Reinventing 2D Convolutions for 3D Medical Images

Jiancheng Yang∗ Xiaoyang Huang∗ Bingbing Ni† Jingwei Xu Canqian Yang Guozheng Xu
{jekyll4168, 13802896410, nibingbing, xjwxjw, charles.young, guozhengxu}@sjtu.edu.cn
Shanghai Jiao Tong University

Abstract

There has been considerable debate over 2D and 3D representation learning on 3D medical images. 2D approaches could benefit from large-scale 2D pretraining, whereas they are generally weak in capturing large 3D contexts. 3D approaches are natively strong in 3D contexts, however few publicly available 3D medical datasets are large and diverse enough for universal 3D pretraining. Even for hybrid (2D + 3D) approaches, the intrinsic disadvantages within the 2D / 3D parts still exist. In this study, we bridge the gap between 2D and 3D convolutions by reinventing the 2D convolutions. We propose ACS (axial-coronal-sagittal) convolutions to perform natively 3D representation learning, while utilizing the pretrained weights from 2D counterparts. In ACS convolutions, 2D convolution kernels are split by channel into three parts, and convoluted separately on the three views (axial, coronal and sagittal) of 3D representations. Theoretically, ANY 2D CNN (ResNet, DenseNet, or DeepLab) is able to be converted into a 3D ACS CNN, with pretrained weights of same parameter sizes. Extensive experiments on proof-of-concept dataset and several medical benchmarks validate the consistent superiority of the pretrained ACS CNNs, over the 2D / 3D CNN counterparts with / without pretraining. Even without pretraining, the ACS convolution can be used as a plug-and-play replacement of standard 3D convolution, with smaller model size.1

1. Introduction

Emerging deep learning technology has been dominating the medical image analysis research [38, 52], in a wide range of data modalities (e.g., ultrasound [7, 17], CT [49, 63, 5], MRI [42, 15, 4], X-Ray [58, 30, 32]) and tasks (e.g., classification [20, 18], segmentation [31, 56], detection [62, 57], registration [2, 12]). Thanks to contributions from dedicated researchers from academia and industry, there have been much larger medical image datasets than even before. With large-scale datasets, strong infrastructures and powerful algorithms, numerous challenging problems in medical images seem solvable. However, the data-hungry nature of deep learning limits its applicability in various real-world scenarios with limited annotations. Compared to millions (or even billions) of annotations in natural image datasets, the medical image datasets are never too large. Especially for 3D medical images, datasets with thousands of supervised training annotations [51, 66, 54] are so-called “large”, due to several difficulties in medical annotations: hardly-accessible and high dimensional medical data, expensive expert annotators (radiologists / clinicians), and severe class-imbalance issues [61].

Transfer learning, with pretrained weights from large-scale datasets (e.g., ImageNet [13], MS-COCO [37]), is a

Pros | Cons
--- | ---
2D pretrained weights on large 2D datasets | Natively 2D representations
Natively 3D representations | Lack of 3D pretrained weights on large datasets
2D + 3D representations | a. 2D representation within 2D parts
b. Lack of 3D pretrained weights
c. Redundant multi-stage / multi-stream models

Figure 1: A comparison between the proposed ACS convolutions and prior art on modeling the 3D medical images: pure 2D / 2.5D approaches with 2D convolution kernels, pure 3D approaches with 3D convolution kernels, and hybrid approaches with both 2D and 3D convolution kernels. The ACS convolutions run multiple 2D convolution kernels among the three views (axial, coronal and sagittal).
de-facto paradigm for tasks with insufficient data. Unfortunately, widely-used pretrained CNNs are developed on 2D datasets, which are non-trivial to transfer to 3D medical images. Prior art on 3D medical images follows either 2D-based approaches or 3D-based approaches (compared in Fig. 1). 2D-based approaches [50, 65, 44] benefit from large-scale pretraining on 2D natural images, while the 2D representation learning are fundamentally weak in large 3D contexts. 3D-based approaches [11, 43, 68] learn natively 3D representations. However, few publicly available 3D medical dataset is large and diverse enough for universal 3D pretraining. Therefore, compact network design and sufficient training data are essential for training the 3D networks from scratch. Hybrid (2D + 3D) approaches [35, 59, 69] seem to get the best of both worlds, nevertheless these ensemble-like approaches do not fundamentally overcome the intrinsic issues of 2D-based and 3D-based approaches. Please refer to Sec. 2 for in-depth discussion on these related methods.

There has been considerable debate over 2D and 3D representation learning on 3D medical images: prior studies choose either large-scale 2D pretraining or natively 3D representation learning. This paper presents an alternative to bridge the gap between the 2D and 3D approaches. To solve the intrinsic disadvantages from the 2D convolutions and 3D convolutions in modeling 3D images, we argue that an ideal method should adhere to the following principles:

1) **Natively 3D representation**: it learns natively 3D representations for 3D medical images;
2) **2D weight transferable**: it benefits from the large-scale pretraining on the 2D images [13, 37, 58, 30, 32];
3) **ANY model convertible**: it enables any 2D model, including classification [24], detection [36] and segmentation [8] backbones, to be converted to a 3D model.

These principles cannot be achieved simultaneously with standard 2D convolutions or standard 3D convolutions, which directs us to develop a novel convolution operator. Inspired from the widely-used tri-planar representations of 3D medical images [50], we propose ACS convolutions satisfying these three principles. Instead of explicitly treating the input 3D volumes as three orthogonal 2D planar images [50] (axial, coronal and sagittal), we operate on the convolution kernels to perform view-based 3D convolutions, via splitting the 2D convolution kernels into three parts by channel. Notably, **no additional 3D fusion layer** is required to fuse the three-view representations from the 3D convolutions, since they will be seamlessly fused by the subsequent ACS convolution layers (see details in Sec. 3).

The ACS convolution aims at a generic and plug-and-play replacement of standard 3D convolutions for 3D medical images. Our experiments empirically prove that, even without pretraining, the ACS convolution is comparable to 3D convolution with a smaller model size. When pretrained on large 2D datasets, it consistently outperforms 2D / 3D convolution by a large margin. To improve research reproducibility, a PyTorch [45] reference implementation of ACS convolution is provided in the supplementary materials. Using the provided function, 2D CNNs could be converted into ACS CNNs for 3D images, with a single line of code.\(^2\)

2. **Related Work on 3D Medical Images**

2.1. **2D / 2.5D Approaches**

Transfer learning from 2D CNNs, trained on large-scale datasets (e.g., ImageNet [13]), is a widely-used approach in 3D medical image analysis. To mimic the 3-channel image representation (i.e., RGB), prior studies follow either multi-planar or multi-slice representation of 3D images as 2D inputs. In these studies, pretrained 2D CNNs are usually fine-tuned on the target medical dataset.

Early study [50] proposes tri-planar representation of 3D medical images, where three views (axial, coronal and sagittal) from a voxel are regarded as the three channels of 2D input. Although this method is empirically effective, there is a fundamental flaw that the channels are not spatially aligned. More studies follow tri-slice representations [14, 65, 44], where a center slice together with its two neighbor slices are treated as the three channels. In these representations, the channels are spatially aligned, which comforms to the inductive biases in convolution. There are also studies [65, 46] combining both multi-slice and multi-planar approaches, using multi-slice 2D representations in multiple views. The multi-view representations are generally averaged [65] or fused by additional networks [46].

Even though these approaches benefit from large-scale 2D pretraining, which is empirically effective in numerous studies [41, 18, 36, 8], both multi-slice and multi-planar representation with 2D convolutions are fundamentally weak in capturing large 3D contexts.

2.2. **3D Approaches**

Instead of regarding the 3D spatial information as input channels in 2D approaches, there are numbers of studies using pure 3D convolutions for 3D medical image analysis [11, 43, 33, 16, 68, 67, 64]. Compared to limited 3D contexts along certain axis in 2D approaches, the 3D approaches are theoretically capable of capturing arbitrarily large 3D contexts in any axis. Therefore, the 3D approaches are generally better at tasks requiring large 3D contexts, *e.g.*, distinguishing small organs, vessels, and lesions.

However, there are also drawbacks for pure 3D approaches. One of the most important is the lack of large-scale universal 3D pretraining. For this reason, efficient training of 3D networks is a pain point for 3D approaches. Several techniques are introduced to (partially) solve this

\(^2\)Code is available at [http://github.com/m3dv/ACSCConv](http://github.com/m3dv/ACSCConv).
issue, e.g., deep supervision [16], compact network design [70, 68]. Nevertheless, these techniques are not directly targeting the issue of 3D pretraining.

### 2.3. Hybrid Approaches

Hybrid approaches are proposed to combine the advantages of both 2D and 3D approaches [35, 59, 69, 46]. In these studies, 2D pretrained networks with multi-slice inputs, and 3D randomly-initialized networks with volumetric inputs are (jointly or separately) trained for the target tasks. The hybrid approaches could be mainly categorized into multi-stream and multi-stage approaches. In multi-stream approaches [35, 69], 2D networks and 3D networks are designed to perform a same task (e.g., segmentation) in parallel. In multi-stage (i.e., cascade) approaches [59, 69, 46], several 2D networks (and 3D networks) are developed to extract representations from multiple views, and a 3D fusion network is then used to fuse the multi-view representations into 3D representations to perform the target tasks.

Although empirically effective, the hybrid approaches do not solve the intrinsic disadvantages of 2D and 3D approaches: the 2D parts are still not able to capture large 3D contexts, and the 3D parts still lacks large-scale pretraining. Besides, these ensemble-like methods are generally redundant to deploy in practice.

### 2.4. Transfer Learning & Self-Supervised Learning

Medical annotations require expertise in medicine and radiology, which are thereby expensive to be scalable. For certain rare diseases or novel applications (e.g., predicting response for novel treatment [55]), the data scale is naturally very small. Transfer learning from large-scale datasets to small-scale datasets is a de-facto paradigm in this case.

Human without any radiological experience could recognize basic anatomy and lesions on 2D and 3D images with limited demonstration. Based on this observation, we believe that transfer learning from universal vision datasets (e.g., ImageNet [13], MS-COCO [37]) should be beneficial for 3D medical image analysis. Although there is literature reporting that universal pretraining is useless for target tasks [23, 47], this phenomenon is usually observed when target datasets are large enough. Apart from boosting target task performance, the universal pretraining is able to improve model robustness and uncertainty quantification [26, 28].

Unfortunately, 2D-to-3D transfer learning has not been adequately studied. Research efforts [33, 19] have been paid to pretrain natively 3D CNNs on 3D datasets, however few publicly available 3D medical dataset is large and diverse enough for universal pretraining. Prior research explores the transfer learning of 3D CNNs trained on spatio-temporal video datasets [29]. However, there are two kinds of domain shift between video and 3D medical images: 1) natural images vs. medical images, and 2) spatio-temporal data vs. 3D spatial data. The domain shift makes video pretraining [22] less applicable for 3D medical images. To reduce domain shift, there is research (Med3D [9]) building pretrained 3D models on numbers of 3D medical image datasets. However, these techniques are not directly targeting the issue of 3D pretraining.

Table 1 compares the sources of transfer learning for 3D medical images. Compared to transfer learning from video [22] / Med3D [9] / Models Genesis [71], the key advantage of 2D image pretraining is the overwhelming data scale and diversity of datasets. With the ACS convolutions proposed in this study, we are able to develop natively 3D CNNs using 2D pretrained weights. We compare these pretraining approaches in our experiments, and empirically prove the superiority of the proposed ACS convolutions.

Note that the contribution of the ACS convolutions is orthogonal to pretraining data. It is possible to pretrain ACS CNNs on 2D images, videos and 3D medical images with supervised / self-supervised learning. The paper uses ACS convolution with supervised pretraining on 2D natural images to demonstrate its effectiveness, flexibility and versatility.

### 3. ACS Convolutional Neural Networks

We introduce the ACS (axial-coronal-sagittal) convolutions, how to convert a 2D CNN to an ACS CNN, and the counterparts and variants of the proposed method.

#### 3.1. ACS Convolutions

Convolution layers capture spatial correlation. Intuitively, the formal difference between 2D and 3D convolutions is the kernel size: the 2D convolutions use...
2D kernels \((C_o \times C_i \times K \times K)\) for 2D inputs \((C_i \times H_i \times W_i)\), whereas the 3D convolutions use 3D kernels \((C_o \times C_i \times K \times K)\) for 3D inputs \((C_i \times D_i \times H_i \times W_i)\), where \(C_i, C_o\) denote the channels of inputs and outputs, \(K\) denotes the kernel size, and \((D_i \times H_i \times W_i)\) denotes the input size. To transfer the 2D kernels to 3D kernels, there are basically two prior approaches: 1) “inflate” the pretrained 2D kernels into 3D kernels size \((K \times K \rightarrow K \times K \times K)\), i.e., Inflated 3D (I3D) [6], where the 2D kernels are repeated along an axis and then normalized; 2) unsqueeze the 2D kernels into pseudo 3D kernels on an axis \((K \times K \rightarrow 1 \times K \times K)\), i.e., AH-Net-like [40], which could not effectively capture 3D contexts. Note that in both cases, the existing methods assume a specific axis to transfer the 2D kernels. It is meaningful to assign a special axis for spatio-temporal videos, while controversial for 3D medical images. Even for anisotropic medical images, any view of the 3D image is still a 2D spatial image.

Based on this observation, we develop ACS convolutions to learn spatial representations from the axial, coronal and sagittal views. Instead of treating channels of 2D kernels equally [6, 40], we split the kernels into three parts for extracting 3D spatial information from the axial, coronal and sagittal views. The detailed calculation of ACS convolutions are shown in Algorithm 1. For simplicity, we introduce the ACS convolutions with same padding (Fig. 2).

Given a 3D input \(X_i \in \mathbb{R}^{C_i \times D_i \times H_i \times W_i}\), we would like to obtain a 3D output \(X_o \in \mathbb{R}^{C_o \times D_o \times H_o \times W_o}\) with pretrained / non-pretrained 2D kernels \(W \in \mathbb{R}^{C_o \times C_i \times K \times K}\). Here, \(C_i\) and \(C_o\) denote the input and output channels, \(D_i \times H_i \times W_i\) and \(D_o \times H_o \times W_o\) denote the input and output sizes, \(K\) denotes the kernel size. Instead of presenting 3D images into tri-planar 2D images [50], we split and reshape the kernels into three parts (named ACS kernels) by the output channel, to obtain the view-based 3D representations for each volume: \(W_a \in \mathbb{R}^{C_o(a) \times C_i \times K \times K \times 1}\), \(W_c \in \mathbb{R}^{C_o(c) \times C_i \times K \times 1}\), \(W_s \in \mathbb{R}^{C_o(s) \times C_i \times 1 \times K}\), where \(C_a^{(c)} + C_c^{(c)} + C_s^{(c)} = C_o\). It is theoretically possible to assign an “optimal” axis for a 2D kernel; However, considering the feature redundancy in CNNs [21], in practice we simply set \(C_a^{(c)} \approx C_c^{(c)} \approx C_s^{(c)} \approx |C_o/3|\). We then compute the view-based 3D features from axial, coronal and sagittal views via 3D convolutions:

\[
X_o^{(a)} = \text{Conv3D}(X_i, W_a) \in \mathbb{R}^{C_o(a) \times D_o \times H_o \times W_o}, \tag{1}
\]
CNNs.

Table 2: Main operator conversion from 2D CNNs into ACS CNNs. $K \times K$, $1 \times 1$ and $1 \times 1 \times 1$ denote the kernel sizes.

| 2D CNNs | ACS CNNs |
|---------|----------|
| Conv2D $K \times K$ | ACSConv $K \times K$ |
| Conv2D $1 \times 1$ | Conv3D $1 \times 1 \times 1$ |
| \{Batch,Group\}Norm2D | \{Batch,Group\}Norm3D |
| \{Max,Avg\}Pool2D | \{Max,Avg\}Pool3D |

Table 3: Theoretical analysis of space and time complexity, for 2D (2.5D), 3D, ACS, Mean-ACS, and Soft-ACS convolutions. Bias terms are not counted in parameter size.

| Kernels | FLOPs | Memory | Parameters |
|---------|-------|--------|------------|
| Conv2D | $O(DHWcC_3K^2)$ | $DHWcO$ | $C_3OcK^2$ |
| Conv3D | $O(DHWcC_3K^3)$ | $DHWcO$ | $C_3OcK^3$ |
| ACSConv | $O(DHWcC_2K^2)$ | $DHWcO$ | $C_2OcK^2$ |
| M-ACSConv | $O(3DHWcC_2K^2)$ | $3DHWcO$ | $C_2OcK^2$ |
| S-ACSConv | $O(3DHWcC_2K^2)$ | $3DHWcO$ | $C_2O(cK^2+3)$ |

3.3. ACS Convolution Variants

Apart from the kernel splitting approach used in the proposed ACS convolutions, there are several possible variants to implement the 2D-transferable, ACS-like convolutions. 

**Mean-ACS convolutions.** Instead of splitting the 2D convolution kernels, we replicate and reshape $W$ into $\mathbf{W}^o \in \mathbb{R}^{C_0 \times C_2 \times K \times K \times 1}$, $\mathbf{W}^c \in \mathbb{R}^{C_0 \times C_2 \times K \times 1 \times K}$, and obtain the 3D feature maps by $X^o(c) = \text{Conv3D}(X_i, \mathbf{W}_c)$, $X^o(c) = \text{Conv3D}(X_i, \mathbf{W}_c)$, $X^o(s) = \text{Conv3D}(X_i, \mathbf{W}_s)$. The output features are

$$X^M_o = X^o(a) + X^o(c) + X^o(s).$$

**Soft-ACS convolutions.** Note that the Mean-ACS convolution uses a symmetric aggregation, thereby it could not distinguish any view-based information. To this regard, we introduce weighted sum of Mean-ACS, i.e., Soft-ACS,

$$X^S_o = \alpha^a X^o(a) + \alpha^c X^o(c) + \alpha^s X^o(s),$$

where $\alpha^a, \alpha^c, \alpha^s \in \mathbb{R}$ are learnable weights.

In Table 3, we compare the time and space complexity. The two variants are more computationally intensive in terms of FLOPs and memory. Unfortunately, they do not provide significant performance boost empirically. Therefore, we only report the model performance of ACS convolutions in Sec. 4, and analyze these variants in Sec. 5.1.

4. Experiments

We experiment with the proposed method on a proof-of-concept dataset and medical benchmarks. To fairly compare model performance, we include several counterparts (2.5D/3D/ACS {Network} r/p) with same experiment setting, where r denotes random initialization, and p denotes pretraining on various sources. We use separate network architectures in different experiments to demonstrate the flexibility and versatility of the proposed method.

4.1. Proof-of-Concept

**Dataset.** We first validate our method on a proof-of-concept dataset to perform semantic segmentation task. As
Table 4: Segmentation performance of 2.5D, 3D and ACS convolution models w/ and w/o pretraining on the proof-of-concept dataset. r. denotes randomly initialized. The 2.5D and ACS UNet p. are pretrained on synthetic 2D images.

| Models        | Dice   | mIoU   | Model Size |
|---------------|--------|--------|------------|
| 2.5D UNet r.  | 82.24  | 72.48  | 1.6 Mb     |
| 2.5D UNet p.  | 82.71  | 73.28  | 1.6 Mb     |
| 3D UNet r.    | 94.63  | 90.78  | 4.7 Mb     |
| ACS UNet r.   | 94.68  | 90.71  | 1.6 Mb     |
| ACS UNet p.   | 95.44  | 91.99  | 1.6 Mb     |

Experiment Setting. We compare our ACS model with 2.5D and 3D counterparts (Sec. 3.2) under random initialization or pretraining setting. All models share a same UNet [48, 11] architecture with downsampling twice, except for the convolution modules. Dice loss is used for training both 2D and 3D UNet. We first train a 2D UNet on 2D dataset until convergence, which reaches a Dice of 98.57 on 2D dataset. Its weights could be used to transfer to 3D models with ACS convolutions. We report the Dice and mIoU averaged on the 5 classes of 3D dataset.

Result Analysis. As shown in Table 4, the performance of ACS UNet w/o pretraining is comparable to that of 3D UNet w/o pretraining, and the ACS UNet with pretraining achieves the best performance. The results indicate that ACS Convolution is an alternative to 3D Convolution with comparable or even better performance, and smaller model size. ACS convolution, as a compact 3D convolution op-
initial learning rate of 0.001 and train the model for 100 epochs, and delay the learning rate by 0.1 after 50 and 75 epochs. For ResNet-18 backbone, in order to keep higher resolution for output feature maps, we modify the stride of first layer (7 × 7 stride-2 convolution) into 1, and remove the first max-pooling. Note that this modification still enables pretraining. A FCN-like [41] decoder is applied with progressive upsampling twice. Dice loss with a batch of 8 is used for segmentation, and binary cross-entropy with a batch of 24 for classification. Dice global and AUC are reported for these two tasks. To demonstrate the flexibility and versatility of ACS convolutions, we also report the results of VGG [53] and DenseNet [27] with similar experiment setting in the supplementary materials, which is consistent with the ResNet-18 performance.

**Result Analysis.** Experiment results are depicted in Table 5. The ACS models consistently outperform all the counterparts by a large margin, including 2.5D and 3D models in both random initialization or pretraining setting. We observe that the 3D models (both ACS and 3D) generally outperform the 2.5 models, indicating that the usefulness of 3D contexts in 3D medical image modeling. Except for the pretrained 2.5D model on classification task, its superior performance over 3D counterparts may explain the prior art [60, 39] with 2D networks on this dataset. As for pretraining, the ImageNet [13] provides significant performance boost (see 2.5D p., 3D p. I3D [6] and ACS p.), while Med3D [9] brings limited performance boost. We conjecture that it is owing to the overwhelming data scale and diversity of 2D image dataset. We provide visualization of the lung nodule segmentation for qualitative evaluation in the supplementary materials. Moreover, to investigate the training speed of ACS vs. 3D convolutions, we plot the training curve on the two tasks in Fig. 4. It is observed that ACS p. converges the fastest and best in the 4 models.

Due to the difference on network architecture (ResNet-based FCN vs. UNet), we experiment with the official code of self-supervised pretrained Models Genesis [71] with exactly same setting. Even without pretraining, the segmentation and classification performance of the UNet-based models are strong on this dataset. Despite this, the pretrained ACS model is still better performing. Besides, negative transferring is observed for classification by the MG [71] encoder-only transferring, whereas the ImageNet pretraining consistently improves the model performance.

Apart from the superior model performance, the ACS model achieves the best parameter efficiency in our experiments. Take the segmentation task for example, the size of ACS model is 49.8 Mb, compared to 49.8 Mb (2.5D), 142.5 Mb (3D) and 65.4 Mb (MG [71]).

| Models                        | Segmentation | Classification |
|-------------------------------|--------------|----------------|
| Models Genesis [71] p.        | 75.9         | 94.1           |
| Models Genesis [71] r.        | 75.5         | 94.3           |
| 2.5D Res-18 r.                | 68.8         | 89.4           |
| 2.5D Res-18 p.                | 69.8         | 92.0           |
| 3D Res-18 r.                  | 74.7         | 90.3           |
| 3D Res-18 p. I3D [6]          | 75.7         | 91.5           |
| 3D Res-18 p. Med3D [9]        | 74.9         | 90.6           |
| 3D Res-18 p. Video [22]       | 75.7         | 91.0           |
| ACS Res-18 r.                 | 75.1         | 92.5           |
| ACS Res-18 p.                 | **76.5**     | **94.9**       |

Table 5: LIDC lung nodule segmentation (Dice global) and classification (AUC) performance. The 2.5D, I3D and ACS ResNet-18 p. are pretrained on ImageNet [13].

**Figure 4:** 3D vs. ACS r. / p. training curves of segmentation and classification on LIDC-IDRI dataset. The curves are smoothed with moving average for better visualization.

### 4.3. Liver Tumor Segmentation (LiTS) Benchmark

**Dataset.** We further experiment with our approach on LiTS [5], a challenging 3D medical image segmentation dataset. It consists of 131 and 70 enhanced abdominal CT scans for training and testing respectively, to segment the liver and liver tumors. The training annotations are open to public while the test ones are only accessible by online evaluation. The sizes of x, y axis are 512, while the sizes of z axis are various in the range of [50, 1000]. We transpose the axes into z, y, x to keep the concept consistent as previously mentioned. For pre-processing, we clip the Hounsfield Unit to [−200, 250] and then normalize to [0, 1], without spatial normalization. Training data augmentation includes random-center cropping, random-axis flipping and rotation, and random-scale resampling.

**Experiment Setting.** A DeepLabv3+ [8] with a backbone of ResNet-101 [24] is used in this experiment. The pretrained 2D model is directly obtained from PyTorch’s torchvision package [45]. The compared baselines are similar to those in the above LIDC experiment (Sec. 4.2). We train all the models for 500 epochs. An Adam opti-
mizer [34] is used with an initial learning rate of 0.001, and we decay the learning rate by 0.1 after 250 and 375 epochs. At training stage, we crop the volumes to the size of $64 \times 224 \times 224$. As for testing stage, we crop the volumes to the size of $64 \times 512 \times 512$ and adopt window sliding at a step of 24 at $z$ axis. Dice global and Dice per case of lesion and liver are reported as standard evaluation on this dataset.

### Result Analysis

As shown in Table 6, similar model behavior to LIDC experiment (Sec. 4.2) can be observed. The pretrained ACS DeepLab achieves better performance than the 2D and 3D counterparts (including self-supervised pretraining [71]) by a large margin; without pretraining, ACS DeepLab achieves comparable or better performance than 3D DeepLab. According to pretraining results on I3D [6], Med3D [9] and Video [22] for 3D DeepLab, negative transferring is observed, probably due to severe domain shift and anisotropy on LiTS dataset. We also report a *state-of-the-art* performance on LiTS dataset using H-DenseUNet [35] as a reference. Note that it adopts a completely different training strategy and network architecture (a heavy cascade of 2D and 3D DenseNet-based [27] models), thereby it is not suitable to compare to other models directly. In further study, it is feasible to integrate these orthogonal contributions into our models to improve the model performance.

| Models          | Lesion DG | DPC | Liver DG | DPC |
|-----------------|-----------|-----|----------|-----|
| H-DenseUNet [35]| 82.4      | 72.2| 96.5     | 96.1|
| Models Genesis [71] | -   | -    | -       | 91.13|
| 2.5D DeepLab r. | 72.6      | 56.7| 92.1     | 91.7|
| 2.5D DeepLab p. | 73.3      | 59.8| 91.9     | 91.0|
| 3D DeepLab r.  | 75.3      | 62.2| 93.8     | 93.8|
| 3D DeepLab p. I3D [6] | 76.4 | 57.7| 93.1     | 92.4|
| 3D DeepLab p. Med3D [9] | 66.8 | 53.9| 91.0     | 92.6|
| 3D DeepLab p. Video [22] | 65.2 | 55.8| 91.5     | 92.2|
| ACS DeepLab r. | 75.2      | 62.1| 94.0     | 93.9|
| ACS DeepLab p. | 78.0      | 65.3| 94.8     | 94.8|

Table 6: LiTS segmentation performance. DG: Dice global. DPC: Dice per case. “DeepLab” denotes 3D / ACS ResNet-101 followed by 3D / ACS ASPP block [8]. The 2.5D, I3D and ACS DeepLab p. are pretrained on MS-COCO [37].

ACS DeepLab is used with an initial learning rate of 0.001, and we decay the learning rate by 0.1 after 250 and 375 epochs. At training stage, we crop the volumes to the size of $64 \times 224 \times 224$. As for testing stage, we crop the volumes to the size of $64 \times 512 \times 512$ and adopt window sliding at a step of 24 at $z$ axis. Dice global and Dice per case of lesion and liver are reported as standard evaluation on this dataset.

### 5. Ablation Study

#### 5.1. Analysis of ACS Convolution Variants

We analyze the variants of ACS Convolutions, including Mean-ACS Convolutions and Soft-ACS Convolutions. We test these three methods on LIDC-IDRI dataset, using the same experiment settings and training strategy specified in Sec. 4.2. As depicted in Table 7, the vanilla ACS outperforms its variants in most situations, and pretraining is useful in all cases. Specifically, Mean-ACS is the worst under pretraining setting, due to its inability to distinguish the view-based difference with a symmetric aggregation. Soft-ACS outperforms others in some cases (e.g., classification with pretraining), though it consumes more GPU memory and time at the training stage. However, it demonstrates the potential to combine these ACS variants or training strategy (e.g., automatic kernel axis assignment) in further study.

#### 5.2. Whole-Network vs. Encoder-Only Pretraining

A key advantage of the proposed ACS convolution is that it enables flexible whole-network conversion together with the pretrained weights. We thereby validate the superiority of whole-network weight transferring (WN) over encoder-only weight transferring (EO). We train 4 models in different pretraining setting: entirely randomly-initialized (ACS r.), only the pretrained ResNet-101 backbone (ACS p.EO) on ImageNet (IMN) [13] and MS-COCO (MSC) [37], and whole pretrained model (ACS p.WN). The results are shown in Table 8. It is observed that with more pretrained weights loaded, the model achieves better performance ($p.WN>p.EO>r.$), and the whole-network pretraining achieves the best. Note that although methods like I3D [6], Med3D [9] and Video [22] provide natively 3D pretrained models, apart from the underperforming performance, these pretraining methods are less flexible and

| Seg | Cls | Memory (Gb) | Time (s) |
|-----|-----|-------------|----------|
| ACS r. | 75.1 | 92.5 | 6.6 | 0.95 |
| M-ACS r. | 74.4 | 89.9 | 7.8 | 1.49 |
| S-ACS r. | 75.0 | 89.3 | 9.9 | 1.58 |
| ACS p. | 76.5 | 94.9 | 6.6 | 0.95 |
| M-ACS p. | 75.1 | 92.7 | 7.8 | 1.49 |
| S-ACS p. | 75.9 | 95.1 | 9.9 | 1.58 |

Table 7: A comparison of ACS convolutions and the Mean-ACS and Soft-ACS variants, with / without pretraining, in terms of LIDC segmentation Dice, classification AUC, actual memory and runtime speed per iteration. Memory and time is measured with a batch size of 2, on a single Titan Xp GPU without gradient checkpointing [10]. The memory consuming differs from the theoretical analysis (Table 3) due to PyTorch internal implementation.
state-of-the-art (backbones) are transferred in previous pretraining methods, more versatile than our method. Generally, only the encoders out of the whole models are also depicted.

network pretraining). The model sizes of pretrained weights p.

training on ImageNet [13]), "p.

ACS 

Table 8: LiTS segmentation performance of ACS DeepLab “r." (initialized randomly), “p.EO-IMN” (encoder-only pretraining on MS-COCO [37]), and “p.EO-MSC” (encoder-only pretraining on MS-COCO [37]), “p.WN” (whole-network pretraining). The model sizes of pretrained weights out of the whole models are also depicted.

| Models      | Size of Pretrained Weights | Lesion Liver |
|-------------|----------------------------|--------------|
| ACS r.      | 0 Mb (0%)                  | DG 62.1      |       |
| ACS p.EO-IMN| 170.0 Mb (72.5%)           | DG 64.3      |       |
| ACS p.EO-MSC| 170.0 Mb (72.5%)           | DG 61.6      |       |
| ACS p.WN    | 234.5 Mb (100%)            | DG 65.3      |       |

6. Conclusion

We propose ACS convolution for 3D medical images, as a generic and plug-and-play replacement of standard 3D convolution. It enables pretraining from 2D images, which consistently provides significant performance boost in our experiments. Even without pretraining, the ACS convolution is comparable or even better than 3D convolution, with smaller model size. In further study, we will focus on automatic ACS kernel axis assignment.

References

[1] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binseng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. The lung image database consortium (lidi) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans. Medical physics, 38(2):915–931, 2011. 6

[2] Guha Balakrishnan, Amy Zhao, Mert R Sabuncu, John Guttag, and Adrian V Dalca. Voxelmorp: a learning framework for deformable medical image registration. IEEE transactions on medical imaging, 2019. 1

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. arXiv preprint arXiv:1905.02249, 2019. 3

[4] Nicholas Bien, Pranav Rajpurkar, Robyn L Ball, Jeremy Irvin, Allison Park, Erik Jones, Michael Bereket, Bhavik N Patel, Kristen W Yeom, Katie Shpanskyat, et al. Deep-learning-assisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of mmnet. PLoS medicine, 15(11):e1002699, 2018. 1

[5] Patrick Bilic, Patrick Ferdinand Christ, Eugene Vorontsov, Grzegorz Chebus, Hao Chen, Qi Dou, Chi-Wing Fu, Xiao Han, Pheng-Ann Heng, Jürgen Hesser, et al. The liver tumor segmentation benchmark (llts). arXiv preprint arXiv:1901.04056, 2019. 1, 7

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In CVPR, pages 6299–6308, 2017. 4, 5, 6, 7, 8, 13

[7] Hao Chen, Yefeng Zheng, Jin-Hyeong Park, Pheng-Ann Heng, and S Kevin Zhou. Iterative multi-domain regularized deep learning for anatomical structure detection and segmentation from ultrasound images. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 487–495. Springer, 2016. 1

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In ECCV, pages 801–818, 2018. 2, 4, 7, 8, 9

[9] Sihong Chen, Kai Ma, and Yefeng Zheng. Med3d: Transfer learning for 3d medical image analysis. arXiv preprint arXiv:1904.00625, 2019. 3, 5, 6, 7, 8

[10] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. arXiv preprint arXiv:1604.06174, 2016. 8

[11] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In MICCAI, pages 424–432. Springer, 2016. 2, 5, 6

[12] Adrian V Dalca, Guha Balakrishnan, John Guttag, and Mert R Sabuncu. Unsupervised learning for fast probabilistic diffeomorphic registration. In MICCAI, pages 729–738. Springer, 2018. 1

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In CVPR, pages 248–255. Ieee, 2009. 1, 2, 3, 6, 7, 8, 9, 13

[14] Jia Ding, Aoxue Li, Zhiqiang Hu, and Liwei Wang. Accurate pulmonary nodule detection in computed tomography images using deep convolutional neural networks. In MICCAI, 2017. 2

[15] Qi Dou, Hao Chen, Lequan Yu, Lei Zhao, Jing Qin, Defeng Wang, Vincent CT Mok, Lin Shi, and Pheng-Ann Heng. Automatic detection of cerebral microbleeds from mr images via 3d convolutional neural networks. IEEE transactions on medical imaging, 35(5):1182–1195, 2016. 1

[16] Qi Dou, Lequan Yu, Hao Chen, Yueming Jin, Xing Yang, Jing Qin, and Pheng-Ann Heng. 3d deeply supervised network for automated segmentation of volumetric medical images. Medical image analysis, 41:40–54, 2017. 2

[17] Richard Droste, Yifan Cai, Harshita Sharma, Pierre Chateauain, Lior Drukker, Aris T Papageorgiou, and J Alison Noble. Ultrasound image representation learning by modeling sonographer visual attention. In International Conference on Information Processing in Medical Imaging, pages 592–604. Springer, 2019. 1

[18] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun.
Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017. 1, 2

[19] Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun I Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, et al. Niftynet: a deep-learning platform for medical imaging. *Computer methods and programs in biomedicine*, 158:113–122, 2018. 3

[20] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016. 1

[21] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 4

[22] Kensho Harai, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 3, 5, 6, 7, 8

[23] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *CVPR*, pages 4918–4927, 2019. 3

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 4, 5, 6, 7, 13

[25] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, Diederik P Kingma, Payam Fawzi, Alexei A Efros, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 3

[26] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 2712–2721, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 3

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 4, 7, 8, 13

[28] Xiaoyang Huang, Jiancheng Yang, Linguo Li, Haoran Deng, Bingbing Ni, and Yi Xu. Evaluating and boosting uncertainty quantification in classification. *arXiv preprint arXiv:1909.06030*, 2019. 3

[29] Sarfaraz Hussein, Kunlin Cao, Qi Song, and Ulas Bagci. Risk stratification of lung nodules using 3d cnn-based multi-task learning. In *International conference on information processing in medical imaging*, pages 249–260. Springer, 2017. 3

[30] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Icuc, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *AAAI*, 2019. 1, 2

[31] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*, 2018. 1

[32] Alistair E W Johnson, Tom J Pollard, Seth Berkowitz, Nathaniel R Greenbaum, Matthew P Langren, Chih-ying Deng, Roger G Mark, and Steven Horng. Mimic-cxr: A large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*, 2019. 1, 2

[33] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multiscale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical image analysis*, 36:61–78, 2017. 2, 3

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6, 7

[35] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi-Wing Fu, and Pheng-Ann Heng. H-denseunet: hybrid densely connected u-net for liver and tumor segmentation from ct volumes. *IEEE transactions on medical imaging*, 37(12):2663–2674, 2018. 2, 3, 8

[36] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 2

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1, 2, 3, 8, 9

[38] Geert Litjens, Thijs Kooi, Babak Eshghian Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohnsen Ghafoorian, Jeroen AW Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017. 1

[39] Li Hao Liu, Qi Dou, Hao Chen, Jing Qin, and Pheng-Ann Heng. Multi-task deep model with margin ranking loss for lung nodule analysis. *IEEE transactions on medical imaging*, 2019. 7

[40] Siqi Liu, Daguang Xu, S Kevin Zhou, Olivier Pauly, Sasa Grbic, Thomas Mertelmeier, Julia Wicklein, Anna Jerzbek, Weidong Cai, and Dorin Comaniciu. 3d anisotropic hybrid network: Transferring convolutional features from 2d images to 3d anisotropic volumes. In *MICCAI*, pages 851–858. Springer, 2018. 4, 5

[41] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, pages 3431–3440, 2015. 2, 7

[42] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kelpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014. 1

[43] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *JDV*, pages 565–571. IEEE, 2016. 2, 5

[44] Tianwei Ni, Lingxi Xie, Huangjie Zheng, Elliot K Fishman, and Alan L Yuille. Elastic boundary projection for 3d medi-
Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Mathias Perslev, Erik Bjørnager Dam, Akshay Pai, and Christian Igel. One network to segment them all: A general, lightweight system for accurate 3d medical image segmentation. In *MICCAI*, pages 30–38. Springer, 2019.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfer learning with applications to medical imaging. In *NeurIPS*, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

Holger R. Roth, Le Lu, Amal Farag, Hoo-Chang Shin, Jianmin Liu, Evrim B Türkbey, and Ronald M Summers. Deeppogan: Multi-level deep convolutional networks for automated pancreas segmentation. In *MICCAI*, pages 556–564. Springer, 2015.

Holger R. Roth, Le Lu, Ari Seff, Kevin M. Cherry, Joanne Hoffman, Shijun Wang, Jianmin Liu, Evrim B Türkbey, and Ronald M. Summers. A new 2.5d representation for lymph node detection using random sets of deep convolutional neural network observations. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, *MICCAI*, pages 520–527, Cham, 2014. Springer International Publishing.

Arnaud Arindra Adiyoso Setio, Alberto Traverso, Thomas De Bel, Moira SN Berens, Cas van den Bogaard, Piergiorgio Cerello, Hao Chen, Qi Dou, Maria Evelina Fantacci, Bram Geurts, et al. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the lunet challenge. *Medical image analysis*, 42:1–13, 2017.

Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Amber L Simpson, Michela Antonelli, Spyridon Bakas, Michel Bilello, Keyvan Farahani, Bram van Ginneken, Annette Kopp-Schneider, Bennett A Landman, Geert Litjens, Bjoern Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019.

Roger Sun, Elaine Johanna Limkin, Maria Vakalopoulou, Laurent Derche, Stéphane Champiat, Shan Rong Han, Loïc Verlingue, David Brandao, Andrea Lancia, Sammy Ammari, et al. A radiomics approach to assess tumour-infiltrating c8d cells and response to anti-pd-1 or anti-pd-11 immunotherapy: an imaging biomarker, retrospective multicohort study. *The Lancet Oncology*, 19(9):1180–1191, 2018.

Hao Tang, Xuming Chen, Yang Liu, Zhipeng Lu, Junhua You, Mingzhou Yang, Shengyu Yao, Guoqi Zhao, Yi Xu, Tingfeng Chen, et al. Clinically applicable deep learning framework for organs at risk delineation in ct images. *Nature Machine Intelligence*, pages 1–12, 2019.

Hao Tang, Chupeng Zhang, and Xiaohui Xie. Nodulenet: Decoupled false positive reduction for pulmonary nodule detection and segmentation. In *MICCAI*, pages 266–274. Springer, 2019.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, pages 2097–2106, 2017.

Yingxia Xia, Lingxi Xie, Fengze Liu, Zhuotun Zhu, Elliot K Fishman, and Alan L Yuille. Bridging the gap between 2d and 3d organ segmentation with volumetric fusion net. In *MICCAI*, pages 445–453. Springer, 2018.

Yutong Xie, Yong Xia, Jianpeng Zhang, David Dagan Feng, Michael Fulham, and Weidong Cai. Transferable multi-model ensemble for benign-malignant lung nodule classification on chest ct. In *MICCAI*, pages 656–664. Springer, 2017.

Ke Yan, Yifan Peng, Veit Sandfort, Mohammadhadi Bagheri, Zhiyong Lu, and Ronald M Summers. Holistic and comprehensive annotation of clinically significant findings on diverse ct images: Learning from radiology reports and label ontology. In *CVPR*, pages 8523–8532, 2019.

Ke Yan, Youbao Tang, Yifan Peng, Veit Sandfort, Mohammadhadi Bagheri, Zhiyong Lu, and Ronald M Summers. Munlan: Multitask universal lesion analysis network for joint lesion detection, tagging, and segmentation. In *MICCAI*, pages 194–202. Springer, 2019.

Ke Yan, Xiaosong Wang, Le Lu, Ling Zhang, Adam P Harrison, Mohammadhadi Bagheri, and Ronald M Summers. Deep lesion graphs in the wild: relationship learning and organization of significant radiology image findings in a diverse large-scale lesion database. In *CVPR*, pages 9261–9270, 2018.

Jiancheng Yang, Rongyao Fang, Bingbing Ni, Yamin Li, Yi Xu, and Linguo Li. Probabilistic radiomics: Ambiguous diagnosis with controllable shape analysis. In *MICCAI*, pages 658–666. Springer, 2019.

Qihang Yu, Lingxi Xie, Yan Wang, Yuyin Zhou, Elliot K Fishman, and Alan L Yuille. Recurrent saliency transformation network: Incorporating multi-stage visual cues for small organ segmentation. In *CVPR*, pages 8280–8289, 2018.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Matthew J Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J Geras, Joe Katsnelson, Hersh Chandarana, et al. fastmri: An open dataset and benchmarks for accelerated mri. *arXiv preprint arXiv:1811.08539*, 2018.

Wei Zhao, Jiancheng Yang, Bingbing Ni, Dexi Bi, Yingli Sun, Mengdi Xu, Xiaoxia Zhu, Cheng Li, Liang Jin, Pan Gao, et al. Toward automatic prediction of egfr mutation status in pulmonary adenocarcinoma with 3d deep learning. *Cancer medicine*, 2019.
[68] Wei Zhao, Jiancheng Yang, Yingli Sun, Cheng Li, Weilan Wu, Liang Jin, Zhiming Yang, Bingbing Ni, Pan Gao, Peijun Wang, et al. 3d deep learning from ct scans predicts tumor invasiveness of subcentimeter pulmonary adenocarcinomas. *Cancer research*, 78(24):6881–6889, 2018. 2, 3

[69] Hao Zheng, Yizhe Zhang, Lin Yang, Peixian Liang, Zhuo Zhao, Chaoli Wang, and Danny Z Chen. A new ensemble learning framework for 3d biomedical image segmentation. In *AAAI*, volume 33, pages 5909–5916, 2019. 2, 3

[70] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 3–11. Springer, 2018. 3

[71] Zongwei Zhou, Vatsal Sodha, Md Mahfuzur Rahman Siddiquee, Ruibin Feng, Nima Tajbakhsh, Michael B Gotway, and Jianming Liang. Models genesis: Generic autodidactic models for 3d medical image analysis. In *MICCAI*, pages 384–393. Springer, 2019. 3, 5, 6, 7, 8
Appendix

A. Details of Proof-of-Concept Dataset

For generating the 2D dataset, we first equally divide a blank 48 × 48 2D image into four 24 × 24 pieces. We randomly choose 3 out of the 4 pieces and in each of the selected piece, we generate a random-size circle or square with same probability at random center. The size is limited in the 24 × 24 piece. Thereby, the generated shape is guaranteed to be non-overlapped. Similarly, for generating 3D dataset, we equally divide a blank 48 × 48 × 48 3D volume into eight 24 × 24 × 24 pieces. We randomly choose 4 out of the 8 pieces and in each of the selected piece, we generate a random-size cone, pyramid, cube, cylinder or sphere with same probability at random center. The size is limited in the 24 × 24 × 24 piece. Thereby, the generated shape is guaranteed to be non-overlapped. For both 2D and 3D datasets, we add \( \mathcal{N}(0, 0.5) \) Gaussian noise on each pixel / voxel. See Fig. A1 for samples of the proof-of-concept 2D and 3D dataset.

B. More Results on LIDC-IDRI

Apart from the ResNet [24], we further experiment with the proposed ACS convolutions on LIDC-IDRI lung nodule classification and segmentation task, using VGG [53] and DenseNet [27]. The experiment settings are exactly the same with ResNet-18, which is stated in the Sec. 4.2. As depicted in Table A1 and A2, the results are consistent with the ResNet-18 performance. The 3D (3D and ACS) models outperform the 2D (2.5D) ones. The randomly-initialized ACS models are comparable or better than the 3D models; when pretrained with 2D datasets (e.g., ImageNet [13]), the ACS models consistently outperform the 3D ones.

| Models      | Segmentation | Classification |
|-------------|--------------|----------------|
| 2.5D VGG-16 r. | 67.4         | 87.4           |
| 2.5D VGG-16 p. | 71.8         | 92.6           |
| 3D VGG-16 r.  | 73.6         | 90.0           |
| 3D VGG-16 p. 13D [6] | 73.6 | 90.0 |
| ACS VGG-16 r. | 73.4         | 89.2           |
| ACS VGG-16 p. | 74.7         | 92.9           |

Table A2: DenseNet-121 [27] results on LIDC lungs on nodule segmentation (Dice global) and classification (AUC) performance. The 2.5D, I3D and ACS DenseNet-121 p. are pretrained on ImageNet [13].

| Seg | Cls | Memory (Seg) | Time (Seg) |
|-----|-----|--------------|------------|
| 2D r. | 68.8 | 89.4 | 5.0 Gb 0.57 s |
| 3D r. | 74.7 | 90.3 | 5.0 Gb 1.01 s |
| ACS r. | 75.1 | 92.5 | 6.6 Gb 0.95 s |

Table A3: Model performance, memory consuming and runtime speed of 2D (2.5D) and 3D and ACS convolutions. Due to the engineering issues, the memory of ACS convolutions is larger than that of 2D (2.5D) and 3D convolutions, yet theoretically identical (see Table 3). It is expected to be fixed (6.6 Gb to 5.0 Gb) in further implementation by custom memory checkpointing. Even though time complexity of ACS and 2D convolutions is the same, the parallelism of the ACS convolutions is weaker than that of 2D convolutions. Thereby, the actual runtime speed of ACS convolutions is slower than that of 2D convolutions.

D. Implementation of ACS Convolutions

We provide a PyTorch implementation of ACS convolutions, available at http://github.com/m3dv/ACSConv. Actual memory consuming and runtime speed are reported in Table A3. Using the provided function, 2D CNNs could be converted into ACS CNNs for 3D images, with a single line of code.

```python
import torchrom torchvision.models import resnet18
from acsconv.converters import ACSConverter
# model_2d is a standard PyTorch 2D model
model_2d = resnet18(pretrained=True)
B, C_in, H, W = (1, 3, 64, 64)
input_2d = torch.rand(B, C_in, H, W)
output_2d = model_2d(input_2d)
# model_3d is dealing with 3D data
model_3d = ACSConverter(model_2d)
B, C_in, D, H, W = (1, 3, 64, 64)
input_3d = torch.rand(B, C_in, D, H, W)
output_3d = model_3d(input_3d)
```

C. Qualitative Results on Nodule Segmentation

We visualize the segmentation masks generated by the 2.5D, 3D and ACS ResNet-18, with or without pretraining in Fig. A2. Combined the visualization with overall performance (Table 5), ACS p. segment the target nodules more precisely than other counterparts in general.
Figure A1: Samples of the proof-of-concept 2D and 3D datasets. Images in the first row are two 2D samples, while those in the next three rows are three 3D samples. Images with blue figures are input (before adding noise), while images with red figures are target segmentations.

Figure A2: Visualization of the segmentation masks generated by the 2.5D, 3D and ACS ResNet-18, with or without pre-training. The number on top of each image indicates the Dice per case of the sample.