy is used to calculate gradients based on outputs at different time stamps. However, in our bidirectional networks, the blocks at the finest-resolution are not recursable and only one final segmentation mask is predicted during the entire inference regardless of recurrence time. To this end, the gradient computation graph remains consistent to an unrolled version of BiO-Net (Fig. 1, bottom). During backpropagation, the gradients are accumulated at each learnable layer and are subsequently used to update the weights for only once.

3.2. Analysis of Computational Burdens in BiO-Net

For a fair validation of the proposed bi-directional skip connections, our BiO-Net structure is constructed identically to U-Net with the same upsampling, fusion and channel expansion strategies. However, we find that the identical setting puts unexpected computation burdens on our BiO-Net and there exists efficient alternatives to reduce the overall complexity.

When using concatenation as the fusion strategy in BiO-Net, the encoder feature channels have to be doubled for carrying additional backward skipped features. We optimize such discord by replacing channel-wise concatenation to element-wise average pooling as the fusion strategy. Moreover, instead of using convolution transpose for upsampling with extra learning parameters, we bilinearly resize the coarse-scale feature maps to be aligned with the finer ones. As a result, the above simple optimizations reduce the complexity of BiO-Net with a 34.86 times reduction on the network parameters.

The optimization strategies are adopted in our other networks including BiO-Net++ and BiX-Net, which will be presented in detail shortly.

3.3. BiO-Net++: A Multi-scale Upgrade of BiO-Net

Multi-scale information provides better guidance for the analysis of various size objects (Yan et al., 2020). To fuse multi-scale features in BiO-Net, precedent encoded/decoded features at all levels are densely connected to every decedent decoding/encoding level through the proposed bi-directional skip connections. We average over the multi-scale features and align inconsistent spatial dimensions via simple bilinear resizing. The suggested multi-scale BiO-Net++ is outlined in Fig. 2, left.

4. Search for Efficient Multi-scale BiO-Net

4.1. BiX-NAS: Two-phase Search for Efficient BiO-Net++

Although BiO-Net++ promotes multi-scale feature fusion, empirically, we found that the *dense* skip connections bring marginal improvements in terms of the overall performance (Table 7). A natural question hence arises about whether every level-to-level skip in BiO-Net++ carries indicative information and if there exists a subset of *sparse* skip connections that not only benefit from multi-scale feature fusions but yield the least complexity.

To this end, we present BiX-NAS, a two-phase NAS algorithm, to automatically find optimal and sparsely connected skip

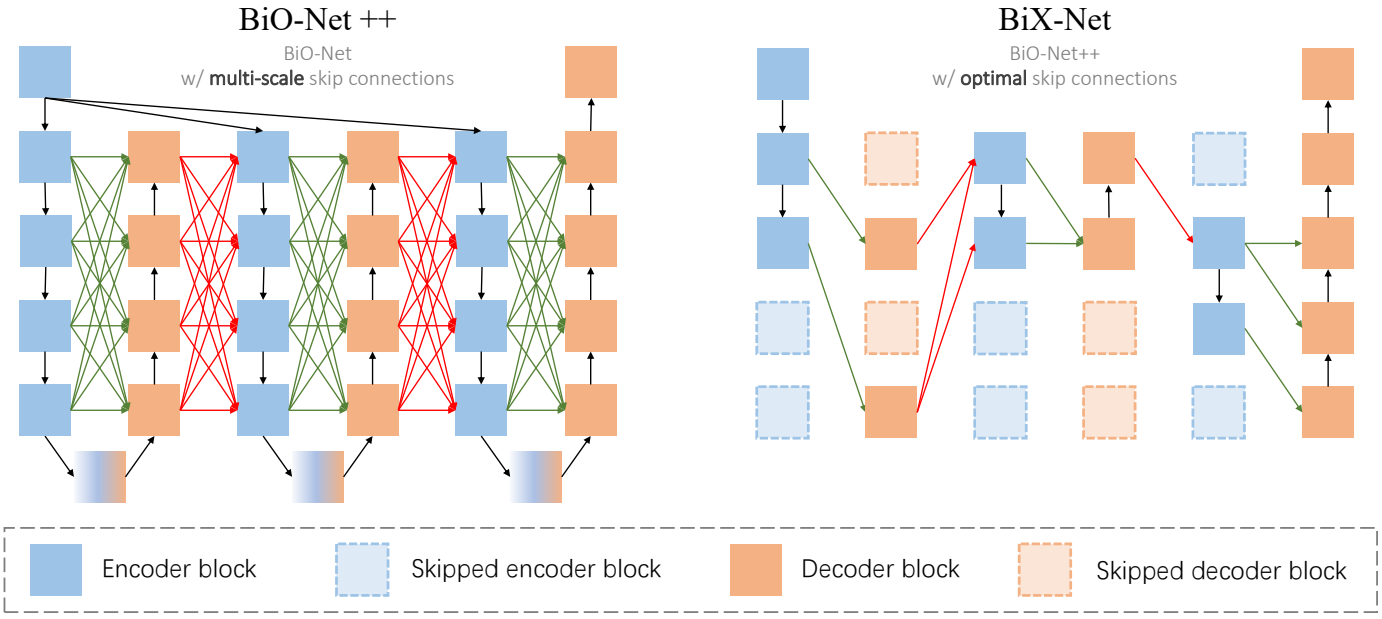

Fig. 2: Unrolled overview of our BiO-Net++ with $\mathbf{T} = 3$ and the finally searched BiX-Net. Building blocks are skipped if there is no connected path linking them to the post-processing block. Compared to the densely connected BiO-Net++, the sparser BiX-Net is more memory economic and computational efficient.

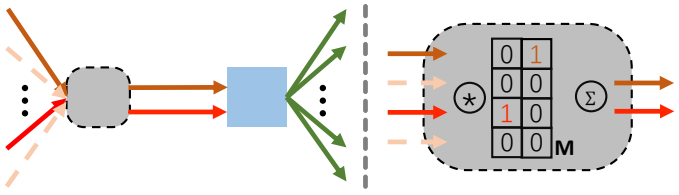

Fig. 3: The proposed selection matrix strategy with an exemplary encoder block. Two skips are selected from four incoming backward skips by the learnable one-hot matrix $\mathbf{M}$.

connections in the BiO-Net++. In Phase1, we follow a differentiable NAS strategy that aims at narrowing down the huge search space swiftly. In Phase2, we design an evolution-based NAS to progressively discover the best skips with the optimal segmentation performance and computation efficiency. In this study, we searched the final architecture on one dataset only and the same architecture is then transferred to all other tasks.

Phase1: Narrowing down search space via selection matrix. To identify a sparse set of skip connections, the dense ones between every pair of extraction stages are evaluated and sifted. Suppose there are N incoming feature streams in a desired searching block, we anticipate that only $k \in [1, N-2]$ candidate(s) of them could be accepted, which results in a search space of $\approx \sum_{k=1}^{(N-2)} \binom{N}{k}^{\mathbf{L}(2\mathbf{T}-1)}$ in the SuperNet BiO-Net++ with $\mathbf{L}$ levels and $\mathbf{T}$ iterations. When $N = 5, \mathbf{L} = 4$ and $\mathbf{T} = 3$ (a common setup), the search space expands to 5^{40}, escalating the searching difficulty for one optimal instance.

To alleviate such difficulty, we determine k *candidate skips* from N incoming skips in each searching block by reducing the easy-to-spot ineffective ones. Intuitively, one-to-one relaxation parameters α (Liu et al., 2019b) for each skip connection x could be registered and optimized along with BiO-Net++. The skip with the highest α is then picked as the output of $\mathbf{\Phi}(\cdot)$, such that $\mathbf{\Phi}(\cdot) = x_{\arg\max \boldsymbol{\alpha}}$, where $\mathbf{x} = \{x_1, \cdots, x_N\}$ and $\boldsymbol{\alpha} = \{\alpha_1, \cdots, \alpha_N\}$ denote the full set of incoming skips, and their corresponding relaxation scores, respectively.

The above formulation outputs a fixed number of skips for every level. However, different levels may fuse different number of skips and the above intuitive formulation cannot suffice. Towards a more flexible skip selection, we construct a learnable *selection matrix* $\mathbf{M} \in \mathbb{R}^{N\times(N-2)}$ that models the mappings between the N incoming skips and k anticipated candidates, and formulate $\mathbf{\Phi}(\cdot)$ as a fully differentiable equation below:

$$\mathbf{\Phi}(\mathbf{x}, \mathbf{M}) = \mathit{Matmul}(\mathbf{x}, \texttt{GumbelSoftmax}(\mathbf{M})), \tag{4}$$

where the `GumbelSoftmax` trick (Jang et al., 2017) forces each of the $(N-2)$ columns of $\mathbf{M}$ to be an one-hot vector that votes for one of the N incoming skips. Our formulation generates $(N-2)$ selected skips with repetition allowed, achieving a flexible selection of $[1, N-2]$ different candidate skips (Fig. 4). The *unique* candidate skips are then averaged out to be fed into subsequent blocks. Moreover, differing from (Liu et al., 2019b,a), our $\mathbf{\Phi}(\cdot)$ formulation unifies the forward propagation behaviour during both searching and network inference stages.

Phase2: Progressive evolutionary search. After squeezing the initial search space, evolving randomly sampled candidate network instances becomes practical. To further reduce skip redundancies and identify the best network instance, we perform an additional evolutionary search to find the optimal skip set at all levels across all iterations. Specifically, we search the candidate skips $\mathbf{C}$ for all levels at the same time between a certain pair of adjacent extraction stages, and then progressively move to the next pair once the current search is concluded. As the connectivity of adjacent extraction stages depends on the con-

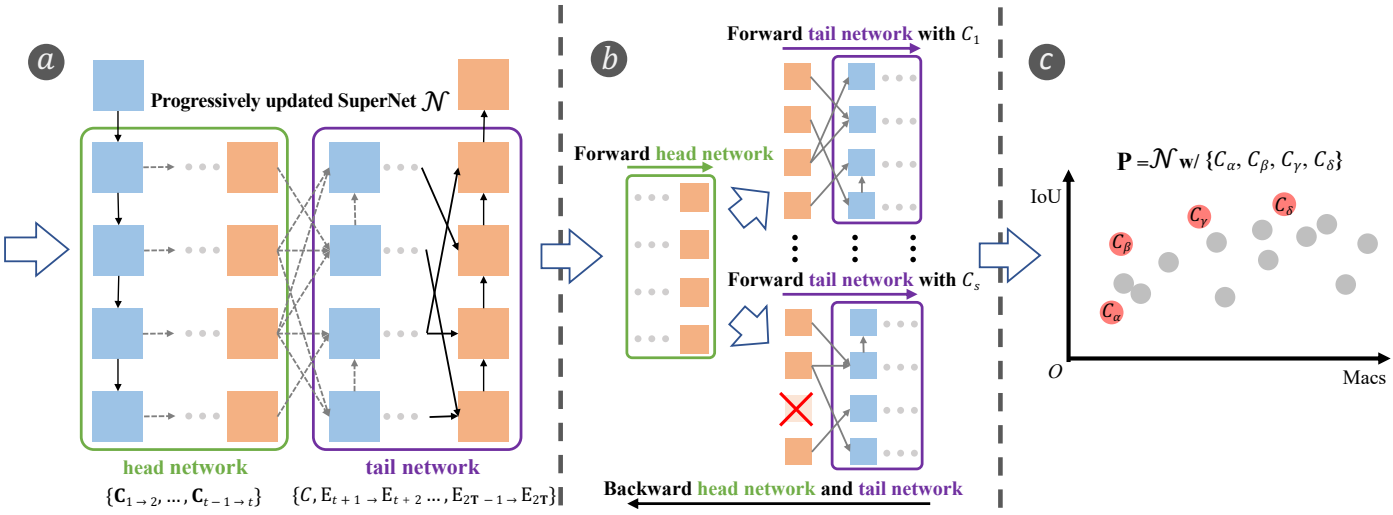

Fig. 4: Our progressive evolutionary search (Phase2) workflow. **(a)** Phase1 searched SuperNet $\mathcal{N}$ can be divided into *head network* and *tail network*. SuperNet $\mathcal{N}$ is sustainable to be updated with the newly evolved skips. **(b)** Proposed forward and backward schemes for head and tail networks. Only one backward pass is required to update all architecture instances. **(c)** Only the searched skips at the Pareto front of a population $\mathbf{P}$ are retained and used to update $\mathcal{N}$.

nectivity of precedent ones, we initiate the search from the last extraction stage pair and progressively move to the first pair.

The conventional evolutionary NAS algorithms optimize the SuperNet with each sub-network in a population $\mathbf{P}$ individually (Real et al., 2019, 2017), and then update $\mathbf{P}$ when all individual training finish. When searching for skip connections rather than operators, there are two major flaws of such strategy: first, optimizing SuperNet with sampled skip sets individually may result in unfair outcomes; second, the searching process could be empirically slow. Assuming the forward and backward of each extraction stage takes $\mathbf{I_F}$ and $\mathbf{I_B}$ time, training all $|\mathbf{P}|$ instances individually per step takes $2\mathbf{T}|\mathbf{P}|(\mathbf{I_F}+\mathbf{I_B})$ in total.

Improving searching fairness and efficiency. To overcome the first flaw above, we define the concept of *skip fairness* and claim that all skip search algorithms need to meet such principle.

Definition 1. Skip Fairness. Let $\mathbf{F} = \{\mathbf{f}^1, \cdots, \mathbf{f}^N\}$ be the incoming skipped features to any searching blocks in each evolving architecture $\mathcal{A}^i$ within a population $\mathbf{P}$. The skip fairness requires $\mathbf{f}^1_{\mathcal{A}^1} \equiv \cdots \equiv \mathbf{f}^1_{\mathcal{A}^{|\mathbf{P}|}}, \cdots, \mathbf{f}^N_{\mathcal{A}^1} \equiv \cdots \equiv \mathbf{f}^N_{\mathcal{A}^{|\mathbf{P}|}}\ \forall \mathbf{f}^i \in \mathbf{F}, \forall \mathcal{A}^i \in \mathbf{P}$.

The above principle yields that, when searching between the same extraction stage pair, any corresponding level-to-level skips across different sampled architectures are required to carry identical features. Otherwise, the inconsistent incoming features would impact the search decision on the skipping topology, hence causing unexpected search unfairness. Gradient-based search algorithms (e.g., Phase1 search algorithm) meet this principle by its definition, as the same forwarded features are distributed to all candidate skips equally. However, the aforementioned conventional strategy violates such principle due to the inconsistent incoming features produced by the individually trained architectures.

Our proposed Phase2 search algorithm meets the skip fairness by synchronizing partial forwarded features in all sampled candidate networks. Specifically, suppose we are searching skips between the t^{th} and $t{+}1^{th}$ extraction stages ($t \in [1, 2\mathbf{T}{-}1]$):

Algorithm 1 Progressive evolutionary search

Input: Iteration $\mathbf{T}$, sampling number $\mathbf{s}$, randomly initialized SuperNet weights $\mathcal{W}$, Phase1 searched candidate skips $\{\mathbf{C}_{1\to 2}, \cdots, \mathbf{C}_{2\mathbf{T}-1\to 2\mathbf{T}}\}$, criterion $\mathcal{L}$.
Output: BiX-Net with evolved skips $\{\mathbf{E}_{1\to 2}, \cdots, \mathbf{E}_{2\mathbf{T}-1\to 2\mathbf{T}}\}$

for $t = 2\mathbf{T}-1, \cdots, 1$ **do**
  for $i = 1, \cdots, \mathbf{s}$ **do**
    for each searching block b **do**
      Randomly sample n skips from $\mathbf{C}^b_{t\to t+1}$: $C^b_i, 1 \le n \le |\mathbf{C}^b_{t\to t+1}|$.
    end for
  end for
  for data batch X, target Y **do**
    Forward the head network with candidate skips $\mathbf{C}_{1\to 2}, \cdots, \mathbf{C}_{t-2\to t-1}$.
    for $i = 1, \cdots, \mathbf{s}$ **do**
      Forward each tail network with sampled skips C_i and previously evolved skips $\mathbf{E}_{t+1\to t+2}, \cdots, \mathbf{E}_{2\mathbf{T}-1\to 2\mathbf{T}}$.
      Calculate loss $l_i = \mathcal{L}(\mathrm{X}, \mathrm{Y})$
    end for
    Optimize $\mathcal{W}$ with the average loss $\frac{1}{\mathbf{s}}\sum_{i=1}^{\mathbf{s}} l_i$.
  end for
  Get Pareto front from $\{C_1, \cdots, C_{\mathbf{s}}\}$ and determine $\mathbf{E}_{t\to t+1}$.
end for

network topology from the 1^{st} to $t-1^{th}$ stages is fixed and the forward process between such stages can be shared. We denote such stages as *head network*. On the contrary, network topology from the t^{th} to $2\mathbf{T}^{th}$ stages varies as the changes of different sampled skips. We then denote such unfixed stages as *tail networks*, which share the same SuperNet weights but with distinct topologies. The forwarded features of head network are fed to all candidate tail networks individually, as shown in Fig. 4. We average the losses of all tail networks, and backward

Table 1: Details of the datasets used in our experiments. For CHAOS, number of data is reported for both 3D volumes and 2D slices.

	MoNuSeg	TNBC	CHAOS	Covid-19	Heart	Hippocampus
#training data	30	-	120(1594)	20	20	260
#testing data	14	50	-	-	-	-
Evaluation[1]	train/val	val	CV	CV	CV	CV
Input size	$512 \times 512 \times 3$	$512 \times 512 \times 3$	$256 \times 256 \times 1$	$160 \times 160 \times 80$	$192 \times 128 \times 80$	$56 \times 40 \times 40$
Modality	microscopy	microscopy	MRI	CT	MRI	MRI
#classes	2	2	5	3	2	3
Batch size	2	2	16	2	2	9

[1] Evaluation strategies. CV denotes 5-fold cross validation on all training data.

the gradients through the SuperNet weights only once. Besides, our Phase2 searching process is empirically efficient and overcomes the second flaw above, as one-step training only requires $\mathbf{I_B} + \sum_{t=1}^{2\mathbf{T}-1}(t\mathbf{I_F} + (2\mathbf{T} - t)\mathbf{I_F} \cdot |\mathbf{P}|)$.

After the search between an extraction stage pair completes, we follow a multi-objective selection criterion that retains the architectures on the Pareto front (Yang et al., 2020) based on both validation accuracy (IoU) and computational complexity (MACs). The proposed progressive evolutionary search details are presented in Algorithm 1 and our finally searched BiX-Net is shown in Fig. 2, right.

5. Experimental Setup

5.1. Datasets

The effectiveness of our methods was evaluated on four different tasks across six 2D and 3D publicly available datasets. Each task requires distinct segmentation targets and represents a different imaging modality. Details of the datasets are presented in Table 1.

5.1.1. MoNuSeg and TNBC

The MoNuSeg dataset (Kumar et al., 2017) consists of tissue image tiles at 40× magnification level from the TCGA (Tomczak et al., 2015) database. The original tissues were sampled from multiple patients with tumors in a variety of organs from different clinics. There are a total of 44 images each of 1000×1000 pixels, and they are divided into a training set of 30 images and a test set of 14 images. Following a common protocol (Graham et al., 2019), we extracted 512×512 patches at 4 corners of each image, which enlarges the dataset by 4 times. Neither numeric normalization nor stain normalization was performed as pre-processing on the dataset.

The TNBC dataset (Naylor et al., 2018) contains 50 sampled tiles from breast tissues with a size of 512×512 pixels. There is no specific training-testing split available for TNBC. Compared to the data in MoNuSeg, TNBC tiles were extracted from considerably independent sources and were processed with different staining and color adjustment strategies. Therefore, we used TNBC as an extra validation set to evaluate the generalization ability of our proposed method on nuclei segmentation task by training on the MoNuSeg dataset only.

5.1.2. CHAOS

The CHAOS (Combined Healthy Abdominal Organ Segmentation) dataset (Kavur et al., 2021) consists CT and MRI volumes of different organs including liver, left kidney, right kidney and spleen. Similar to (Yan et al., 2020), we use the training MRI image slices in this work to evaluate our methods on 2D multi-class organ segmentation task.

There are two types of MRI sequences in the dataset with each consists of 120 DICOM volumes: T1-DUAL (40 data for both in phase and out phase) and T2-SPIR (40 data). Each volume has being performed to scan abdomen under different radio frequency pulse and gradient combinations. The datasets are acquired by a 1.5T Philips MRI, which produces the 12 bit DICOM images with 256×256 resolution. The ISDs vary between 5.5-9 mm (average 7.84 mm), x-y spacing is between 1.36 - 1.89 mm (average 1.61 mm) and the number of slices is between 26 and 50 (average 36). In total, there is 1594 slices (532 slice per sequence) in the dataset. Before being processed by the networks, we performed additional pre-processing techniques on the raw sequences by min-max normalization and histogram auto-contrast that extend values to span over [0, 1].

5.1.3. Covid-19

We used the most recent Covid-19 chest CT datasets (Ma et al., 2020) to validate our methods on 3D segmentation tasks. Each of the 20 3D volumes in the dataset has been annotated into 4 classes: background, left lung, right lung, and Covid-19 infection. With focus on the segmentation of Covid-19 infection regions, following (Müller et al., 2020) we combine both lung classes, resulting in a 3-class segmentation task. The CT volumes have an average number of 176 slices, and were collected either from the Coronacases Initiative or the Radiopaedia with a spatial resolution of 512×512 for Coronacases Initiative and 630×630 for Radiopaedia.

For fair comparisons, we followed the preprocessing strategies introduced in (Müller et al., 2020). Specifically, based on the Hounsfield units (HU), the raw pixel intensity values within [-1250, 250] are kept, which cover the range of valid lung regions ([-1000, -700]) and Covid-19 infection regions ([50, 100]). The clipping was only applied on the Coronacases Initiative CTs. We then performed z-score normalization on the clipped data to obtain the standardized gray-scale values. To resample the volumes for better neural network training, we adopted a target spacing of $1.58 \times 1.58 \times 2.70\ mm^3$ and resam-

Table 2: Comparison on MoNuSeg testing set and TNBC. **AS** denotes if the network is automatically searched. **RC** denotes if the network is recurrent. Highlighted cells represent the results significantly lower than BOTH BiO-Net and BiX-Net based on the statistical test (p-value < 0.05). For each metric, the best result is in bold and the second best result is in underline.

			MoNuSeg		TNBC			
Methods	**AS**	**RC**	**IoU (%)**	**DICE (%)**	**IoU (%)**	**DICE (%)**	**#Params**	**MACs**[1]
U-Net	✗	✗	68.2±0.3	80.7±0.3	46.7±0.6	62.3±0.6	8.64 M	65.83 G
U-Net++	✗	✗	69.4±0.3	81.5±0.4	53.9±0.4	67.2±0.5	9.16 M	138.60 G
Att U-Net	✗	✗	68.3±0.2	81.1±0.2	56.4±0.5	69.9±0.6	8.73 M	66.97 G
R-UNet (**T** = 2)	✗	✓	68.5±0.2	80.8±0.2	53.7±0.4	66.0±0.6	4.50 M	103.24 G
R-UNet (**T** = 3)	✗	✓	68.8±0.3	81.1±0.2	55.5±0.5	68.3±0.5	4.50 M	154.86 G
R2U-Net (**T** = 2)	✗	✓	68.8±0.4	80.9±0.3	56.2±0.6	68.9±0.7	9.78 M	152.89 G
R2U-Net (**T** = 3)	✗	✓	69.1±0.3	81.2±0.3	60.1±0.5	71.3±0.6	9.78 M	197.16 G
NAS-UNet	✓	✗	68.4±0.3	80.7±0.3	54.5±0.6	69.6±0.5	2.42 M	67.31 G
AutoDeepLab	✓	✗	68.5±0.2	81.0±0.3	57.2±0.5	70.8±0.5	27.13 M	60.33 G
MS-NAS	✓	✗	68.8±0.4	80.9±0.3	58.8±0.6	71.1±0.5	14.08 M	72.71 G
BiO-Net (**T** = 3)	✗	✓	**69.9±0.2**	82.0±0.2	62.2±0.4	75.8±0.5	14.99 M	115.67 G
BiX-Net	✓	✓	69.9±0.3	**82.2±0.2**	**68.0±0.4**	**80.8±0.3**	**0.38 M**	**28.00 G**

[1] MACs are calculated based on the input size of $512 \times 512 \times 3$.

Table 3: Quantitative comparison on CHAOS MRI. Darker highlighted cells represent the results are significantly lower than BOTH BiO-Net and BiX-Net (p-value < 0.05). Lighter highlighted cells represent the results are significantly lower than BiO-Net (p-value < 0.05). For each metric, the best result is in bold and the second best result is in underline.

	Liver		Left Kidney		Right Kidney		Spleen	
Methods	mIoU (%)	DICE (%)	mIoU (%)	DICE (%)	mIoU (%)	DICE (%)	mIoU (%)	DICE (%)
U-Net	78.1±2.0	86.8±1.8	61.3±1.1	73.8±1.2	63.5±1.1	76.2±1.1	62.2±2.1	74.4±2.3
U-Net++	78.8±1.7	87.5±1.6	65.1±1.3	74.7±1.4	66.0±1.3	77.2±1.7	64.4±1.6	75.6±1.5
R-UNet	78.2±2.1	86.7±1.9	63.6±1.5	75.0±1.6	64.4±1.3	75.8±1.7	63.4±2.0	75.3±1.4
R2U-Net	77.9±1.9	86.4±1.8	63.7±1.7	75.1±1.6	64.6±1.9	76.1±1.9	63.2±1.9	75.0±1.5
NAS-UNet	79.1±1.8	87.2±1.8	65.5±1.5	75.0±1.3	66.2±1.2	77.7±1.0	64.1±1.3	75.8±1.6
AutoDeepLab	79.8±1.9	88.1±1.8	66.7±1.6	75.0±1.7	61.9±0.9	75.7±1.1	63.9±1.2	75.5±1.4
MS-NAS	72.6±2.3	82.6±2.1	71.0±1.3	81.9±1.3	70.1±1.9	81.1±1.8	62.5±2.1	74.0±2.3
BiO-Net	**85.8±2.0**	**91.7±1.8**	**75.7±1.1**	**85.1±1.2**	**78.2±1.0**	**87.2±1.1**	**73.2±2.3**	**82.8±2.3**
BiX-Net	82.6±1.5	89.8±1.5	71.0±1.0	82.1±1.1	71.9±0.8	82.7±1.0	66.0±1.7	76.5±2.0

pled all data to the median shape of $267 \times 254 \times 104$ mm^3.

5.1.4. *Heart and Hippocampus*

The Heart and Hippocampus datasets are parts of the Medical Segmentation Decathlon (MSD) (Simpson et al., 2019), which is a commonly used 3D medical image segmentation benchmark. The Heart dataset contains 20 training MRI volumes scanned over the entire heart during a single cardiac phase. Images were originally obtained with the voxel resolution $1.25 \times 1.25 \times 2.7 mm^3$. The target of this task is the segmentation of left atrium, resulting in a binary segmentation task. The Hippocampus dataset includes MRI volumes acquired in 90 healthy adults and 105 adults with a non-affective psychotic disorder (56 schizophrenia, 32 schizoaffective disorder, and 17 schizophreniform disorder). All collected volumes are in the resolution of $1.0 \times 1.0 \times 1.0 mm^3$ with labels for anterior and posterior regions, resulting in a three-class segmentation task.

Our pre-processing strategy follows a common protocol (Isensee et al., 2018) which includes 3 main steps. First, due to the large number of ineffective signals (i.e., background) exist in the datasets, feeding the raw volumes directly to the networks is computational expensive and time consuming. We thus cropped the data to the region of nonzero values with at least one non-background voxel presented in each volume. Second, similar to the Covid-19 dataset, we resampled voxel spacing to ease network training. For the Heart dataset, raw volumes were resampled to a target shape of $115 \times 320 \times 232$ voxels. For the Hippocampus dataset, the target shape is $36 \times 50 \times 35$ voxels. Third order spline interpolation was used for input volumes and nearest neighbor interpolation was used for the ground truth masks. Finally, all volumes were z-score normalized individually before being fed into the networks. When the cropping step drops more than 1/4 of the average sizes, we applied the normalization only within the nonzero regions.

5.2. *Evaluation Metrics*

Multiple evaluation metrics are used to quantify the experimental results. For the 2D datasets (MoNuSeg, TNBC, and CHAOS), we rely on mean Intersection over Union (mIoU) and Dice coefficient (DICE) scores to evaluate the models' performances. The two metrics are good indicators of the rate of true positive predictions. For the 3D datasets (Covid-19, Heart, and

Hippocampus), metrics including the DICE score, sensitivity and specificity are used. Additionally, total number of network parameters and Multiple-ACcumulate operations (MACs) are also reported for network complexity.

For the datasets without standardized train/test splits (CHAOS, Covid-19, Heart and Hippocampus), we adopted 5-fold cross validation (CV) on the training data.

5.3. Statistical Analysis

For all comparison experiments, we report the average results along with the standard deviation of multiple independent runs for each model. For the datasets with clear testing set split (MoNuSeg and TNBC), we repeat the evaluation three times with different random seeds. For the datasets that require cross-validation (CHAOS, Covid-19, Heart and Hippocampus), we compute the statistics across all validation folds.

Additionally, we perform two-tail paired t-test to analyze the statistical significance between our methods and the competing methods. Smaller p-values indicate greater statistical significance of improvement that our methods have achieved.

6. Implementation Details

Our empirical experiments are designed as follows: **(1)** We firstly compare our BiO-Net and BiX-Net with other state-of-the-art segmentation networks on the 2D nuclei segmentation tasks and the multi-class organ segmentation task. **(2)** Then, we extend our networks to 3D segmentation tasks, compare to the counterparts that particularly designed for 3D data. **(3)** Extensive ablation studies are conducted on our BiO-Net and BiX-NAS separately using the nuclei segmentation datasets. Note that we utilized only the MoNuSeg dataset for searching the BiX-Net, and the same architecture is then transferred to all other 2D and 3D tasks.

6.1. Network Implementation Details

BiO-Net implementation details. Unless explicitly specified, our BiO-Net is constructed with an encoding depth of $\mathbf{L} = 4$ and a backward skip connection built at each stage of the network, except for pre-processing and post-processing blocks. As a balance between computation efficiency and segmentation performance, we set the recurrence time $\mathbf{T} = 3$, such that all backward skip connections are triggered 2 times (relevant studies are presented in Sec. 7.2.2).

BiX-Net searching details. Following the same set up as BiO-Net, we construct the SuperNet BiO-Net++ with $\mathbf{L} = 4$ and $\mathbf{T} = 3$ as well, resulting in the same hierarchy in the finally searched BiX-Net. In Phase1 search process of our BiX-NAS, we jointly optimize the weights of SuperNet and the selection matrices using the same optimizer, rather than optimized separately (Liu et al., 2019b). The Phase1 searching process took roughly 0.09 GPU-Day. In Phase2, there were in total 5 searching iterations when $\mathbf{T} = 3$. At each iteration, for each retained architecture from the preceding iteration, we sampled $\mathbf{s} = 15$ different skip sets to form the new $\mathbf{P}$. Due to GPU memory limitation, we applied a channel expansion parameter of $\mathbf{W} = 0.75$ and only retained less than three final architectures ranked by IoU at the Pareto front per search iteration. Our Phase2 searching process consumed 0.37 GPU-Day. After Phase2 searching finished, we inserted the searched bi-directional skip connections into a re-initialized encoder-decoder architecture. For the blocks that have been discarded by our search algorithm, we abandoned all computations (i.e., convolutions, normalizations and activations) in such blocks but still resized the incoming features to the corresponding scale.

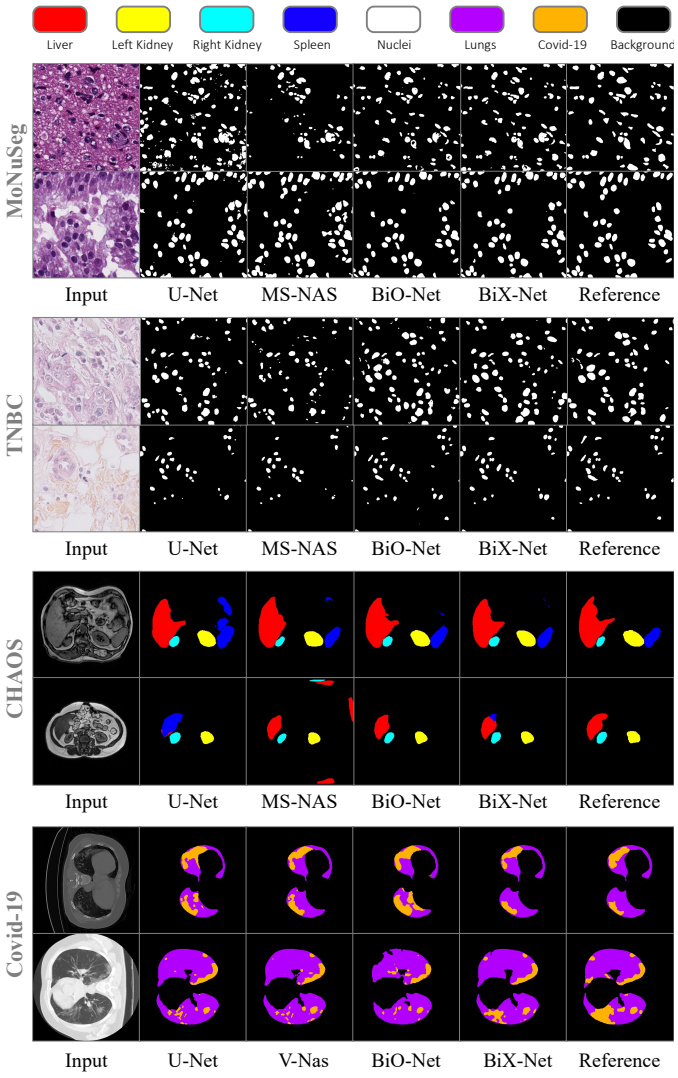

Fig. 5: Qualitative comparisons between our BiO-Net, BiX-Net and other competing methods.

6.2. Training Details

All our training configurations followed common usages without any explicit tuning. The competing methods were trained under the identical settings for fair comparison. Note that we utilized only the MoNuSeg dataset for searching the BiX-Net, and the same architecture is transferred to all other 2D and 3D tasks.

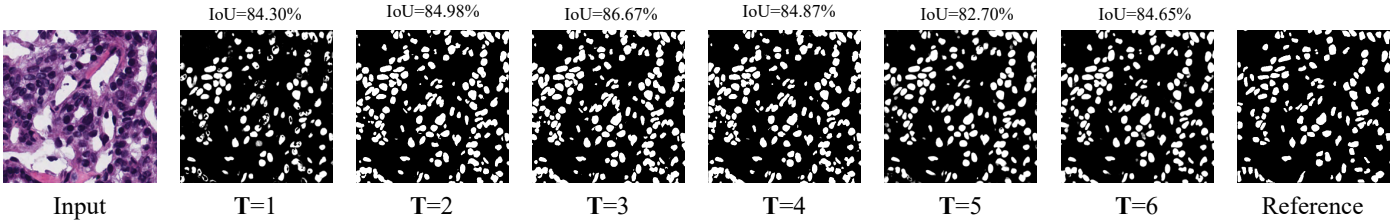

Fig. 6: Qualitative results on different recurrence time **T** in BiO-Net. Instance IoU scores are shown as well.

Table 4: Quantitative comparison on the Covid-19 dataset. For each metric, the best result is in bold.

	Lungs			Infection			
Methods	DICE (%)	Sens. (%)	Spec. (%)	DICE (%)	Sens. (%)	Spec. (%)	#Params
3D U-Net	95.6±2.6	95.6±2.4	99.8±0.2	76.1±10.7	73.0±15.3	99.9±0.0	22.6 M
V-Nas	93.3±4.4	94.8±3.0	99.8±0.0	76.0±11.3	73.7±15.8	99.9±0.0	1.6 M
UXNet	**95.8±1.4**	95.3±1.7	99.8±0.0	75.4±6.1	72.4±11.1	99.9±0.0	9.5 M
BiO-Net	91.0±7.7	92.4±8.2	99.7±0.2	75.1±10.5	**77.8±14.0**	99.8±0.2	44.9 M
BiX-Net	94.8±1.8	**96.1±0.7**	**99.8±0.0**	**78.1±8.1**	76.6±10.2	**99.9±0.0**	**1.4 M**

6.2.1. MoNuSeg and TNBC

Standard data augmentation strategies were applied to the input data on-the-fly including: random rotation (within the range [-15°, +15°]), random translation (in both x- and y-directions; within the range of [-5%, 5%]), random shearing, random zooming (within the range [0, 0.2]), and random flipping (both horizontally and vertically). We train all models using the Adam (Kingma and Ba, 2015) optimizer for 300 epochs with batch size 2. For manually crafted networks including BiO-Net, the initial learning rate was set to 0.01 with a decay rate of 0.003 at every epoch. For all NAS searched networks including BiX-Net, the initial learning rate was set to 0.001 with the same decay rate.

During the Phase1 search, we trained the BiO-Net++ SuperNet for 300 epochs with a base learning rate of 0.001 and a decay rate of 3e-3. During the Phase2 search, at each searching iteration, we trained the sampled candidate networks 40 epochs starting from a learning rate of 0.001 and then decayed by 0.1 every 10 epochs.

6.2.2. CHAOS

Since there are many more training slices in the CHAOS dataset than the nuclei segmentation datasets, we lowered the initial learning rate to 0.001, which remains consistent across both handcrafted and NAS searched networks. The learning rate was cosine-annealing scheduled to 10^{-7} in 300 epochs. Data augmentations including random spatial translation within the range of [-30%, 30%], random flipping, and random elastic transformation with 1.5 alpha and 0.07 sigma were used. We trained all models using the SGD optimizer for 300 epochs with momentum 0.9, weight decay 0.0001 and batch size 16.

6.2.3. Covid-19

Differing from the above 2D segmentation tasks, we employed the Adam optimizer to minimize the both the cross entropy loss and the Tversky loss (Salehi et al., 2017) with 0.5 alpha, 0.5 beta and batch size 2. The initial learning rate is set to 0.0003 for BiO-Net and BiX-Net and 0.001 for other competing methods (recommended settings in (Müller et al., 2020)). Learning rates were scaled down by 0.1 once learning stagnates with a patience of 20 epochs. The minimum limit of the learning rate is $1e^{-5}$. Following (Müller et al., 2020), the data augmentation process includes: random mirroring, random elastic transformations, random rotations, random brightness enhancement, random contrast, random gamma corrections, and random Gaussian noise injections. The probability of triggering each of the augmentation strategy is 15%.

Due to GPU memory limitations, the networks were trained with $160 \times 160 \times 80$ voxel patches randomly sampled from the raw volumes. During inference, we sampled the patches via the sliding window strategy enabling an overlap of half the patch size ($80 \times 80 \times 40$ voxels) to stabilize boundary predictions.

6.2.4. Heart and Hippocampus

For fair comparisons, we followed the same training settings and augmentation strategies introduced in (Isensee et al., 2018). Adam optimizer with an initial learning rate of 0.0003 was used across all experiments. We trained the networks with randomly sampled patches from the pre-cropped volumes. For better stabilizing optimization, we ensure that at least 1/3 of each batch contains data in foreground class. We denote 250 training steps as one epoch, and set the maximum epoch limit to be 1000. Similar to Covid-19, the learning rate was reduced by a factor of 5 when the training loss does not change by at least 0.005. We stopped the training when the learning rate fell below 0.000001 and no significant change is observed in the training loss. The augmentation strategies include random rotations, random scaling, random elastic deformations, gamma corrections, and mirroring.

7. Results

In this section, we present the experimental results obtained following the implementation details introduced in Sec. 6.

Table 5: Quantitative comparison on the Heart and Hippocampus datasets. For each metric, the best result is in bold.

	Heart			Hippocampus					
Class	1			1			2		
Methods	DICE (%)	Sens. (%)	Spec. (%)	DICE (%)	Sens. (%)	Spec. (%)	DICE (%)	Sens. (%)	Spec. (%)
nnU-Net[1]	93.3±0.8	93.9±1.4	92.9±2.0	89.8±0.4	89.5±1.0	90.3±0.7	88.1±0.2	**88.0±0.4**	88.4±0.5
BiO-Net	**93.4±0.8**	**94.0±1.5**	93.0±2.1	**89.9±0.4**	**89.8±0.9**	90.2±0.8	**88.2±0.3**	**88.0±0.5**	88.6±0.6
BiX-Net	**93.4±1.4**	93.9±2.1	**93.1±2.2**	**89.9±0.2**	89.5±0.9	**90.6±0.7**	**88.2±0.2**	**88.0±0.6**	**88.7±0.9**

[1] nnU-Net (Isensee et al., 2018) modifies the original U-Net structure and utilizes different network variants for different tasks. Here we use their official model checkpoints for fair comparison.

7.1. Comparison Results

7.1.1. MoNuSeg and TNBC

The results on the MoNuSeg and TNBC datasets are reported in Table 2. We compare our networks with three types of counterparts: **(1)** state-of-the-art U-Net variants: U-Net++[2] (Zhou et al., 2018), Attention U-Net[3] (Oktay et al., 2018); **(2)** recurrent networks: R-UNet[4] (Wang et al., 2019), R2U-Net[5] (Wang et al., 2019); **(3)** NAS searched networks: NAS-UNet[6] (Weng et al., 2019), AutoDeepLab (Liu et al., 2019a), MS-NAS (Yan et al., 2020).

Although the U-Net variants demonstrate good performances on the validation set of MoNuSeg, they require a large number of network parameters. When validated on the TNBC dataset that has different data distributions, the generalization ability of such networks becomes limited. One advantage of recurrent networks is that the number of parameters remains the same with increased recurrence time. By recursing through the same network layers, performance is improved on both MoNuSeg and TNBC datasets. However, as discussed earlier, the computational costs (MACs) inevitably escalate with more recurrence time. With a good balance between segmentation performance and network complexity, the NAS searched networks yield better generalization results on the TNBC dataset than the manually crafted U-Net variants.

By incorporating our proposed bi-directional skip connections, BiO-Net demonstrates superior performances on both datasets. However, when $\mathbf{T} = 3$, an increase of 75.7% MACs is observed against the plain U-Net baseline. As expected, our proposed BiX-NAS successfully discovered an optimal network instance with minimum computational costs. Our resource-aware BiX-Net achieves on par performance to the heavy BiO-Net on the MoNuSeg validation set and even better generalization results on the TNBC dataset.

7.1.2. CHAOS

Giving the great results achieved on binary segmentation tasks, we then examined if the same performance gain can be obtained on a multi-class segmentation task. The results on organ segmentation tasks for the CHAOS dataset are shown in Table 3. Compared to other competing methods, our BiO-Net and BiX-Net demonstrate superior performances on all metrics across all classes. It is also important to note that our efficient BiX-Net was searched on MoNuSeg data only, and the results show that it can directly transfer to the CHAOS dataset and surpasses all comparison methods with the least network complexity. Although the results achieved by BiX-Net are slightly lower than the plain BiO-Net, BiO-Net suffers from computation burdens, which are a ×4 of computational complexity, and a ×34.9 of trainable parameters.

7.1.3. Covid-19

After the evaluations on 2D datasets, we then validate our BiO-Net and BiX-Net on 3D segmentation tasks. First, we present the quantitative results on the Covid-19 dataset in Table 4. The compared methods include: the baseline network 3D U-Net (Çiçek et al., 2016), the state-of-the-art NAS searched networks V-Nas (Zhu et al., 2019) and UXNet (Ji et al., 2020). Since 3D segmentation is much more challenging than the 2D tasks, the results achieved by our BiO-Net and BiX-Net are not statistically significant (i.e., p-values > 0.05) compared to other methods and therefore no cells are highlighted in Table 4. However, except for the DICE score on lung segmentations, our proposed networks yield the highest results on all other metrics. Our findings are as follows: **(1)** A naive bi-directional network recursion for 3D segmentation cannot bring the same performance enhancement as in 2D tasks. This can be observed by comparing BiO-Net with the non-recursable 3D U-Net. Our understanding is that 3D features are more complex than the 2D ones. Therefore, the network would easily saturate with the same capacity. The same behaviour can also be observed in Sec. 7.2.1 where BiO-Net can hardly benefit from bi-directional skip connections with limited parameters. **(2)** BiX-Net stands out with superior results on almost all metrics. Noticeably, our BiX-Net is searched on a single 2D dataset (MoNuSeg), and the great performances prove the generalization ability to transfer the identical architecture to tasks in significant different domains.

7.1.4. Results on Heart and Hippocampus

Finally, our proposed methods were evaluated on the Heart and Hippocampus datasets from the medical segmentation decathlon. We compare our networks with the second place in the MSD official leader board: nnUNet, and used their provided

[2] https://github.com/4uiiurz1/pytorch-nested-unet
[3] https://github.com/LeeJunHyun/Image_Segmentation
[4] https://github.com/kcyu2014/recurrent-unet
[5] https://github.com/LeeJunHyun/Image_Segmentation
[6] https://github.com/tianbaochou/NasUnet

Table 6: Ablative results on backward skip connections in BiO-Nets. IoU (DICE) and number of parameters are reported.

	MoNuSeg (%)			TNBC (%)			
	T = 1	**T** = 2	**T** = 3	**T** = 1	**T** = 2	**T** = 3	#Params
Reference[1]	**68.0(80.3)**	**69.4(81.6)**	**69.9(82.0)**	**45.6(60.8)**	**54.8(69.3)**	**61.8(75.1)**	15.0 M
M = 1.25	68.5(81.3)	69.8(81.9)	69.5(81.7)	49.0(63.7)	55.7(69.7)	62.3(75.8)	23.5 M
M = 0.75	67.6(80.0)	67.8(80.5)	69.1(81.5)	51.6(66.1)	57.1(71.0)	59.8(73.8)	8.5 M
M = 0.50	66.8(79.2)	68.0(80.6)	69.1(81.4)	49.1(64.4)	54.3(67.9)	61.1(74.2)	3.8 M
M = 0.25	66.7(79.1)	67.8(80.4)	67.7(80.4)	52.4(67.4)	53.5(67.8)	57.5(71.0)	0.9 M
W = 3	68.0(80.3)	69.4(81.7)	68.8(81.4)	45.6(60.8)	51.0(65.6)	62.0(75.7)	15.0 M
W = 2	68.0(80.3)	67.2(80.1)	68.6(81.3)	45.6(60.8)	52.7(67.9)	60.1(74.2)	14.9 M
L = 3	68.1(80.6)	67.9(80.5)	68.9(81.2)	61.3(74.2)	59.4(73.3)	61.5(74.1)	3.8 M
L = 2	69.0(81.0)	69.5(81.7)	69.7(81.8)	59.6(73.4)	64.7(77.5)	59.6(73.5)	0.9 M

[1] Our adopted BiO-Net set up with **M** = 1.0, **W** = 4, and **L** = 4.

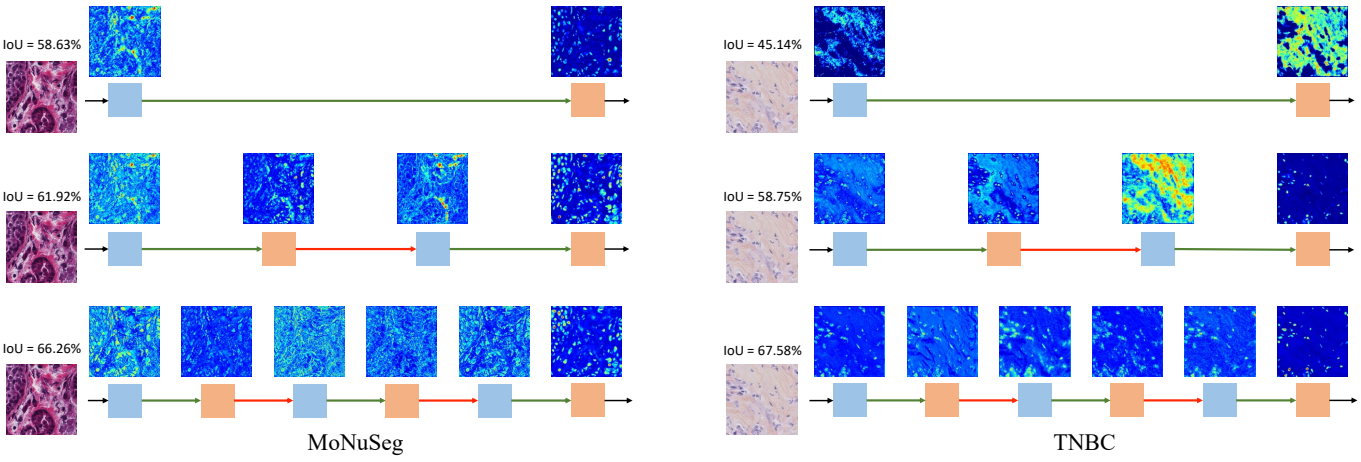

Fig. 7: Visualizations of the averaged feature maps obtained at different iterations. Instance IoU scores for the input image are reported.

model checkpoints[7] to produce their predictions. The comparisons are reported in Table 5. Due to the difficulty of the tasks, our methods only demonstrate marginal improvements. However, such improvements can be observed on nearly all the metrics. Noteworthy, as a computational efficient mobile network, our BiX-Net is able to achieve the state-of-the-art results.

7.1.5. *Qualitative Results*

For more intuitive comparisons, we present two qualitative segmentation results on each of the MoNuSeg, TNBC, CHAOS, and Covid-19 datasets in Fig. 5. Clearly, our BiO-Net and BiX-Net produce the most accurate segmentation masks. According to the visualization results, there are two primary advantages of our methods: **(1)** Our methods include less noise in the final prediction (MoNuSeg first case, TNBC first case). **(2)** Our methods produce the least false positive predictions (CHAOS).

7.2. *Ablation Studies*

We conducted extensive ablative experiments to inspect the behaviours of our methods under different setups. Unless explicitly specified, we used the nuclei segmentation datasets for the ablation studies.

[7] https://github.com/MIC-DKFZ/nnUNet

7.2.1. *Effectiveness of Bi-directional Skip Connections*

To better investigate the effectiveness of our backward skip connections, we here study the impact of 3 different factors under different scenarios: different network size controlled by a channel expansion parameter **M**, different number of backward skip connections used from the deepest encoding level **W** and different encoding depth of the network **L**. For more comprehensive studies, we report the ablative results on different recurrence time from **T** = 1 to **T** = 3. We ignore all other influential factors and use the plain BiO-Net for evaluation.

We present the ablation results impacted by different factors in Table 6. With the increasing recurrence time, performance improvements can be easily observed on nearly all experiments. Specifically, we observe that: **(1)** When the number of training parameters is restricted (e.g., **M**=0.25), our BiO-Net could hardly benefit from reusing convolutional layers through bi-directional skip connections. This observation aligns with the ones spotted in 3D segmentation tasks. **(2)** Without significant reductions on training parameters, using less backward skip connection while maintaining the same forward ones (e.g., **W**=3) causes slight performance drop on MoNuSeg when **T** increases. However, the generalization results on TNBC appear even better than the reference setting, with further 0.6 improvement on the DICE score. **(3)** Surprisingly, building bi-

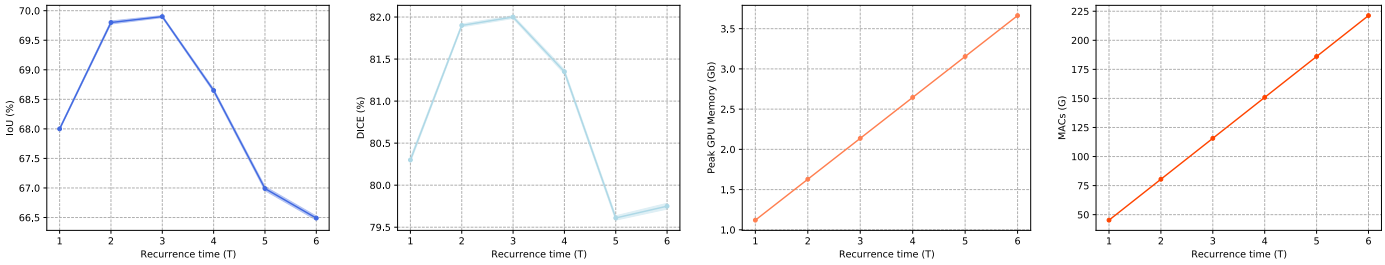

Fig. 8: Ablative results on different recurrence time **T** in BiO-Net. GPU memory is measured during network inference.

directional skips on a shallower network (e.g., **L**=3) not only reduces the total parameters but leads to impressive results on the MoNuSeg datasets. The same results performance boost was however not observed on the TNBC dataset.

Given the quantitative improvements brought by our bi-directional skip connections, one may wonder if the additional recurrence could actually generate meaningful features, instead of generating replicated or uninformative values. We visualize the averaged feature maps inferred at different recurrence steps in Fig. 7. The averaged feature maps vary along different time steps, validating that the recurrence inference through bi-directional skip connections is indeed able to generate indicative feature representations.

7.2.2. *Impact of Recurrence Time*

With the help of backward skip connections, our network is able to recurse through the encoder and decoder for an iterative refinement of the features. Under the memory constraints, we are interested in inspecting how recurrence time **T** impacts the final segmentation results. To this end, we conducted experiments on varying the recurrence time for $\mathbf{T} \in [1, 6]$. In addition to the metrics measuring segmentation performances, extra metrics including peak inference GPU memory usage and MACs are reported to demonstrate the computational burdens change along with the increasing recurrent time.

Although recursing through encoder and decoder earns extra performance improvement, we hypothesize that our BiO-Net would eventually saturate giving the fixed training parameters, and results in performance drops after a certain time of recurrence. As shown in Fig. 8, segmentation performances of BiO-Net consistently increase when $\mathbf{T} \leq 3$ and reach the greatest results at $\mathbf{T} = 3$. However, a longer network recurrence harms the feature representation and leads to even worse results. When $\mathbf{T} \geq 5$ both IoU and DICE scores are even poorer than $\mathbf{T} = 1$. Also, we note the peak GPU memory usage and MACs are linearly increased along with the recurrence time, hence, unrestricted recurrent inference is impractical. In Fig. 6, we compare the network predictions under different maximum **T**. The above observations validate the demand of a resource-aware and efficient substitute, which motivate our BiX-Net design. As a good trade-off between network performance and computation costs, we adopt $\mathbf{T} = 3$ to be our default setting and search BiX-Net across 6 extraction stages only (3 encoders + 3 decoders).

7.2.3. *Necessity of the Progressive Evolutionary Search*

We claimed in Sec. 4.1 that the Phase1 searched architecture still has computation redundancies and a following evolution-based Phase2 search can spot more effective and efficient subset skips. We conducted two extensive experiments to prove the necessity of the proposed progressive evolutionary search by training and evaluating the SuperNet BiO-Net++ and the Phase1 searched architecture directly.

We report the quantitative results on different intermediate networks in Table 7. Compared to BiO-Net that is constructed of identical building blocks and fusion functions to the plain U-Net, the modified BiO-Net++ requires only 1/34.86 parameters and 1/4 computations. With the help of multi-scale feature fusions, BiO-Net++ achieves even better results on both datasets. After performing BiX-NAS to reduce the skip redundancies in BiO-Net++, our Phase1 searched architecture reduces the computations by 8.6% further with nearly no influences on the segmentation results. Finally, our Phase2 searched BiX-Net shrinks the computation of BiO-Net++ by 17% and still achieves on par results.

7.2.4. *Efficiency of the Progressive Evolutionary Search*

We theoretically analyzed that our proposed searching method is computational efficient and could potentially lead to better results by adhering the skip fairness concept. Here we provide empirical validations on our claims: First, we compare two counterpart strategies: **(1)** randomly search skip connections across all extraction stages from the SuperNet (i.e., without progressive evolution); and **(2)** progressively search skip connections and train each network instance independently (i.e., without adhering skip fairness). Then, we construct the optimal network instances searched by the above strategies, and compare the searching cost and retraining performances on the MoNuSeg and TNBC datasets.

As shown in Table 8, without an evolution schema, searching directly on randomly sampled skip connections leads to a sub-optimal architecture instance. The results on both datasets are the lowest and the search process requires a great amount of time. As discussed earlier, a limited number of samples will not suffice to completely explore the vast search space and the true optimal instance is unlikely to be covered.

By evolving the skip connections from the last pair of extraction stages to the first, the final network instance achieves much better performances on both datasets. However, independent instance training violates the skip fairness and consumes an unneglectable search time, thus resulting in inferior segmentation

Table 7: Ablative results on different searching phases.

	MoNuSeg		TNBC					
Methods	IoU (%)	DICE (%)	IoU (%)	DICE (%)	#Params	Overhead[1]	MACs	Overhead[1]
BiO-Net	69.9±0.2	82.0±0.2	62.2±0.4	75.8±0.5	14.99 M	3845%	115.67 G	313%
BiO-Net++	70.0±0.3	82.2±0.3	67.5±0.4	80.4±0.5	0.43 M	13%	34.36 G	23%
Phase1 searched	69.8±0.2	82.1±0.2	66.8±0.6	80.1±0.4	0.43 M	13%	31.41 G	12%
BiX-Net	69.9±0.3	82.2±0.2	68.0±0.4	80.8±0.3	**0.38 M**	0%	**28.00 G**	0%

[1] Overhead compared to BiX-Net.

Table 8: Ablative results on different Phase2 search strategies.

	MoNuSeg		TNBC			
Phase2 search strategies	IoU (%)	DICE (%)	IoU (%)	DICE (%)	Search time	MACs
Random search	67.5±0.4	80.4±0.4	57.4±0.5	69.8±0.6	0.69 GPU/day	30.40 G
Independent training	68.4±0.3	81.1±0.4	60.9±0.6	73.6±0.7	0.73 GPU/day	29.68 G
Progressive evolutionary search	**69.9±0.3**	**82.2±0.2**	**68.0±0.4**	**80.8±0.3**	**0.37 GPU/day**	**28.00 G**

performances and exceeding search time.

Eventually, with the proposed progressive evolutionary search, our BiX-Net achieves the best performances in terms of all metrics. By sharing the head network among the candidates, the skip fairness is perfectly met and greatly reduces the overall search time. Compared to other search strategies, our proposed algorithm only demands about half of the time, proving the efficiency and effectiveness of our progressive evolutionary search.

8. Conclusion

In this work, we studied the skip schema in encoder-decoder architectures. Paired to forward skip connections, we proposed backward skip connections to ship decoded features back to encoder. The bi-directional skip connections are incorporated into a simple encoder-decoder architecture, namely BiO-Net that reuses network parameters in a recurrent manner. We analyzed the complexity overhead of the plain BiO-Net and designed the economic multi-scale BiO-Net++ with dense skip connections. A two-phase NAS algorithm, BiX-NAS, was further proposed to automatically search for the optimal sub-set skip connections in BiO-Net++. At the first phase, we utilized a novel selection matrix to narrow down the search space. Our modification supports a flexible output of skips and unifies the forward behaviours during both searching and network inference. At the second phase, we progressively searched for the optimal skip connections between every pair of encoder and decoder. The finally searched BiX-Net yields only 1/39.4 parameters and 1/4.1 computation compared to the plain BiO-Net, but achieves even better segmentation performances on different benchmarks. We evaluated our methods on 3 2D datasets and 3 3D datasets that achieved state-of-the-art performances. Extensive ablation studies were conducted to provide more comprehensive analysis of each of the proposed methods and components.

References

Alom, M.Z., Yakopcic, C., Taha, T.M., Asari, V.K., 2018. Nuclei segmentation with recurrent residual convolutional neural networks based u-net (r2u-net), in: IEEE National Aerospace and Electronics Conference, IEEE. pp. 228–233.

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence 40, 834–848.

Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O., 2016. 3d u-net: learning dense volumetric segmentation from sparse annotation, in: International conference on medical image computing and computer-assisted intervention, Springer. pp. 424–432.

Graham, S., Vu, Q.D., Raza, S.E.A., Azam, A., Tsang, Y.W., Kwak, J.T., Rajpoot, N., 2019. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. Medical Image Analysis 58, 101563.

Guo, Q., Yu, Z., Wu, Y., Liang, D., Qin, H., Yan, J., 2019. Dynamic recursive neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5147–5156.

Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J., 2020. Single path one-shot neural architecture search with uniform sampling, in: European Conference on Computer Vision (ECCV), Springer. pp. 544–560.

Han, W., Chang, S., Liu, D., Yu, M., Witbrock, M., Huang, T.S., 2018. Image super-resolution via dual-state recurrent networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1654–1663.

Haralick, R.M., Shapiro, L.G., 1985. Image segmentation techniques. Computer vision, graphics, and image processing 29, 100–132.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

Hinton, G.E., Zemel, R.S., 1994. Autoencoders, minimum description length, and helmholtz free energy. Advances in neural information processing systems 6, 3–10.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning (ICML), pp. 448–456.

Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P.F., Kohl, S., Wasserthal, J., Koehler, G., Norajitra, T., Wirkert, S., et al., 2018. nnu-net: Self-adapting framework for u-net-based medical image segmentation. arXiv preprint arXiv:1809.10486 .

Jang, E., Gu, S., Poole, B., 2017. Categorical reparameterization with gumbel-softmax, in: International Conference on Learning Representations (ICLR).

Ji, Y., Zhang, R., Li, Z., Ren, J., Zhang, S., Luo, P., 2020. Uxnet: Searching multi-level feature aggregation for 3d medical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 346–356.

Kavur, A.E., Gezer, N.S., Barış, M., Aslan, S., Conze, P.H., Groza, V., Pham,

D.D., Chatterjee, S., Ernst, P., Özkan, S., et al., 2021. CHAOS challenge-combined (CT-MR) healthy abdominal organ segmentation. Medical Image Analysis 69, 101950.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: International Conference on Learning Representations (ICLR).

Kumar, N., Verma, R., Sharma, S., Bhargava, S., Vahadane, A., Sethi, A., 2017. A dataset and a technique for generalized nuclear segmentation for computational pathology. IEEE Transactions on Medical Imaging 36, 1550–1560.

Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A.L., Fei-Fei, L., 2019a. Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 82–92.

Liu, H., Simonyan, K., Yang, Y., 2019b. DARTS: Differentiable architecture search, in: International Conference on Learning Representations (ICLR).

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.

Ma, J., Wang, Y., An, X., Ge, C., Yu, Z., Chen, J., Zhu, Q., Dong, G., He, J., He, Z., et al., 2020. Towards efficient covid-19 ct annotation: A benchmark for lung and infection segmentation. arXiv preprint arXiv:2004.12537 .

Mehta, R., Sivaswamy, J., 2017. M-net: A convolutional neural network for deep brain structure segmentation, in: 14th International Symposium on Biomedical Imaging (ISBI), IEEE. pp. 437–440.

Milletari, F., Navab, N., Ahmadi, S.A., 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 4th International Conference on 3D Vision (3DV), IEEE. pp. 565–571.

Müller, D., Rey, I.S., Kramer, F., 2020. Automated chest ct image segmentation of covid-19 lung infection based on 3d u-net. arXiv preprint arXiv:2007.04774 .

Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on Machine Learning (ICML), pp. 807–814.

Naylor, P., Laé, M., Reyal, F., Walter, T., 2018. Segmentation of nuclei in histopathology images by deep regression of the distance map. IEEE Transactions on Medical Imaging 38, 448–459.

Oktay, O., Schlemper, J., Folgoc, L.L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N.Y., Kainz, B., et al., 2018. Attention u-net: Learning where to look for the pancreas. 1st Conference on Medical Imaging with Deep Learning (MIDL) .

Real, E., Aggarwal, A., Huang, Y., Le, Q.V., 2019. Regularized evolution for image classifier architecture search, in: Proceedings of the AAAI conference on artificial intelligence, pp. 4780–4789.

Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A., 2017. Large-scale evolution of image classifiers, in: International Conference on Machine Learning, PMLR. pp. 2902–2911.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer. pp. 234–241.

Salehi, S.S.M., Erdogmus, D., Gholipour, A., 2017. Tversky loss function for image segmentation using 3d fully convolutional deep networks, in: International workshop on machine learning in medical imaging, Springer. pp. 379–387.

Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., et al., 2019. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 .

Tomczak, K., Czerwińska, P., Wiznerowicz, M., 2015. The cancer genome atlas (tcga): an immeasurable source of knowledge. Contemporary oncology 19, A68.

Wang, W., Yu, K., Hugonot, J., Fua, P., Salzmann, M., 2019. Recurrent U-Net for resource-constrained segmentation, in: The IEEE International Conference on Computer Vision (ICCV).

Wang, X., Xiang, T., Zhang, C., Song, Y., Liu, D., Huang, H., Cai, W., 2021. Bix-nas: Searching efficient bi-directional architecture for medical image segmentation. arXiv:2106.14033.

Weng, Y., Zhou, T., Li, Y., Qiu, X., 2019. NAS-Unet: Neural architecture search for medical image segmentation. IEEE Access 7, 44247–44257.

Xia, X., Kulis, B., 2017. W-net: A deep model for fully unsupervised image segmentation. arXiv preprint arXiv:1711.08506 .

Xiang, T., Zhang, C., Liu, D., Song, Y., Huang, H., Cai, W., 2020. BiO-Net: Learning recurrent bi-directional connections for encoder-decoder architecture, in: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer. pp. 74–84.

Yan, X., Jiang, W., Shi, Y., Zhuo, C., 2020. MS-NAS: Multi-scale neural architecture search for medical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer. pp. 388–397.

Yang, Z., Wang, Y., Chen, X., Shi, B., Xu, C., Xu, C., Tian, Q., Xu, C., 2020. CARS: Continuous evolution for efficient neural architecture search, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1829–1838.

Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J., 2018. UNet++: A nested u-net architecture for medical image segmentation, in: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (DLMIA). Springer, pp. 3–11.

Zhu, Z., Liu, C., Yang, D., Yuille, A., Xu, D., 2019. V-nas: Neural architecture search for volumetric medical image segmentation, in: 2019 International Conference on 3D Vision (3DV), IEEE. pp. 240–248.

Zoph, B., Le, Q.V., 2016. Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 .